# 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 $\delta$.

**CL-Scheme-Verify** $(ID, MPK, M, \delta) \rightarrow Accept/Reject$**.**
By verifying signature $\delta$ 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 $\delta'$ 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 $\delta$ 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 $\delta$. 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 $\delta$, 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 $\delta$ 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 $\delta$, he or she also can believe that the received message $m_3$ is equal to the value $m_3$ in signature $\delta$. 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 $\delta_{modify}$ 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 $\delta''$ 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.