An Improved Certificateless Signcryption in the Standard Model

Lin Cheng and Qiaoyan Wen (Corresponding author: Lin Cheng)

State Key Laboratory of Networking and Switch Technology, Beijing University of Posts and Telecommunications Beijing 100876, China

(Email: stonewoods302@163.com)

(Received Mar. 21, 2013; revised and accepted Jan. 7 & Feb. 6, 2014)

Abstract

Signeryption is a cryptographic primitive which can offer simultaneously security requirements of confidentiality and authentication, and is more efficient than the traditional sign-then-encrypt way. Recently, Liu et al. proposed the first certificateless signeryption scheme in the standard model. However, their scheme is proved to have some security weaknesses. In this paper, we propose a corrected version of Liu et al.'s scheme and prove our scheme is indistinguishable against adaptive chosen ciphertext attacks and is existentially unforgeable against chosen message attacks in the standard model. Performance analysis shows the new scheme has smaller public parameter size than the previous certificateless signeryption schemes without using the random oracles.

Keywords: Cryptography, provable security, signcryption

1 Introduction

In traditional public key cryptography, it needs a certificate issued by the certification authority (CA) to achieve authentication of the user's public key. However, the cost of certificate management is very high. To conquer this problem in traditional public key cryptography, Shamir [18] introduced the Identity-based cryptography. In identity-based cryptography, the user's public key is derived directly from its name, email-address or other identity information, but it requires a trusted third party called Key Generation Center (KGC) generate the user's private key. Unfortunately, we are confronted with the key escrow problem in identity-based cryptography, that is, KGC knows user's private key so that it can decrypt any ciphertext and sign any message on behalf of the user. At 2003, Al-Riyami and Paterson [1] introduced certificateless public key cryptography, which resolves the inherent key escrow problem in identity-based cryptography, without requiring certificates as used in traditional public key cryptography. In certificateless public key cryptography, the user's public key is independently generated by the user, and the user's private key is a combination partial private key computed by KGC and some user-chosen secret value, in such a way that the key escrow problem can be eliminated without requiring certificates.

Confidentiality and authenticity are two fundamentally different security requirements and realized through encryption and signature schemes respectively. A natural solution to offering simultaneously both requirements is using sign-then-encrypt approach. Signcryption, first introduced by Zheng [26], is a cryptographic primitive that combines the functionality of public encryption with digital signature and is more efficient than the traditional signature-then-encrypt approach. Since then, many signcryption schemes [7, 11, 19, 23] were proposed. The first certificateless signcryption (CLSC) scheme was introduced by Barbosa and Farshim [3]. Later, some efficient CLSC schemes were proposed [12, 22, 25]. However, all of these CLSC schemes are provably secure in the random oracle model [5], which can only be considered as a heuristic argument [6]. It has been shown in [4] that the security of the scheme may not preserve when the random oracle is instantiated with a particular hash function such as SHA-1. Based on Waters' identity-based encryption scheme [20] and its variants [8, 9, 13, 24], Liu et al. [14] introduced the first CLSC scheme in the standard model. Unfortunately, in [15, 17], Liu et al.'s CLSC scheme [14] is proved to be not secure against a type I adversary who can compromise users secret value or replace user public key, but neither compromise master secret key nor get access to partial private key. Weng et al. [21] showed that Liu et al.'s CLSC scheme [14] is also not secure against the malicious-but-passive KGC attack [2], where a malicious KGC can control the generation of master public/secret key pair, but cannot compromise user's secret value nor replace any public key. Though a rescued scheme has been proposed by Jin et al. [10], it still can not resist the attacks in [21]. This is because Jin et al.'s scheme has the same signcryption algorithm as Liu et al.'s CLSC scheme [14]. To the best of our knowledge, a secure CLSC scheme without random oracles is still an open problem. In this paper, we propose an improved Liu et al.'s CLSC scheme which can resist all the attacks in [15, 17, 21]. In addition, by using Naccache's methods [16], our new scheme has a smaller system parameters size than schemes [10, 14].

The rest paper is organized as follows. We provide some preliminaries in Section 2. Then, we recall the definition of certificateless signcryption scheme and its security model in Section 3. We propose a corrected version of Liu et al.'s scheme in Section 4. Its formal security proof is presented in Section 5. Finally a concluding remark is given in Section 6.

2 Preliminaries

In this Section, we recall the bilinear pairing and complexity assumptions [14].

2.1 Bilinear Pairing

Let G and G_T be two (multiplicative) cyclic groups with prime order p. A bilinear pairing is a map $e: G \times G \to G_T$ with the following properties:

- 1) Bilinear: $\forall g_1, g_2 \in G, \forall a, b \in Z_p^*$, we have $e\left(g_1^a, g_2^b\right) = e\left(g_1, g_2\right)^{ab}$;
- 2) Non-degeneracy: There exist $g_1, g_2 \in G$ such that $e(g_1, g_2) \neq 1_{G_T}$, where 1_{G_T} denotes the identity element of group G_T ;
- 3) Computability: There exists an efficient algorithm to compute $e(g_1, g_2)$ for $\forall g_1, g_2 \in G$.

2.2 Decisional Bilinear Diffie-Hellman Assumption

Given a group G of prime order p with generator g, a bilinear pairing $e: G \times G \longrightarrow G_T$ and elements $g^a, g^b, g^c \in G, \ e(g,g)^z \in G_T$ where a, b, c, z are selected randomly from Z_p^* . Let $\beta \in 0, 1$ be a random binary coin. If $\beta = 1$, it outputs the tuple $(g, A = g^a, B =$ $g^b, C = g^c, Z = e(g,g)^{abc}$. If $\beta = 0$, it outputs the tuple $(g, A = g^a, B = g^b, C = g^c, Z = e(g,g)^z)$. The decisional Bilinear Diffie-Hellman (**DBDH**) assumption is that no *t*-time algorithm \mathcal{B} has at least ϵ advantage in determining the value of β , where the advantage is defined as $| Pr[1 \longleftarrow \mathcal{B}(g, g^a, g^b, g^c, e(g, g)^{abc})] - Pr[1 \longleftarrow$ $\mathcal{B}(g, g^a, g^b, g^c, e(g, g)^z)].$

2.3 Computational Diffie-Hellman Assumption

Given (g, g^a, g^b) where a, b are selected randomly from Z_p^* . The computational Diffie-Hellman (CDH) assumption is that no *t*-time algorithm \mathcal{B} has at least ϵ advantage in computing g^{ab} , where the advantage is defined as $Pr[\mathcal{B}(g, g^a, g^b) = g^{ab}].$

3 Formal Model of Certificateless Signcryption

3.1 Definition of Certificateless Signcryption

- A CLSC scheme consists of the following six algorithms:
- **Setup**(k). On input a security parameter k, this setup algorithm generates a master public/secret key pair (mpk, msk).
- **PartialPrivateKeyGen.** On input msk, mpk, and a user identity ID, it generates the user's partial Private key psk_{ID} .
- **UserKeyGen.** On input mpk and psk_{ID} , it generates the public/private key pair (pk_{ID}, sk_{ID}) .
- **User-Key-Generate.** On input *params* and a user identity ID, it returns a randomly chosen secret value x_{ID} and a corresponding public key pk_{ID} for the user.
- **Private-Key-Extract.** On input *params*, a user's partial private key psk_{ID} and secret value x_{ID} , it returns the user's full private key sk_{ID} .
- **Signcrypt.** On input *params*, a message M, a sender's private key sk_S , identity ID_S and public key pk_S , and a receivers identity ID_R and public key pk_R , it returns a ciphertext δ or an error symbol \perp .
- **Unsigncrypt.** On input a ciphertext δ , the receiver's private key sk_R , and the sender's public key pk_S , it outputs a plaintext M or an error symbol \perp .

For consistency, these algorithms must satisfy that if $\delta = Signcrypt(params, M, sk_S, ID_R, pk_R)$, then Mshould be a part of $Unsigncrypt(params, \delta, sk_R, pk_S)$.

3.2 Security Models

An adversary \mathcal{A} is allowed to access to the following oracles.

- **Public-Key-Broadcast-Oracle.** On input of any identity ID, challenger returns corresponding public key. If such a key does not yet exist, challenger computes the corresponding public key pk_{ID} and returns pk_{ID} to \mathcal{A} .
- **Partial-Private-Key-Oracle.** On input of any identity ID, challenger computes the corresponding partial private key psk_{ID} for this identity and returns psk_{ID} to \mathcal{A} .
- **Public-Key-Replacement-Oracle.** On input of an identity ID and a new valid public key value pk'_{ID} , challenger replaces the current public key with pk'_{ID} .

- Private-Key-Extract-Oracle. On input of an identity ID whose public key was not replaced, challenger computes the private key sk_{ID} for this identity and returns sk_{ID} to \mathcal{A} .
- Signcrypt. On input of a sender's identity ID_S , a receiver's identity ID_R and a message M, challenger responds by running the **Signcrypt** algorithm on the message M, the sender's private key sk_S and the receiver's public key pk_R . It is possible for the challenger not to be aware of the sender's secret value when the associated public key has been replaced. In this case, we require \mathcal{A} to provide the sender's secret key.
- **Unsignerypt.** On input of a ciphertext δ , a sender's identity ID_S and a receiver's identity ID_R , challenger returns the result of running the **Unsigncrypt** algorithm on the ciphertext δ , the receiver's private key sk_R and the sender's public key pk_S . It is possible for the challenger not aware of the receiver's secret value when the associated public key has been replaced. In this case, \mathcal{A} is required to provide the receiver's secret key.

Definition 1. A CLSC scheme is said to have the indistinguishability against adaptive chosen ciphertext attacks property (IND-CLSC-CCA), if no polynomially bounded adversaries $\mathcal{A}(\mathcal{A}_I \text{ and } \mathcal{A}_{II})$ have non-negligible advantage of winning the following game.

- **Initialization.** If the adversary is A_I , challenger runs algorithm Setup to generate the master key msk and the master public key mpk, and then gives mpk to \mathcal{A}_{I} and keeps msk secret. If the adversary is A_{II} , adversary \mathcal{A}_{II} runs algorithm **Setup** to generate the master secret key msk and the master public key mpk. \mathcal{A}_{II} then gives mpk and msk to challenger.
- **Phase 1.** In this phase, A adaptively performs a polynomially bounded number of oracle queries. Actually, \mathcal{A}_{II} does not need to issue partial private key queries, since it can compute them from the master key by itself.
- Challenge. At the end of Phase 1, the adversary outputs two distinct identities ID_{S^*} , ID_{R^*} and two equal length messages $\{M_0, M_1\}$. The challenger chooses a bit γ randomly and signcrypts M_{γ} under the ID_{S^*} 's private key and the ID_{R^*} 's public key to produce δ^* . The challenger returns δ^* to the adversary.
- Phase 2. The adversary continues to probe the challenger with the same type of queries that it made in Phase 1. To capture insider security, the adversaries are assumed to have access to the private key of the sender ID_{S^*} of a signcrypted message.
- **Response.** The adversary returns a bit γ' . We say that Let G and G_T be groups of prime order p and g be a the adversary wins the game if $\gamma' = \gamma$ and the adversary fulfills the following conditions:

- 1) \mathcal{A}_I , \mathcal{A}_{II} cannot extract the private key for ID_{R^*} at any point.
- 2) \mathcal{A}_I cannot extract the private key for any identity if the corresponding public key has already been replaced.
- 3) \mathcal{A}_I cannot extract the partial private key of ID_{R^*} if \mathcal{A}_I replaced the public key pk_{R^*} before the challenge phase.
- 4) In Phase 2, \mathcal{A}_I cannot make an unsigncryption query on the challenge ciphertext δ^* under ID_{S^*} and ID_{R^*} unless the public key pk_{S^*} of the sender or that of the receiver pk_{R^*} used to signcrypt M_{γ} has been replaced after the challenge was issued.
- 5) In Phase 2, A_{II} cannot make an unsigncryption query for the challenge ciphertext δ^* under ID_{S^*} and ID_{R^*} and public key pk_{R^*} that were used to signcrypt M_{γ} .

 \mathcal{A} 's advantage is defined as $Adv_A^{IND-CL-CCA} =$ $|2Pr\left[\gamma'=\gamma\right]-1|.$

Definition 2. A CLSC scheme is said to be secure against an existential forgery for adaptive chosen message attacks (EUF-CLSC-CMA), if no polynomially bounded adversaries (\mathcal{A}_{I} and \mathcal{A}_{II}) have non-negligible advantage of winning the following game.

Initialization. It is the same as above.

- Queries. A may adaptively issue a polynomially bounded number of queries to \mathcal{B} as above Phase 1. To deal with the insider security, assuming the adversary can gain access to the private key of the receiver of a signcrypted message.
- **Output.** Eventually, \mathcal{A} outputsanewtriple $(\delta^*, ID_{S^*}, ID_{R^*})$, which is not produced by the signcryption query. The adversary wins if the result of unsigncrypt $(\delta^*, ID_{S^*}, pk_{S^*}, sk_{R^*})$ is not the symbol \perp and the queries are subject to the following constraints:
 - 1) \mathcal{A}_I , \mathcal{A}_{II} cannot extract the private key for ID_{S^*} at any point.
 - 2) \mathcal{A}_I cannot extract the private key for any identity if the corresponding public key has already been replaced.
 - 3) \mathcal{A}_I cannot extract the partial private key of ID_{S^*} .

We define \mathcal{A} 's success probability in the game above to be $Succ_{\mathcal{A}}^{EUF-CLSC-CMA} = Pr[\mathcal{A}wins].$

Improved Scheme 4

generator of G, and let $e: G \times G \to G_T$ be a bilinear pairing. $\phi: \Re \to G_T$ is a bijection while ϕ^{-1} is its inverse mapping, \Re is a subset of $\{0,1\}^{m+n}$ with p elements. Identity ID will be represented as n dimensional vectors $d_{ID} = (d_{ID,1}, \cdots, d_{ID,n})$ where each $d_{ID,i}$ is an $\ell\text{-bit}$ integer, and $n'=n\cdot\ell$ is the length of an identity in binary string representation. $H': \{0,1\}^* \to \{0,1\}^{n'}, H:$ $\{0,1\}^* \to \{0,1\}^m$ are two collision-resistant hash functions.

- **Setup.** The KGC selects randomly values α, u', v' in Z_p^* and two random vectors $\mathbf{U} = (u_i)_n$, $\mathbf{V} = (v_j)_m$ and then computes $g_1 = g^{\alpha}$, and selects randomly $g_2 \in G$. The master public key mpk and the master secret key msk are respective $(q_1, q_2, u', v', \mathbf{U}, \mathbf{V})$ and q_2^{α} .
- PartialPrivateKeyGen. The KGC picks a random value $r \in Z_p^*$ and computes partial private key psk_{ID} = $(psk_{ID,1}, psk_{ID,2})$ $\begin{pmatrix} g_2^{\alpha}(u'\prod_{i=1}^{i=n}u_i^{d_{ID,i}})^r, g^r \end{pmatrix}.$ The receiver's partial private keys are The sender and the

$$psk_{S} = (psk_{S,1}, psk_{S,2}) = \left(g_{2}^{\alpha} \cdot (u'\prod_{i=1}^{i=n}u_{i}^{d_{S,i}})^{r}, g^{r}\right)$$
$$psk_{R} = (psk_{R,1}, psk_{R,2}) = \left(g_{2}^{\alpha} \cdot (u'\prod_{i=1}^{i=n}u_{i}^{d_{R,i}})^{r}, g^{r}\right)$$

UserkeyGen. Pick a secret value $x_{ID} \in Z_p^*$, and generate public key $pk_{ID} = \{pk_{ID,1}, pk_{ID,2}, pk_{ID,3}\} =$ $\{g^{x_{ID}}, g_1^{x_{ID}}, g_2^{x_{ID}}\}$. Then it randomly picks r' from Z_p^* and computes private key sk_{ID} as

$$(sk_{ID,1}, sk_{ID,2}) = \left(psk_1^{x_{ID}^2} \cdot (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^{r'}, psk_2^{x_{ID}^2} \cdot g^{r'} \right)$$
$$= \left(g_2^{\alpha x_{ID}^2} \cdot (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^t, g^t \right)$$

where $t = rx_{ID}^2 + r'$.

Signcrypt. To send a message $M \in \{0,1\}^m$ to an identity ID_R with public key pk_R , first check whether the public key pk_R is correctly formed, by checking $e(pk_{R,1}, g_1) = e(g, pk_{R,2})$ and $e(pk_{R,1}, g_2) =$ $e(g, pk_{R,3})$. If not, output \perp and abort the algorithm. Otherwise, the sender first selects a random value $s \in Z_p^*$ and $R \in \{0,1\}^n$ such that $M \parallel R \in \Re$, and compute: (Let w be a n-bit string and w_i be the *i*-th bit w).

$$\begin{aligned}
\delta_{1} &= \phi(M ||R) \cdot e(pk_{R,2}, pk_{R,3})^{s}, \\
\delta_{2} &= g^{s}, \\
\delta_{3} &= (u' \prod_{i=1}^{i=n} u_{i}^{d_{R,i}})^{s}, \\
\delta_{4} &= sk_{S,2}, \\
\delta_{5} &= sk_{S,1} \cdot F(w)^{s}
\end{aligned}$$

where
$$w = H(\delta_1, \delta_2, \delta_3, \delta_4, R, pk_S, pk_R) \in \{0, 1\}^m$$

and $F(w) = v' \prod_{j=1}^m v_j^{w_j}$.

Unsignerypt. Upon receiving a ciphertext C $(\delta_1, \delta_2, \delta_3, \delta_4, \delta_5)$, first compute

$$\phi^{-1}\left(\delta_1 \cdot e\left(\delta_3, sk_{R,2}\right) / e\left(\delta_2, sk_{R,1}\right)\right) \to M \|R\|$$

and then check whether the public key pk_S is correctly formed, by checking $e(pk_{S,1}, g_1) = e(g, pk_{S,2})$ and $e(pk_{S,1}, g_2) = e(g, pk_{S,3})$. If not, output \perp and abort the algorithm. Otherwise, Accept the message M if

$$e(\delta_{5},g) = e(pk_{S,2}, pk_{S,3}) e(u' \prod_{i=1}^{i=n} u_{i}^{d_{S,i}}, \delta_{4}) e(F(w), \delta_{2})$$

where $w = H(\delta_1, \delta_2, \delta_3, \delta_4, R, pk_S, pk_R) \in \{0, 1\}^m$.

It is easy to see the proposed scheme is consistent. In the next Section, we will give a formal security proof.

Analysis Improved 5 of the Scheme

Security Analysis 5.1

We now prove that the above proposed scheme is secure in the standard model. Our proof very much falls along the lines of the security proof in Liu et al.'s scheme [14].

Theorem 1. The new CLSC scheme is indistinguishable against adaptive chosen ciphertext attacks (IND - CLSC - CCA) in the standard model under the decisional BDH intractability assumption.

This theorem follows Lemmas 1 and 2.

Lemma 1. The new CLSC scheme is indistinguishable against the Type I attacker in the standard model if the decisional BDH assumption holds.

Proof. Assume there exists a type I adversary \mathcal{A}_I against our scheme. We construct a PPT simulator \mathcal{B} that makes use of \mathcal{A}_I to solve the **DBDH** problem with probability at least ϵ' and in time at most t'. \mathcal{B} is given a **DBDH** problem instance $(g, A = g^a, B = g^b, C = g^c, Z)$ and replies the queries of \mathcal{A}_I as follows.

- Setup. Let $l_v = 2(q_{pp} + q_p + q_s + q_u)$ and $l_w = 2q_u$. \mathcal{B} randomly chooses the following elements:
 - 1) Two integers $k_v (0 \le k_v \le 2^{\ell} \cdot n), k_w (0 \le k_w \le m).$ We assume that $2^{\ell}(n+1)l_v < p, (m+1)l_w < p$ for the given values of n and m.
 - 2) An integer $x'(x' \in Z_{l_v})$ and a vector $\vec{X} =$ $(x_i)_n (x_i \in Z_{l_n}).$
 - 3) An integer $z'(z' \in Z_{l_w})$ and a vector \overrightarrow{Z} = $(z_i)_m (z_i \in Z_{l_w}).$

4) Two integers
$$y', t' \in Z_p$$
 and three vectors
 $\overrightarrow{Y} = (y_i)_n (y_i \in \mathbf{Z_p}), \ \overrightarrow{T} = (t_j)_m (t_j \in \mathbf{Z_p}),$
 $\overrightarrow{w} = (w_j)_m (w_j \in \mathbf{Z_2}).$

Identity ID will be represented as n dimensional vectors $d_{ID} = (d_{ID,1}, \ldots, d_{ID,n})$ where each $d_{ID,i}$ is an ℓ -bit integer, and $n' = n \cdot \ell$ is the length of an identity ID in binary string representation. For convenience, we define as follows:

$$F(d_{ID}) = x' - l_v k_v + \sum_{i=1}^{i=n} x_i d_{ID,i},$$

$$J(d_{ID}) = y' + \sum_{i=1}^{i=n} y_i d_{ID,i},$$

$$K(w) = z' - l_w k_w + \sum_{j=1}^{j=m} z_j w_j,$$

$$L(w) = t' + \sum_{j=1}^{j=m} w_j t_j.$$

Then the challenger constructs a set of public parameters as follows:

$$g_{1} = g^{a},$$

$$g_{2} = g^{b},$$

$$u' = g_{2}^{z'-l_{v}k_{v}}g^{y'}$$

$$v' = g_{2}^{z'-l_{w}k_{w}}g^{t}$$

$$u_{i} = g_{2}^{x_{i}}g^{y_{i}},$$

$$v_{j} = g_{2}^{z_{j}}g^{t_{j}}.$$

Note that the master secret key will be $g_2^a = g^{ab}$ and the following equation holds:

$$u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}} = g_2^{F(d_{ID})} g^{J(d_{ID})}$$
$$v' \prod_{j=1}^{j=m} v_j^{w_j} = g_2^{K(w)} g^{L(w)}.$$

- **Phase 1.** In the query phase, \mathcal{B} answers the queries of \mathcal{A}_I as follows:
 - **Public-Key-Broadcast-Oracle.** Upon receiving a query for a public key of an identity ID, if (ID, pk_{ID}) exists in PublicKeyList, \mathcal{B} returns pk_{ID} as the answer. Otherwise, \mathcal{B} runs the algorithm **UserKeyGen** to generate public key $pk_{ID} = \{pk_{ID,1}, pk_{ID,2}, pk_{ID,3}\} = \{g^{x_{ID}}, g_1^{x_{ID}}, g_2^{x_{ID}}\}$. \mathcal{B} adds (ID, x_{ID}) to SecretValueList and adds (ID, pk_{ID}) to PublicKeyList, then returns the public key pk_{ID} as the answer.
 - Partial-Private-Key-Extract-Oracle. Upon receiving a query for a partial private key of

an identity ID, \mathcal{B} first searches PartialPrivateKeyList for a tuple (ID, psk_{ID}) . If it exists, \mathcal{B} returns (ID, psk_{ID}) as the answer. Otherwise, \mathcal{B} can construct a partial private key by assuming $F(d_{ID}) \neq 0 \mod p$. \mathcal{B} randomly chooses $r \in \mathbf{Z}_{\mathbf{p}}$ and computes a partial private key:

$$psk_{ID} = (psk_{ID,1}, psk_{ID,2})$$

= $(g_1^{-J(d_{ID})/F(d_{ID})}(u'\prod_{i=1}^{i=n} u_i^{d_{ID,i}})^r, g_1^{-1/F(d_{ID})}g^r).$

 psk_{ID} is a valid partial private key for the identity ID shown as follows.

$$psk_{ID,1} = g_1^{-J(d_{ID})/F(d_{ID})} (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^r$$

$$= g_2^a (g_2^{F(d_{ID})} g^{J(d_{ID})})^{r-a/F(d_{ID})}$$

$$= g_2^a (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^{r'},$$

$$psk_{ID,2} = g_1^{-1/F(d_{ID})} g^r = g^{r-a/F(d_{ID})} = g^{r'}$$

where $r' = r - a/F(d_{ID})$. From $-p < F(d_{ID}) < p$, we conclude that $F(d_{ID}) = 0 \mod p$ implies $F(d_{ID}) = 0 \mod l_v$, so $F(d_{ID}) \neq 0 \mod l_v$ suffices to have $F(d_{ID}) \neq 0 \mod p$. \mathcal{B} adds (ID, psk_{ID}) to its PartialPrivateKeyList and returns the partial private key psk_{ID} as the query output. If, on the other hand, $F(d_{ID}) = 0 \mod p$, \mathcal{B} aborts and randomly chooses its guess β' of β .

Private-Key-Extract-Oracle. Upon receiving a query for a private key of an identity ID, if the PrivateKeyList contains (ID, sk_{ID}) , \mathcal{B} returns sk_{ID} . Otherwise, \mathcal{B} can construct a private key by assuming $F(d_{ID}) \neq 0 \mod p$. \mathcal{B} searches SecretValueList to find out x_{ID} . If it does not exist, \mathcal{B} runs the algorithm **UserKeyGen** to generate secret-public key pair (x_{ID}, pk_{ID}) , and adds (ID, x_{ID}) to SecretValueList and adds (ID, pk_{ID}) to PublicKeyList, then \mathcal{B} chooses $r \in Z_p$ randomly and computes

$$sk_{ID,1} = (g_1^{x_{ID}^2})^{-J(d_{ID})/F(d_{ID})} (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^r$$

$$= g_2^{ax_{ID}^2} (g_2^{F(d_{ID})} g^{J(d_{ID})})^{r-ax_{ID}^2/F(d_{ID})}$$

$$= g_2^{ax_{ID}^2} (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^t,$$

$$sk_{ID,2} = (g_1^{x_{ID}^2})^{-1/F(d_{ID})} g^r$$

$$= g^{r-ax_{ID}^2/F(d_{ID})} = g^t,$$

where $t = r - ax_{ID}^2 / F(d_{ID})$. \mathcal{B} adds (ID, sk_{ID}) to PrivateKeyList and returns the private key

 sk_{ID} . If, on the other hand, $F(d_{ID}) = 0 \mod p$, \mathcal{B} aborts and randomly chooses its guess β' of β .

- **Public-Key-Replacement-Oracle.** Upon receiving a query for replacing the current public key pk_{ID} of an identity ID with a new and valid public key pk'_{ID} , \mathcal{B} finds out pk_{ID} in its PublicKeyList, and replaces it with the new public key pk'_{ID} . If pk_{ID} does not exist, \mathcal{B} directly sets $pk_{ID} = pk'_{ID}$, while the adversary delivers x'_{ID} to \mathcal{B} . Then \mathcal{B} adds (ID, x_{ID}) to SecretValueList and adds (ID, pk_{ID}) to PublicKeyList. \mathcal{B} sets sta = 1 for the identity ID.
- Signcrypt-Oracle. Upon receiving a query for a message M and identities ID_S and ID_R , if $F(d_S) \neq 0 \mod p$, \mathcal{B} obtains the public key pk_R of ID_R and the private key sk_S of ID_S by running Public-Key-Broadcast-Oracle and Private-Key-Extract-Oracle, then runs the Signcrypt algorithm to create a ciphertext δ and sends it to \mathcal{A}_I . If $F(d_S) = 0 \mod p \mathcal{B}$ aborts and randomly chooses its guess β' of β .
- **Unsigncrypt-Oracle.** Upon receiving a unsigncryption query on a ciphertext $\delta = (\delta_1, \delta_2, \delta_3, \delta_4, \delta_5)$, and identities ID_S and ID_R , \mathcal{B} computes the unsigncryption as follows:
 - 1) If sta = 0 for ID_R , \mathcal{B} obtains the private key sk_R of ID_R by running **Private-Key-Extract-Oracle** (assume $F(d_R) \neq 0 \mod l_v$), then runs the **Unsigncrypt** algorithm to recover the message M. \mathcal{B} executes the verification part of the **Unsigncrypt** algorithm. If the verification does not succeed, \mathcal{B} returns a failure symbol \perp . Otherwise, returns M to \mathcal{A}_I .
 - 2) If sta = 1 for ID_R , or $F(d_R) = 0 \mod l_v$, \mathcal{B} will try to decrypt the ciphertext δ . Assume $K(w) \neq 0 \mod l_w$, where w = $H(\delta_1, \delta_2, \delta_3, \delta_4, R, pk_S, pk_R)$. \mathcal{B} retrieves the secret value x_R s.t. pk_R , and computes $g_2^s = (\delta_5/(sk_{S,1}\delta_2^{L(w)}))^{1/K(w)}$ and M || R = $\Phi^{-1}(\delta_1/e(g_1, g_2^s)^{x_R^2})$. \mathcal{B} executes the verification part of the **Unsigncrypt** algorithm. If the verification does not succeed, \mathcal{B} returns a failure symbol \perp . Otherwise, returns M to \mathcal{A}_I . If $L(w) = 0 \mod l_w$, \mathcal{B} aborts and randomly chooses its guess β' of β .
- **Challenge.** At the end of the first stage, \mathcal{A}_I outputs two equal length messages M_0, M_1 together with two identities ID_S and ID_R on which it wishes to be challenged. If $F(d_{R^*}) \neq 0 \mod l_v$, \mathcal{B} aborts. Otherwise, chooses a random bit γ from $\{0, 1\}$ and constructs a ciphertext of M_{γ} as follows. Let pk_{S^*}, pk_{R^*} be ID_{S^*}, ID_{R^*} 's public keys, respectively. \mathcal{B} retrieves

the secret values x_{S^*} , x_{R^*} , $t_{S^*} \in Z_p$ and $R \in \{0, 1\}^n$ such that $M_{\gamma} || R \in \Re$, then computes as follows:

$$\begin{split} \delta_1^* &= \phi\left(M_{\gamma} \, \| R\right) \cdot Z^{x_{R^*}^2}, \\ \delta_2^* &= C, \\ \delta_3^* &= C^{J(d_{R^*})}, \\ \delta_4^* &= (g_1^{x_{S^*}^2})^{-1/F(d_{S^*})} g^{t_{S^*}}, \\ w^* &= H\left(\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, R, pk_{S^*}, pk_{R^*}\right). \end{split}$$

If $K(w^*) \neq 0 \mod p$, \mathcal{B} aborts. Otherwise, \mathcal{C} sets $\delta_5^* = (g_1^{x_{S^*}^2})^{-J(d_{S^*})/F(d_{S^*})}(u'\prod_{i=1}^{i=n}u_i^{d_{S^*,i}})^{t_{S^*}}C^{L(w^*)}$. \mathcal{B} returns the ciphertext $\delta^* = (\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, \delta_5^*)$ to the adversary.

- Phase 2. \mathcal{A}_I continues to perform the same type of queries made in Phase 1. But in this phase, \mathcal{A}_I can not make any **Unsigncrypt** query on the challenge ciphertext δ^* for ID_{S^*}, ID_{R^*} .
- **Guess.** Finally, \mathcal{A}_I outputs a guess γ' of γ . If $\gamma' = \gamma$ then \mathcal{B} outputs 1 indicating $Z = e(g,g)^{abc}$, and else outputs 0 indicating Z is a random element of G_T .

Remark 1. Liu et al. showed their scheme is secure if the user's public key is with the correctly form $e(g_1, g_2)^{x_{ID}}$ i.e the \mathcal{A}_I can replace public key only by choosing a different secret value x'_{ID} . However, in [15, 17], Liu et al.'s scheme is showed that a Type I adversary \mathcal{A}_I can cheat the sender and decrypt the ciphertext by replacing receiver's public key with $e(g, g)^{x'_R}$. The weakness in [14] is that receiver's public key pk_R is just a group element $e(g_1, g_2)^{x_R}$, and it can not check whether the public key pk_R is correctly formed during signcryption stage. In order to defend against attacks [15, 17], we revise **UserkeyGen** so that the receiver's public key pk_R can be checked whether it is correctly formed during signcryption stage. We omit the analysis of the success probability and the time complexity, which are similar to that of Liu et al. [14].

Lemma 2. The new CLSC scheme is indistinguishable against the Type II attacker in the standard model if the decisional BDH assumption holds.

Proof. Assume there exists a type II adversary \mathcal{A}_{II} against our scheme. We construct a PPT simulator \mathcal{B} that makes use of \mathcal{A}_{II} to solve the **DBDH** problem with probability at least ϵ' and in time at most t'. \mathcal{B} is given a**DBDH** problem instance $(g, A = g^a, B = g^b, C = g^c, Z)$ and replies the queries of \mathcal{A}_{II} as follows.

Setup. Let $l_v = 2(q_p + q_s + q_u)$ and $l_w = 2q_u$. The Type II adversary \mathcal{A}_{II} chooses a random integer $\alpha \in \mathbf{Z}_{\mathbf{p}}$ as the master secret key and computes $g_1 = A^{\alpha}$. The other public parameters are identical to those of Theorem 1. Then \mathcal{A}_{II} sends all public parameters and the master secret key α to \mathcal{B} .

- **Phase 1.** A_{II} can compute partial private key of any identity by itself and carry out the following queries.
 - **Public-Key-Broadcast-Oracle.** Upon receiving a query for a public key of an identity ID, if (ID, pk_{ID}) exists in PublicKeyList, \mathcal{B} returns pk_{ID} as the answer. Otherwise, \mathcal{B} runs the algorithm **User-Key-Gen** to generate public key $pk_{ID} = (g^{x_{ID}}, A^{\alpha x_{ID}}, B^{x_{ID}}), \mathcal{B}$ adds (ID, x_{ID}) to SecretValueList and adds (ID, pk_{ID}) to PublicKeyList, and returns the public key to \mathcal{A}_{II} .
 - **Private-Key-Extract-Oracle.** Upon receiving a query for a private key of an identity ID, if the PrivateKeyList contains (ID, sk_{ID}) , \mathcal{B} returns sk_{ID} . Otherwise, \mathcal{B} can construct a private key by assuming $F(d_{ID}) \neq 0 \mod p$. \mathcal{B} first searches SecretValueList to find out x_{ID} . If it does not exist, \mathcal{B} runs the algorithm **UserKeyGen** to generate secret-public key pair (x_{ID}, pk_{ID}) , and adds (ID, x_{ID}) to SecretValueList and adds (ID, pk_{ID}) to PublicKeyList, then \mathcal{B} chooses $r \in Z_p$ randomly and computes

$$sk_{ID,1} = (A^{\alpha x_{ID}^2})^{-J(d_{ID})/F(d_{ID})} (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^r$$

$$= g_2^{a\alpha x_{ID}^2} (g_2^{F(d_{ID})} g^{J(d_{ID})})^{r-a\alpha x_{ID}^2/F(d_{ID})}$$

$$= g_2^{a\alpha x_{ID}^2} (u' \prod_{i=1}^{i=n} u_i^{d_{ID,i}})^t,$$

$$sk_{ID,2} = (A^{\alpha x_{ID}^2})^{-1/F(d_{ID})} g^r$$

$$= g^{r-a\alpha x_{ID}^2/F(d_{ID})} = g^t,$$

where $t = r - a\alpha x_{ID}^2/F(d_{ID})$. \mathcal{B} adds (ID, sk_{ID}) to the PrivateKeyList and returns the private key sk_{ID} . If $F(d_{ID}) = 0 \mod p$, \mathcal{B} simply aborts and randomly outputs a guess β' of β .

- **Signcrypt-Oracle.** For a signcryption query for a message M and identities ID_S and ID_R , \mathcal{B} answers the signcryption query in the same way as Lemma 1.
- **Unsigncrypt-Oracle.** Upon receiving a unsigncryption query on a ciphertext $\delta = (\delta_1, \delta_2, \delta_3, \delta_4, \delta_5)$, and a sender's identity ID_S and a receiver's identity ID_R , \mathcal{B} computes the unsigncryption as follows:
 - 1) If $F(d_R) \neq 0 \mod l_v$, \mathcal{B} searches PrivateKeyList to find out (ID_R, sk_R) , then performs the Unsigncrypt algorithm to recover the message M, and sends it to A_{II} . If the corresponding entry does not exist, \mathcal{B} obtains the private key sk_R of ID_R by running **Private-Key-Extract-Oracle**, then runs the **Unsigncrypt** algorithm to recover the message M. \mathcal{B} executes the verification part of the **Unsigncrypt** algorithm.

If the verification does not succeed, \mathcal{B} returns a failure symbol \perp . Otherwise, returns M to \mathcal{A}_{II} .

- 2) If $F(d_R) = 0 \mod l_v$, \mathcal{B} will try to decrypt the ciphertext δ . Assume $K(w) \neq 0 \mod$ l_w , where $w = H(\delta_1, \delta_2, \delta_3, \delta_4, R, pk_S, pk_R)$. \mathcal{B} searches PrivateKeyList to obtain ID_S 's private key sk_S (to deal with the insider security, we assume that the adversary has access to the private key of the sender) and retrieve the secret value x_R s.t. pk_R . \mathcal{B} can compute $g_2^s = (\delta_5 / (sk_{S,1}\delta_2^{L(w)}))^{1/K(w)},$ $M \| R = \Phi^{-1}(\delta_1 / e(A^{\alpha}, g_2^s)^{x_R^2}).$ Then \mathcal{B} executes the verification part of the Unsign**crypt** algorithm. If the verification does not succeed, \mathcal{B} returns a failure symbol \perp . Otherwise, returns M to \mathcal{A}_{II} . If L(w) = $0 \mod l_w, \mathcal{B}$ aborts and randomly chooses its guess β' of β .
- Challenge. At the end of the first stage, \mathcal{A}_{II} outputs two equal length messages M_0, M_1 together with two identities ID_S and ID_R on which it wishes to be challenged. If $F(d_{R^*}) \neq 0 \mod l_v$, \mathcal{B} aborts. Otherwise, chooses a random bit γ from $\{0, 1\}$ and constructs a ciphertext of M_γ as follows. Let pk_{S^*}, pk_{R^*} be ID_{S^*}, ID_{R^*} 's public keys, respectively. \mathcal{B} retrieves the secret values x_{S^*}, x_{R^*} and randomly chooses a bit $\gamma \in \{0,1\}, t_{S^*} \in Z_p$ and $R \in \{0,1\}^n$ such that $M_\gamma || R \in \mathfrak{R}$, then computes as follows: $\delta_1^* = \phi(M_b || R) \cdot Z^{\alpha x_{R^*}^2}, \delta_2^* =$ $C, \delta_3^* = C^{J(d_{R^*})}, \delta_4^* = (g_1^{x_{S^*}^2})^{-1/F(d_{S^*})}g^{t_{S^*}}, w^* =$ $H(\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, R, pk_{S^*}, pk_{R^*})$. If $K(w^*) \neq$ 0 mod p, \mathcal{B} aborts. Otherwise, \mathcal{C} sets $\delta_5^* = (g_1^{x_{S^*}^2})^{-J(d_{S^*})/F(d_{S^*})}(u'\prod_{i=1}^{i=n} u_i^{d_{S^*,i}})^{t_{S^*}}C^{L(w^*)}$. \mathcal{B} returns the ciphertext $\delta^* = (\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, \delta_5^*)$ to the adversary.
- **Phase 2.** \mathcal{A}_{II} continues to perform the same type of queries made in Phase 1. But in this phase, \mathcal{A}_{II} can not make any **Unsigncrypt** query on the challenge ciphertext δ^* for ID_{S^*}, ID_{R^*} .
- **Guess.** Finally, \mathcal{A}_{II} outputs a guess γ' of γ . If $\gamma' = \gamma$ then \mathcal{B} outputs 1 indicating $Z = e(g,g)^{abc}$, and else outputs 0 indicating Z is a random element of G_T .

Remark 2. Weng et al. [21] proved that Liu et al's scheme is not indistinguishable against a Type II adversary. That is, given a challenged ciphertext δ^* , \mathcal{A}_{II} could convert the challenged ciphertext δ^* into a new valid ciphertext δ' in phase 2. When adversary issues an unsigncryption query on the ciphertext δ' , the challenger has to return the underlying message M_{γ} to \mathcal{A}_{II} . With M_{γ} , adversary \mathcal{A}_{II} can certainly know the value γ , and thus wins the game. One of the main difference between Liu et al. and our proof is the signcryption ciphertext that returns at the stage of **Challenge**. In our proof, the challenged ciphertext δ^* includes a random binary string R which \mathcal{A}_{II} does not know, so the above defect can be avoided.

Theorem 2. The new CLSC scheme is existentially unforgeable against chosen message attacks (EUF-CLSC-CMA) in the standard model under the CDH intractability assumption.

This theorem follows Lemma 3 and 4.

Lemma 3. The new CLSC scheme is existentially unforgeable against the Type I attacker in the standard model if the CDH assumption holds.

Proof. Assume that there exists a Type I forger A_I against our scheme. In the following, we construct an algorithm \mathcal{B} to solve the CDH problem.

Suppose \mathcal{B} is given a random instance of the CDH problem (g^a, g^b) . Its goal is to output g^{ab} . The simulation process is the same as that described in Lemma 1. Finally, the adversary \mathcal{A}_I produces a new ciphertext $\delta^* = (\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, \delta_5^*)$ on message $w^* =$ $H(\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, R, pk_S, pk_R)$. If $F(d_{S^*}) \neq 0 \mod p$ or $K(w^*) \neq 0 \mod p$, then \mathcal{B} aborts. Otherwise, $F(d_{S^*}) =$ $0 \mod p$ and $K(w^*) = 0 \mod p$, \mathcal{B} computes

$$= \frac{\frac{\delta_5^*}{(\delta_4^*)^{J(d_S*)}(\delta_2^*)^{L(w^*)}}}{g_2^{ax_{S^*}^2}(u'\prod_{i=1}^{i=n}u_i^{d_{S^*,i}})^{t_{S^*}}(\delta_2^*)^{L(w^*)}(v'\prod_{j=1}^{j=m}v_j^{w_j})^s}{g^{J(d_{S^*})t_{S^*}}g^{L(w^*)r''}}$$

$$= g_2^{ax_2^2}$$

$$= g^{abx_{S^*}^2}.$$

 \mathcal{B} retrieves the secret value x_{S^*} s.t. pk_{S^*} and thus can output g^{ab} as the solution to the CDH problem instance.

Remark 3. Liu et al. showed their scheme is secure against an existential forgery for adaptive chosen message attacks (EUF-CLSC-CMA) if the user's public key is with the correctly form $e(g_1, g_2)^{x_{ID}}$ i.e the \mathcal{A}_I can replace public key only by choosing a different secret value x'_{ID} . However, in [15], Liu et al. scheme is showed that a Type I adversary \mathcal{A}_I can cheat the receiver and forge a valid signcrypted text by replacing the sender's public key $e(g,g)^{x'_{S}}$. The weakness in [14] is that the sender's public key pk_S is just a group element $e(g_1, g_2)^{x_S}$, and it can not check whether the public key pk_S is correctly formed during unsigncrypt stage. In order to defend against attacks [15], we revise **UserkeyGen** so that the receiver can check whether the sender's public key pk_S is correctly formed during the unsigncrypt stage. We omit the analysis of the success probability and the time complexity, which are similar to that of Liu et al. [14].

Lemma 4. The new CLSC scheme is existentially unforgeable against the Type II attacker in the standard model if the CDH assumption holds.

Proof. Assume that there exists a Type II forger \mathcal{A}_{II} against our scheme forger. In the following, we construct an algorithm \mathcal{B} to solve the CDH problem.

Suppose \mathcal{B} is given a random instance of the CDH problem (g^a, g^b) . Its goal is to output g^{ab} . The simulation process is the same as that described in Lemma 2.

Finally, if \mathcal{B} does not abort, the adversary \mathcal{A}_{II} returns a new ciphertex $\delta^* = (\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, \delta_5^*)$ on message $w^* = H(\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, R, pk_{S^*}, pk_{R^*})$ where w^* has never been queried. If $F(d_{S^*}) \neq 0 \mod p$ or $K(w^*) \neq 0 \mod p$, then \mathcal{B} aborts. Otherwise, $F(d_{S^*}) = 0 \mod p$ and $K(w^*) = 0 \mod p$, \mathcal{B} computes

$$= \frac{\frac{\delta_5^*}{(\delta_4^*)^{J(d_{S^*})}(\delta_2^*)^{L(w^*)}}}{g_2^{a\alpha x_{S^*}^2}(u'\prod_{i=1}^{i=n}u_i^{d_{S^*,i}})^{t_{S^*}}(\delta_2^*)^{L(w^*)}(v'\prod_{j=1}^{j=m}v_j^{w_j})^s}{g^{J(d_{S^*})t_{S^*}}g^{L(w^*)r''}}$$

= $g_2^{a\alpha x^2}$
= $g^{ab\alpha x_{S^*}^2}$.

Since \mathcal{B} has the value x_{S^*} and the master secret key α , it can output g^{ab} as the solution to the CDH problem instance.

Remark 4. Weng et al. [21] proved that a Type II adversary can use a ciphertext generated by a sender to arbitrarily forge signcryption on behalf of this sender. In our improved scheme, we embed a random binary string R into the signcryption ciphertex. Since \mathcal{A}_{II} does not know R, he can not successfully launch the same attacks as in [21].

5.2 Performance Analysis

The existing CLSC schemes without using random oracles are given in [10, 14]. However, there exists security weakness in these two schemes [10, 14], that is, we can not check whether the user's public key is correctly formed during signeryption and unsigneryption stages. To avoid the security weakness, we has to add verification equations which results in our improved scheme has more computational cost in the signcryption and unsigncryption stages. Due to adopting Naccache's methods [16] in our improved scheme, identity ID with $n' = n \cdot \ell$ bit length can be reduced to n dimensional vectors $d_{ID} =$ $(d_{ID,1},\ldots,d_{ID,n})$ where each $d_{ID,i}$ is an ℓ -bit integer. So the new scheme is with a smaller master public size than the other existing CLSC schemes [10, 14] The detailed comparisons of our scheme with these schemes [10, 14] are summarized in Table 1 where H denotes the Hash function computation, $E_{G_{T}}$ denotes an exponentiation computation in G_T , P denotes a pairing computation, and $aG \mid$ denotes the binary length of a elements in G.

Schemes	Public parameter size	Operations	Ciphertext size	Security
[10]	$ (m+n\cdot l+4)G $	$2H + 1E_{G_T} + 5P$	$ 4G + 1G_T $	No
[14]	$ (m+n\cdot l+4)G $	$2H + 1E_{G_T} + 5P$	$ 4G + 1G_T $	No
Our Scheme	(m+n+4)G	$2H + 1E_{G_T} + 15P$	$ 4G + 1G_T $	Yes

Table 1: Comparisons among different CLSC schemes

From Table 1, we know our scheme has more computational cost in the signcryption and unsigncryption stages, but it can provide the provable security and has the same ciphertext size as schemes [10, 14] and has smaller public parameter size than schemes [10, 14].

6 Conclusions

Liu et al. [14] proposed the first CLSC scheme in the standard model. However, their scheme has some security weaknesses [15, 17, 21]. In this paper, we propose a corrected version of Liu et al's scheme and prove the new scheme is secure against Type I and Type II (a maliciousbut-passive KGC) adversaries in the standard model. Our new scheme not only provides the provable security but also has smaller public parameter size than the previous schemes [10, 14].

Acknowledgments

We would like to thank the anonymous reviewers for giving valuable comments. This work is supported by NSFC (Grant Nos. 61272057, 61202434, 61170270, 61100203, 61003286, 61121061), the Fundamental Research Funds for the Central Universities (Grant No. 2012RC0612, 2011YB01).

References

- S. S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology* (Asiacrypt'03), pp. 452–473, 2003.
- [2] M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang, "Malicious KGC attacks in certificateless cryptography," in *Proceedings of the 2007 ACM Symposium on Information, Computer and Commu*nications Security, pp. 302–311, 2007.
- [3] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM Sym*posium on Information, Computer and Communications Security, pp. 369–372, 2008.
- [4] M. Bellare, A. Boldyreva, and A. Palacio, "An uninstantiable random-oracle-model scheme for a hybridencryption problem," in *Advances in Cryptology (Eurocrypt'04)*, pp. 171–188, 2004.

- [5] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the first ACM Conference on Computer and Communications Security*, pp. 62–73, 1993.
- [6] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *Journal of the ACM*, vol. 51, no. 4, pp. 557–594, 2004.
- [7] H. Chen, Y. Li, and J. Ren, "A practical identitybased signcryption scheme," *International Journal of Network Security*, vol. 15, no. 6, pp. 484–489, 2013.
- [8] A. W. Dent, B. Libert, and K. G. Paterson, "Certificateless encryption schemes strongly secure in the standard model," in *Proceedings of the 11th International Workshop on Practice and Theory in Public Key Cryptography*, pp. 344–359, 2008.
- [9] Y. H. Hwang, J. K. Liu, and S. S. M. Chow, "Certificateless public key encryption secure against malicious KGC attacks in the standard model," *Journal of Universal Computer Science*, vol. 14, no. 3, pp. 463–480, 2008.
- [10] Z. Jin, Q. Wen, and H. Zhang, "A supplement to Liu et al.'s certificateless signcryption scheme in the standard model," in *IACR Eprint Archive*, 2010.
- [11] F. Li, X. Xin, and Y. Hu, "Id-based signcryption scheme with (t,n) shared unsigncryption," *International Journal of Network Security*, vol. 3, no. 2, pp. 155–159, 2006.
- [12] P. Li, M. He, and W. Liu, "Efficient and provably secure certificateless signcryption from bilinear pairings," *Journal of Computational Information Systems*, vol. 6, no. 11, pp. 3643–3650, 2011.
- [13] J. K. Liu, M. H. Au, and W. Susilo, "Self-generatedcertificate public key cryptography and certificateless signature/encryption scheme in the standard model," in *Proceedings of the Second ACM Sympo*sium on Information, Computer and Communications Security, pp. 273–283, 2007.
- [14] Z. Liu, Y. Hu, X. Zhang, and H. Ma, "Certificateless signcryption scheme in the standard model," *Information Sciences*, vol. 180, no. 3, pp. 452–464, 2010.
- [15] S. Miao, F. Zhang, S. S. Li, and Y. Mu, "On security of a certificateless signcryption scheme," *Information Sciences*, vol. 232, pp. 475–481, 2013.
- [16] D. Naccache, "Secure and practical identity-based encryption," *Iet Information Security*, vol. 1, no. 2, pp. 59–64, 2007.
- [17] S. S. D. Selvi, S. S. Vivek, and C. P. Rangan, "Certificateless signcryption," in *Cryptology ePrint Archive*, *Report 2010/92*, 2010.

- [18] A. Shamir, signature schemes," in Advances in Cryptology (Crypto'84), pp. 47–53, 1984.
- [19] M. Toorani and A. Beheshti, "Cryptanalysis of an elliptic curve-based signcryption scheme," International Journal of Network Security, vol. 10, no. 1, pp. 51–56, 2010.
- [20] B. Waters, "Efficient identity-based encryption without random oracles," in Advances in Cryptology (Eurocrypt'05), pp. 114-127, 2005.
- [21] J. Weng, G. X. Yao, R. H. Deng, M. R. Chen, and X. X. Li, "Cryptanalysis of a certificateless signcryption scheme in the standard model," Information Sciences, vol. 181, no. 3, pp. 661-667, 2011.
- [22] C. Wu and Z. Chen, "A new efficient certificateless signcryption scheme," in The International Symposium on Information Science and Engieering, pp. 661-664, 2008.
- [23] H. Xiong, J. Hu, and Z. Chen, "Security flaw of an ECC-based signcryption scheme with anonymity," International Journal of Network Security, vol. 15, no. 4, pp. 317-320, 2013.
- [24] H. Xiong, Z. G. Qin, and F. G. Li, "Certificateless public key encryption secure against malicious KGC attacks in the standard model," Fundamenta Informaticae, vol. 88, pp. 193–206, 2008.
- [25] G. Yu, Z. Yang, S. Fan, Y. Shen, and W. Han, "Efficient certificateless signcryption scheme," in Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops, pp. 55-59, 2010.
- [26] Y. Zheng, G. Goos, J. Hartmanis, and J. V. Leeuwen, "Digital signcryption or how to achieve cost (signature and encryption) $\leq cost$ (signature) + cost(encryption)," in Advances in Cryptology (Crypto'97), pp. 291-312, 1997.

"Identity-based cryptosystems and Lin Cheng biography. He is currently a PhD candidate in State Key Laboratory of Networking and Switch Technology, Beijing University of Posts and Telecommunications. His present research interests are cryptography, information security and cloud computing..

> Qiaoyan Wen biography. She received her BS and MS from Shanxi normal University in 1981 and 1984, respectively, and her PhD from Xidian University in 1997. Now, she is a professor of Beijing University of Posts and Telecommunications. Her present research interests include cryptography, information security, and cloud computing.