Wireless Pers Commun (2018) 98:2693–2709
https://doi.org/10.1007/s11277-017-4995-4

An Enhanced Certificateless Signcryption in the Standard Model

Ming Luo[1] · **Yuwei Wan**[1]

Published online: 26 October 2017

Abstract Signcryption is a cryptography prototype which performs message encryption and signature in a logical step. Certificateless public key cryptography successfully resolves the problem of certificate management in traditional public key cryptography and key escrow problem in identity-based public key cryptography. There are lots of efficient certificateless signcryption schemes that have been proposed, most of which are proved secure under the random oracle model. But when applied in practical situations, the random oracle model will cause many security problems due to its own defects. Nowadays, more and more people pay attention to the standard model which provides a stronger security. In this paper, we present an efficient certificateless signcryption scheme that is provably secure in the standard model. Under the Decisional Bilinear Diffie–Hellman and Computational Diffie–Hellman hard problems, our scheme satisfies the ability of indistinguishability against adaptive chosen ciphertext attack and existential unforgeability against adaptive chosen message attack. Moreover, our scheme satisfies known session-specific temporary information security that most of signcryption schemes in the standard model cannot achieve this security attribute. Compared with other signcryption schemes, our scheme achieves shorter ciphertext length, better performance efficiency and stronger security.

Keywords Signcryption · Certificateless · Standard model · Bilinear pairing · KSSTIS

✉ Ming Luo
lmhappy21@163.com

Yuwei Wan
wvv1222@qq.com

[1] School of Software, Nanchang University, Nanchang 330047, JiangXi, People's Republic of China

1 Introduction

Public key cryptography (PKC) is an important milestone in the history of cryptography. It has experienced three important stages: the traditional public key cryptography, Identity-based public key cryptography (ID-PKC) and certificateless public key cryptography (CL-PKC). In an ID-PKC scheme, user's public key is constructed with his own identity information while the private key is generated by a trusted authority named Private Key Generator (PKG). Compared with traditional public key cryptography, ID-PKC simplifies the work of Public Key Infrastructure (PKI). However, since the private keys are all generated by PKG, ID-PKC has serious key escrow problem. PKG can decrypt all the messages transmitted between different users and forge any user's signature.

CL-PKC was first proposed by Al-Riyami and Paterson [1] in 2003. To avoid the key escrow problem, they divided user private key into two parts, one part is still generated by a trusted authority named Key Generation Center (KGC) and the other is set by user himself. Confidentiality and unforgeability are two important characteristics of cryptography, and the traditional implementation method is the "encrypt-then-signature" mechanism. Signcryption is a concept that encrypts and signs the plaintext simultaneously. This primitive can not only streamline the calculation process, but also save the communication overhead. Currently this concept in many network environments is widely used to design more efficient cryptographic schemes.

In 2008, the first certificateless signcryption scheme was proposed by Barbosa and Farshim [2] and proved to be secure under the random oracle model. After that, lots of certificateless signcryption schemes have been proposed [3, 4]. However, most of these schemes were proved secure under random oracle model which may engender serious security problems in real implementation. It has been shown that when random oracles are instantiated with concrete hash functions, the resulting scheme may not be secure [5, 6]. In recent years, the standard model has been widely concerned and used in the security analysis of many signcryption schemes. Under the standard model, schemes' security only relies on some hard problems, where these problems still have not been solved in mathematics, thus these schemes are more secure in practical use. Yu et al. [7] proposed the first identity-based signcryption scheme without random oracle model in 2009, but this scheme has some flaws in encryption design and indistinguishability proof. Based on Yu's scheme, Li et al. [8] made great improvements and proposed a more secure identity-based signcryption scheme with less computation. In 2010, Liu et al. [9] proposed the first efficient and provably secure certificateless signcryption scheme in the standard model. Unfortunately, Miao et al. [10] pointed out that scheme [9] cannot satisfy the confidentiality and unforgeability requirements. Meanwhile, Weng et al. [11] found that Liu's scheme cannot resist attacks from malicious-but-passive KGC. Jin et al. [12] optimized and enhanced Liu's scheme with a new idea, and they showed that their improvements were really secure in the standard model. But Xiong [13] showed that Jin's scheme cannot resist chosen ciphertext attacks, and was vulnerable to malicious-but-passive KGC. Then he utilized Bellare and Shoup's one-time signature and Li's signcryption scheme [8] to propose another certificateless signcryption scheme in the standard model. It is proved that Xiong's scheme meet the confidentiality and unforgeability requirements. In 2015, Cheng et al. [14] modified Liu's scheme to resist two types of adversaries attacks with less

computations. Recently, Zhou et al. [15] proposed a more efficient scheme than [13] and [14] schemes, but their scheme has too many pairing operations and the comparison contains the offline computation. The offline computation can be performed with the public system parameter and communication user's public key before the message to be signcrypted and the session-specific temporary information are given. Moreover, we find these certificateless signcryption schemes above in the standard model cannot satisfy known session-specific temporary information security (KSSTIS). Most of certificateless signcryption schemes use some randomized private temporary information to produce an encryption key and an encrypted ciphertext in each run of the signcryption stage. KSSTIS means if the compromise of this private input should not compromise the secrecy of the generated encryption key and the encrypted ciphertext).

In this paper, we proposed a provably secure certificateless signcryption scheme in the standard model. Under the Decisional Bilinear Diffie–Hellman and Computational Diffie–Hellman hard problems, our scheme satisfies the main security feature of indistinguishability against adaptive chosen ciphertext attack and existential unforgeability against adaptive chosen message attack. Also our scheme achieves the KSSTIS attribute. Compared with other similar schemes through efficiency analysis, our scheme has less exponentiation computation and pairing operation. The result is that our scheme has stronger security and needs less calculation costs for network communication.

This paper is organized by following parts: We provide some preliminaries in Sect. 2. In Sect. 3, we describe the formal model of certificateless signcryption scheme (CLSC), which includes the generic model and security model of CLSC. Section 4 describes our schemes in details. After that, the correctness and security analysis have been given in Sect. 5. Compared with other schemes by making security and efficiency analysis, we showed the performance of our scheme in Sect. 6. Section 7 is a conclusion of this paper.

2 Preliminaries

In this section, we firstly review some prior knowledge of bilinear maps and complexity assumptions. Let G_1, G_2 be two cyclic multiplicative groups with prime order p and let g be the generator of G_1. For a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$, it must meet the following characteristics:

1. Bilinear: for all $M, N \in G_1$ and $a, b \in Z_p^*$, $\hat{e}(M^a, N^b) = \hat{e}(M, N)^{ab}$;
2. Computability: for all $(M, N) \in G_1 \times G_1$, there is an efficient algorithm that can compute $\hat{e}(M, N)$;
3. Non-degenerate: $\hat{e}(g, g) \neq 1_{G_2}$.

The security of our scheme depends on the hardness of following problems:

Definition 1 Randomly select $a, b \in Z_p^*$, let g be a generator of G_1. For a group G_1, the Computational Diffie–Hellman (CDH) is to compute $g^{ab} \in G_1$ given a triple (g, g^a, g^b).

Definition 2 Randomly select $a, b, c, z \in Z_p^*$, let g be a generator of G_1. Decisional Bilinear Diffie–Hellman (DBDH) problem is to determine whether h_1 and h_2 is equal given two quintuples $h_1 = (g, g^a, g^b, g^c, \hat{e}(g, g)^{abc})$ and $h_2 = (g, g^a, g^b, g^c, \hat{e}(g, g)^z)$.

3 Formal Models of certificateless Signcryption Scheme

3.1 Generic Model

For a certificateless signcryption scheme (CLSC), it generally consists of the following five algorithms:

Setup Given a security parameter λ as an input, then KGC outputs system parameter *params* and master secret key α. The α is kept secretly by KGC and *params* will be published to all users.

User-Key-Generate (UKG) Given system parameter *params* and user identity *ID*, this algorithm is executed by the user himself to get user secret key x_{ID} and public key PK_{ID}.

Partial-Private-Key-Extraction (PPKE) Given system parameter *params*, master key α and user identity, this algorithm is executed by KGC to get user partial private key d_{ID}, which will be sent to corresponding user.

Signcrypt While input system parameter *params*, plaintext m, sender's private key pair (x_s, d_s), receiver's identity ID_r and public key PK_r, this algorithm is executed by the sender and outputs ciphertext $c = Signcrypt(params, m, x_s, d_s, ID_r, PK_r)$.

Unsigncrypt On input of system parameter *params*, ciphertext c, the receiver's private key pair (x_r, d_r), sender's identity ID_s and public key PK_s, the receiver firstly verifies the validity of the ciphertext c, if it is valid, the receiver recovers the plaintext $m = Unsigncrypt(params, c, x_r, d_r, ID_s, PK_s)$ and outputs plaintext m, otherwise he outputs character "$\perp$".

3.2 Security Model

For a secure CLSC scheme, it should have the indistinguishability against adaptive chosen ciphertext attack and unforgeability against adaptive chosen message attack. Definition 3 is for confidentiality of ciphertext and Definition 4 is for unforgeability of messages.

The certificateless signcryption system includes two types of attackers named A_I and A_{II}. For A_I, we treat him as an ordinary attacker that does not have the master secret key of KGC. A_I can replace any user's public key with a new public key PK'_u, and he does not need to provide the corresponding secret value of PK'_u. For A_{II}, we treat him as a malicious-but-passive KGC A_{II} attacker. As an attacker, A_{II} can generate KGC's master secret key maliciously but cannot replace user's public key.

Definition 3 *(Confidentiality)* For attacker $A_{i(i=I,II)}$, if he doesn't have non-negligible advantage to win the Game1 and Game2 within polynomial time respectively, we will say that the CLSC has the indistinguishability against adaptive chosen ciphertext attack (*IND-CLSC-CCA2*).

Game1

Setup Challenger C firstly runs the setup algorithm defined in generic model to generate system parameters, then sends attacker A_I the parameter *params* while secretly keeps the master secret key α.

Phase 1 This is a probing phase. During the simulation, attacker A_I can perform the following polynomial times queries.

1. *Public-key-generate query* A_I asks for user public key with identity ID_u. Then C runs the *UKG* algorithm $UKG(params, ID_u) \rightarrow (PK_u, x_u)$ and returns the result PK_u to A_I.

2. *Partial-private-key-extraction query* A_I asks for user partial private key with identity ID_u. Then C runs the *PPKE* algorithm $PPKE(params, \alpha, ID_u)$ and returns the result d_u to A_I.
3. *Corruption query* A_I asks for user secret key with identity ID_u. Then C runs the *UKG* algorithm $UKG(params, ID_u) \rightarrow (PK_u, x_u)$ and returns the result x_u to A_I.
4. *Public-key-replace (PKR) query* A_I can replace any user's public key with any data in the valid range.
5. *Signcrypt query* A_I submits two separate identities ID_i, ID_j to act as the sender and receiver, and along with plaintext m for Signcrypt query. C runs the *Signcrypt* algorithm $c = Signcrypt(params, m, x_i, d_i, ID_j, PK_j)$ and returns ciphertext c to A_I.
6. *Unsigncrypt query* A_I submits two separate identities ID_i, ID_j to act as the sender and receiver, and along with ciphertext c for a Unsigncrypt query. If the ciphertext is valid, then C runs the *Unsigncrypt* algorithm $m = Unsigncrypt(params, c, x_j, d_j, ID_i, PK_i)$ and sends plaintext m to A_I, otherwise outputs character "$\perp$".

Challenge A_I outputs two messages m_0 and m_1 with the same length, two identities ID_i^* and ID_j^* on which he wishes to be challenged. Either ID_i^* or ID_j^* cannot be the identity that has been used for PPKE query. C randomly selects $d \in \{0,1\}$ to compute $c^* = Signcrypt(params, m_d, x_i^*, d_i^*, ID_j^*, PK_j^*)$ and returns c to A_I.

Phase 2 A_I performs polynomial times queries as in phase 1. But A_I is forbidden to perform PPKE query on ID_i^* and ID_j^*. Meanwhile, A_I cannot perform Unsigncrypt query on ciphertext c^* under ID_i^* and ID_j^*.

Guess A_I outputs d' as a guess at the end. If $d' = d$, A_I wins the Game 1.

The advantage of A_I is defined to be $Adv_{IND-CCA2}(A_I) = |2\Pr[d' = d] - 1|$.

Game2

Setup Challenger C runs the Setup algorithm defined in generic model to generate system parameters and sends attacker A_{II} the parameter *params* and master secret key α.

Phase 1 During this phase, A_{II} performs bounded times of polynomial queries just like Game 1. Notes that A_{II} doesn't need to perform PPKE and Public-key-replace queries.

Challenge A_{II} outputs two messages m_0 and m_1 with the same length, two challenge identities ID_i^* and ID_j^*. The identity ID_j^* cannot have been performed by Corruption query. C randomly selects $d \in \{0,1\}$ and computes $c^* = Signcrypt(params, m_d, x_i^*, d_i^*, ID_j^*, PK_j^*)$, then returns ciphertext c^* back to A_{II}.

Phase 2 A_{II} performs polynomial queries just like in phase 1. But A_{II} is not allowed to perform Corruption query on ID_j^*. Meanwhile, A_I cannot perform Unsigncrypt query on ciphertext c^* under ID_i^* and ID_j^*.

Guess A_{II} also outputs d' as a guess at the end. If $d' = d$, A_{II} wins the Game 2.

The advantage of A_{II} is defined to be $Adv_{IND-CCA2}(A_{II}) = |2\Pr[d' = d] - 1|$.

Definition 4 *(Unforgeability)* For attacker $A_{i(i=I,II)}$, if he doesn't have non-negligible advantage to win the Game 3 and Game 4 within polynomial time respectively, we will say that the CLSC has the unforgeability against adaptive chosen message attack (*EUF-CLSC-CMA*).

Game3

Setup C runs the Setup algorithm defined in generic model to generate system parameters and sends attacker A_I the parameter *params* while secretly keeps the master secret key α.

Probing A_I performs bounded times polynomial queries just like Game 1 in Definition 3.

Forge A_I outputs a signcryption ciphertext c^* related to ID_i^*, ID_j^* and one message m^*. A_I cannot perform Signcrypt query on message m^* under ID_i^* and ID_j^* during the Probing phase. Similarly, ID_i^* cannot be an identity that has been performed by PKR query. In the end, if c^* is a valid signature for triple (m^*,ID_i^*,ID_j^*) and the result of *Unsigncrypt* algorithm is not character "$\perp$", we say A_I wins the game.

Game4
Setup C runs the Setup algorithm defined in generic model to generate system parameters and sends attacker A_{II} the parameter *params* and the master secret key α.
Probing A_{II} performs bounded times polynomial queries just like Game 2 in Definition 3.
Forge A_{II} outputs a signcryption ciphertext c^* related to ID_i^*, ID_j^* and one message m^*. Notes that A_{II} cannot perform Signcrypt query on message m^* under ID_i^* and ID_j^* during the Probing phase. Similarly, ID_j^* cannot be an identity that has been performed by Corruption query. In the end, if c^* is a valid signature for triple (m^*,ID_i^*,ID_j^*) and the result of *Unsigncrypt* algorithm is not character "$\perp$", we consider A_{II} wins the game.

4 Proposed CLSC Scheme

In this section, we will describe the details of the CLSC scheme that is provably secure in the standard model. The scheme consists of the following five algorithms: Setup, UKG (user key generate), PPKE (partial private key extraction), Signcrypt, and Unsigncrypt algorithms. Here, we assume the identity of all users is a string of length B_u.
Setup Randomly select a number v as a secure system parameter, the execution process of this algorithm is as follows:

1. Let G_1, G_2 be two cyclic multiplicative group of prime order p (p is a big prime).$\hat{e}$: $G_1 \times G_1 \to G_2$ is a bilinear mapping and let g be a generator of G_1.
2. Randomly select u', $m' \in G_1$ and two vectors $\vec{u} = (u_i)_{B_u}$, $\vec{m} = (m_j)_{B_m}$ with length of B_u and B_m respectively. Select an integer $\alpha \in Z_p^*$, and let $H : \{0,1\}^* \to \{0,1\}^{B_m}$ be a collision-resistant hash function (Similarly in schemes [6, 13, 14]).
3. Let α be the master secret key and define $P_{pub} = g^{\alpha}$. Then make public the system parameter $params = \{G_1, G_2,\hat{e}, p, g, P_{pub}, u', m',\vec{u},\vec{m}, H\}$.

UKG Given system parameter *params* and user identity $u \in \{0,1\}^{B_u}$, then user himself selects a random number $x_u \in Z_p^*$ and computes $PK_u = g^{x_u}$. So, he gets a tuple (x_u, PK_u), in which x_u is his secret key and PK_u is his public key. We assume that in a communication system, sender A's secret key/public key pair is (x_A, PK_A); receiver B's secret key/public key pair is (x_B, PK_B).
PPKE Given system parameter *params* and user identity $u \in \{0,1\}^{B_u}$. Let $U \subset \{1,2,\ldots,B_u\}$ be a set that for every $i \in U$ we have $u_i = 1$(i.e. the *i-th* bit of identity u equals bit 1). This algorithm outputs user partial private key $d_u = (u' \prod_{i \in U} u_i)^{\alpha}$. We define sender A's partial private key is $d_A = (u' \prod_{i \in U_A} u_i)^{\alpha}$, and receiver B's partial private key is $d_B = (u' \prod_{i \in U_B} u_i)^{\alpha}$.
Signcrypt User A selects a random number $k \in Z_p^*$ and computes $S = g^k$, then executes the following steps:

1. Encrypt message M:

$$R = M \cdot PK_B^{x_A} \cdot \hat{e}\left(d_A \cdot PK_B^k, u' \prod_{i \in U_B} u_i\right)$$
$$= M \cdot PK_B^{x_A} \cdot \hat{e}\left(d_A, u' \prod_{i \in U_B} u_i\right) \cdot \hat{e}\left(PK_B, u' \prod_{i \in U_B} u_i\right)^k$$

2. Compute signature. Firstly, A computes $m = H(R, S, u_A, u_B, PK_A, PK_B)$. Let $\bar{M} \subset \{1,2,\ldots,B_m\}$ be a set that for every $j \in \bar{M}$ we have $m[j] = 1$. Then A computes signature $T = d_A \cdot (m' \prod_{j \in \bar{M}} m_j)^{k+x_A}$.
3. Send the signcryption information $\sigma = (R, S, T)$ to user B.

Unsigncrypt User B adopts the verify-then-decrypt method.

1. Firstly, user B computes $m = H(R, S, u_A, u_B, PK_A, PK_B)$. Let $\bar{M} \subset \{1,2,\ldots,B_m\}$ be a set that for every $j \in \bar{M}$ and $m[j] = 1$.
2. Verification: B checks whether the following formula is valid:

$$\hat{e}(T, g) = \hat{e}\left(P_{pub}, u' \prod_{i \in U_A} u_i\right) \cdot \hat{e}\left(S, m' \prod_{j \in \bar{M}} m_j\right) \hat{e}\left(PK_A, m' \prod_{j \in \bar{M}} m_j\right)$$

   If the formula established, B proceeds to decrypt the ciphertext. Otherwise, B returns character "$\perp$".
3. Decrypt the ciphertext: $M = R / \left(PK_A^{x_B} \cdot \hat{e}\left(d_B, u' \prod_{i \in U_A} u_i\right) \cdot \hat{e}\left(S^{x_B}, u' \prod_{i \in U_B} u_i\right)\right)$.

Note: In order to improve computation efficiency, user's private key consists of two parts in our scheme, but there are three parts in Zhou's scheme [15], where the private key is used to encrypt and decrypt the message. At the unsigncryption stage, there is one exponentiation computation and one pairing operation related to the private key in our scheme, but one exponentiation computation and two pairing operations are related to the private key in Zhou's scheme. Moreover, our scheme selects one randomized private input, but Zhou's scheme chooses two random numbers, where the random values are used to generate the signcryption. At the signcryption stage, there are three exponentiation computation and three signcryption values related to the random value in our scheme, but six exponentiation computation and six signcryption values are related to the random numbers in Zhou's scheme.

5 Analysis of Proposed Scheme

5.1 Correctness

From signcryption phase to verification phase, and in decryption phase, all operations in our scheme are correct. The proof of correctness is as follows.

In the verification phase:

$$\begin{aligned}\hat{e}(T,g) &= \hat{e}\left(d_A \cdot \left(m' \prod_{j\in\vec{M}} m_j\right)^{k+x_A}, g\right)\\ &= \hat{e}\left(g, \left(u' \prod_{i\in U_A} u_i\right)^{\alpha} \cdot \left(m' \prod_{j\in\vec{M}} m_j\right)^{k+x_A}\right)\\ &= \hat{e}\left(P_{pub}, u' \prod_{i\in U_A} u_i\right) \cdot \hat{e}\left(S, m' \prod_{j\in\vec{M}} m_j\right) \cdot \hat{e}\left(PK_A, m' \prod_{j\in\vec{M}} m_j\right)\end{aligned}$$

In the decryption phase:

$$\begin{aligned}&S^{x_B} = (g^k)^{x_B} = (g^{x_B})^k = PK_B^k;\\ &R/\left(PK_A^{x_B} \cdot \hat{e}\left(d_B, u' \prod_{i\in U_A} u_i\right) \cdot \hat{e}\left(S^{x_B}, u' \prod_{i\in U_B} u_i\right)\right)\\ &= M \cdot PK_B^{x_A} \cdot \hat{e}\left(d_A, u' \prod_{i\in U_B} u_i\right)\\ &\cdot \hat{e}\left(PK_B^k, u' \prod_{i\in U_B} u_i\right) / \left(PK_A^{x_B} \cdot \hat{e}\left(d_B, u' \prod_{i\in U_A} u_i\right) \cdot \hat{e}\left(PK_B^k, u' \prod_{i\in U_B} u_i\right)\right)\\ &= M \cdot \hat{e}\left(d_A, u' \prod_{i\in U_B} u_i\right) / \hat{e}\left(d_B, u' \prod_{i\in U_A} u_i\right) = M\end{aligned}$$

5.2 Security Proof

5.2.1 Confidentiality

Theorem 1 *The proposed CLSC scheme is indistinguishable against the adversary A_I in the standard model assuming the DBDH problem is hard.*

Proof C receives a DBDH instance $(g, g^a, g^b, g^c, \hat{e}(g,g)^z)$, in which $a, b, c, z \in Z_p^*$, and he is required to distinguish $\hat{e}(g,g)^{abc} \stackrel{?}{=} \hat{e}(g,g)^z$. C treats A_I as a partner and replies the queries of A_I as follows. □

Setup

1. C selects four positive integers v_u, v_m, c_u and c_m, where $0 \le v_u \le B_u$, $0 \le v_m \le B_m$, $c_u (B_u + 1) < p$ and $c_m (B_m + 1) < p$.
2. C selects two elements x', y' from c_u, where the range of x' is $\{x_1,\ldots,x_{B_u}\}$ and the range of y' is $\{y_1,\ldots,y_{B_u}\}$. Similarly, C selects two elements e', f' from c_m, where the range of e' is $\{e_1,\ldots,e_{B_m}\}$ and the range of f' is $\{f_1,\ldots,f_{B_m}\}$.
3. Let $P_{pub} = g^a, u' = g^{y'+x'-c_u v_u}$, $m' = g^{f'+\varepsilon'-c_m v_m}$ and compute $u_i = g^{x_i+y_i}$, $m_j = -(1 \le i \le B_u,\ 1 \le j \le B_m)$. Let vectors $\vec{u} = (u_i)$ and $\vec{m} = (m_j)$.
4. C sends parameter *params* to A_I, where *params* $= \{g, P_{pub}, \vec{u}, \vec{m}, u', m'\}$.

5. C maintains three lists L_u, L_p and L_r to preserve relevant information. L_u is used to simulate *UKG* oracle, L_p is used to simulate *PPKE* oracle and L_r is used to simulate *PKR* oracle. All the lists are empty at the beginning.

In order to make the analysis of the simulation easier, we define several functions:

$$F_u = \sum_{i \in U} x_i - c_u v_u + x', \quad J_u = \sum_{i \in U} y_i + y'$$
$$K_m = \sum_{j \in \bar{M}} e_j - c_m v_m + e', \quad L_m = \sum_{j \in \bar{M}} f_j + f'$$

Then we have:

$$u' \prod_{i \in U} u_i = g^{F_u + F_u}, \quad m' \prod_{j \in \bar{M}} m_j = g^{K_m + L_m}$$

PKG Query

While input *params* and user identity ID_u, C randomly selects $x_u \in Z_p^*$ to compute $PK_u = g^{x_u}$, then preserves (ID_u, x_u, PK_u) in list L_u and sends PK_u to A_I.

PPKE Query

On input of *params* and user identity ID_u, C randomly selects i_b, i_c from {1,2,…,q_p} where q_p is the number of PPKE queries which can be performed. For the *i-th* query, C executes the following judgments:

1. If $i = i_b$, then let $ID_u = ID_B$, $u' \prod_{i \in U_B} u_i = g^b$ and $d_u = d_B = \perp$;
2. If $i = i_c$, then let $ID_u = ID_C$, $u' \prod_{i \in U_C} u_i = g^c$ and $d_u = d_C = \perp$;
3. For other PPKE queries, C computes user partial private key d_u as follows:

$$\begin{aligned} d_u &= \left(P_{pub} \cdot g^{-1}\right)^{F_u + J_u} \left(u' \prod_{i \in U} u_i\right) \\ &= \left(g^a \cdot g^{-1}\right)^{F_u + J_u} \cdot g^{F_u + J_u} \\ &= \left(g^a\right)^{F_u + J_u} \\ &= \left(u' \prod_{i \in U} u_i\right)^a \end{aligned}$$

In the above three cases, C sends d_u to A_I and preserves (ID_u, d_u) into list L_p.

PKR Query

A_I firstly performs PKG query with identity ID_u and gets a tuple (ID_u, x_u, PK_u). We assume that A_I can replace PK_u with any valid PK_u' and preserve (ID_u,$\perp$, PK_u') into L_r.

Corruption Query

While input *params* and user identity ID_u, A_I asks for the user secret key. C checks whether there is a matching item of (ID_u, x_u, PK_u) in L_u firstly. If exists, then C sends x_u to A_I. Otherwise, C performs PKG query to get and send x_u to A_I.

Signcrypt Query

Adversary A_I submits a Signcrypt query for plaintext M with user identity ID_1 and ID_2. Suppose that A_I has performed PKG query with ID_1 and ID_2 before this query. In this query, we consider two situations.

1. If $ID_1 \notin \{ID_B, ID_C\}$, C performs PPKE query and Corruption query with ID_1 firstly, then executes *Signcrypt* algorithm and sends the signcryption information $\sigma = (R, S, T)$ to A_I.
2. If $ID_1 \in \{ID_B, ID_C\}$, C returns character "$\perp$" and stops the challenge.

Unsigncrypt Query

Adversary A_I submits an Unsigncrypt query for ciphertext σ with user identity ID_1 and ID_2.We also consider two situations. Suppose that A_I has performed PKG query with ID_2 before this query. Then we have:

1. If $ID_2 \notin \{ID_B, ID_C\}$, C firstly verifies the validity of σ. If fails, he returns character "$\perp$". Otherwise, C performs PPKE query and Corruption query with ID_2 and executes *Unsigncrypt* algorithm, then returns message M to A_I.
2. If $ID_2 \in \{ID_B, ID_C\}$, C returns character "$\perp$" and stops the challenge.

All the processes above are just the first round of queries. At the end of this phase, A_I outputs two challenge identities ID_1^* and ID_2^*, two different messages M_0 and M_1 with equal length. If $ID_1^*, ID_2^* \notin \{ID_B, ID_C\}$, C stops the challenge. Otherwise, C selects a random number $k \in Z_p^*$ and $\omega \in \{0,1\}$, then computes $S^* = g^k$ and the ciphertext. C computes $R^* = M_\omega \cdot (PK_1^*)^{x_2} \hat{e}((PK_2^*)^k, u' \prod_{i \in U_2^*} u_i) \cdot u^*$, in which $u^* = \hat{e}(d_2^*, u' \prod_{i \in U_1^*} u_i) = \hat{e}((g^c)^a, g^b) = \hat{e}(g,g)^{abc}$ (u^* is the candidate answer for the DBDH problem), then computes $m^* = H(R^*, S^*, u_1^*, u_2^*, PK_1^*, PK_2^*)$ and $T^* = P_{pub} \cdot \left(S^* \cdot PK_1^*\right)^{K_m + L_m}$. Finally, C sends ciphertext $\sigma^* = (R^*, S^*, T^*)$ to A_I.

A_I performs the second round queries the same as the first round. After the simulation, A_I outputs ω' as a guess of ω. If $\omega' = \omega$, C outputs u^* as a solution to the DBDH problem. Otherwise, the DBDH problem cannot be resolved.

Theorem 2 *The proposed CLSC scheme is indistinguishable against the adversary A_{II} in the standard model assuming the DBDH problem is hard.*

Proof C receives a DBDH instance $(g, g^a, g^b, g^c, \hat{e}(g,g)^z)$, in which $a, b, c, z \in Z_p^*$, and he is required to distinguish $\hat{e}(g,g)^{abc} \stackrel{?}{=} \hat{e}(g,g)^z$. C treats A_{II} as a partner and replies the queries of A_{II} as follows. □

Setup Query

C does the same steps as the proof of Theorem 1 to get $params = \{g, \vec{u}, \vec{m}, u', m', P_{pub} = g^\alpha\}$. We also define $u' \prod_{i \in U} u_i = g^{F_u + J_u}$, $m' \prod_{j \in \bar{M}} m_j = g^{K_m + L_m}$. C maintains three lists L_u, L_p and L_r.

PKG Query

While input *params* and identity ID_u, C randomly selects i_b from $\{1,2,\ldots,q_p\}$, where q_p is the number of PKG query which can be performed. For the *i-th* query, C executes the following judgments:

1. If $i = i_b$, let $x_u = x_B = \perp$, $ID_u = ID_B$, $PK_u = PK_B = g^b$. Preserve $(ID_B, \perp, PK_B)$ into list L_u.
2. For other PKG queries, C randomly selects number $x_u \in Z_p^*$ to compute $PK_u = g^{x_u}$. Preserve (ID_u, x_u, PK_u) into list L_u.

C sends PK_u to A_{II} and updates L_u.

Corruption Query

On input of *params* and identity ID_u, if $ID_u = ID_B$, C terminates the challenge. Otherwise, C checks whether there is a matching item of (ID_u, x_u, PK_u) in L_u, If exists, C sends x_u to A_{II}. Otherwise, C performs PKG query to get and send x_u.

Signcrypt Query

Adversary A_{II} submits a Signcrypt query for plaintext M with user identity ID_1 and ID_2. Suppose that A_{II} has performed PKG query with ID_1 and ID_2 before this query. We consider two situations.

1. If $ID_1 \neq ID_B$, A_{II} performs Corruption query with ID_1 firstly, then executes *Signcrypt* algorithm and sends the signcryption information $\sigma = (R, S, T)$ to A_{II}.
2. If $ID_1 = ID_B$, C randomly selects a number $k \in Z_p^*$ to compute $S = g^k$, then computes $R = M \cdot \hat{e}(d_1 \cdot S^{x_2}, u' \prod_{i \in U_2} u_i)$, $m = H(R, S, u_1, u_2, PK_1, PK_2)$ and signature $T = d_1 \cdot (g^k)^{K_m + L_m} \cdot (PK_1)^{K_m + L_m}$. C returns signcryption information $\sigma = (R, S, T)$ to A_{II}.

Unsigncrypt Query

Adversary A_{II} submits a Unsigncrypt query for ciphertext σ with user identity ID_1 and ID_2. We also consider two situations. Suppose that A_{II} has performed PKG query with ID_1 and ID_2 before Unsigncrypt query. Then we have:

1. If $ID_2 \neq ID_B$, C verifies the validity of σ. If it is invalid, C returns character "$\perp$". Otherwise, C performs PPKE query and Corruption query with ID_2 and executes *Unsigncrypt* algorithm, then returns message M to A_{II}.
2. If $ID_2 = ID_B$, C returns character "$\perp$" and stop the challenge.

All the above processes are the first round of queries. At the end of this phase, A_{II} outputs two challenge identities ID_1^* and ID_2^*, two different messages M_0 and M_1 with equal length. If $ID_2^* \neq ID_B$, C stops the challenge. Otherwise, C randomly selects a number $k \in Z_p^*$ and $\omega \in \{0,1\}$, then computes $S^* = g^a$ and $u' \prod_{i \in U_2^*} u_i = g^c$, then computes the ciphertext $R^* = M_\omega \cdot (PK_2^*)^{x_1} \cdot \hat{e}(d_2^*, u' \prod_{i \in U_1^*} u_i) \cdot u^*$ in which $u^* = \hat{e}((S^*)^{x_2^*}, u' \prod_{i \in U_2^*} u_i) = \hat{e}(PK_2^*, u' \prod_{i \in U_2^*} u_i)^a = \hat{e}(g^b, g^c)^a = \hat{e}(g,g)^{abc}$ (u^* is the candidate answer for the DBDH problem), computes $m^* = H(R^*, S^*, u_1^*, u_2^*, PK_1^*, PK_2^*)$ and $T^* = d_1^* \cdot (S^* \cdot PK_1^*)^{K_m + L_m}$. Finally, C sends ciphertext $\sigma^* = (R^*, S^*, T^*)$ to adversary A_{II}.

A_{II} performs the second round queries just like the first round. After the simulation, A_{II} outputs ω' as a guess of ω. If $\omega' = \omega$, C outputs u^* as a solution to the DBDH problem. Otherwise, the DBDH problem is not resolved.

In terms of Theorems 1 and 2, we hold that there exist algorithms that can resolve the DBDH problem with non-negligible advantages if adversary $A_{i(i=I,II)}$ decrypts the signcryption by analyzing ciphertext. To date, no satisfactory algorithms have been found that resolve the DBDH problem in probabilistic polynomial time. So our scheme has the indistinguishability against adaptive chosen ciphertext attack.

5.2.2 Unforgeability

Theorem 3 *The proposed CLSC scheme is unforgeable against the adversary A_I in the standard model assuming that the CDH problem is hard.*

Proof Challenger C receives a CDH instance (g, g^a, g^b), where $a, b \in Z_p^*$, and he is required to compute g^{ab}. C treats A_I as a partner and replies the queries of A_I as follows.□

Setup Query

A_I does the same steps as the proof of Theorem 1 to get $params = \{g,\vec{u},\vec{m}, u', m', P_{pub} = g^{\alpha}\}$. Similarly, we define $u' \prod_{i\in U} u_i = g^{F_u+J_u}$ and $m' \prod_{j\in \bar{M}} m_j = g^{K_m+L_m}$. C also maintains three lists L_u, L_p and L_r. Let $m' \prod_{j\in \bar{M}} m_j = g^{K_m+L_m} = g^a$.

PKG Query, Corruption Query and PKR Query

The same as the proof of Theorem 1.

PPKE Query

On input of *params* and user identity ID_u, C executes the following calculation to get user partial private key d_u:

$$\begin{aligned} d_u &= \left(P_{pub} \cdot g^{-1}\right)^{F_u+J_u} \left(u' \prod_{i\in U} u_i\right) \\ &= \left(g^a \cdot g^{-1}\right)^{F_u+J_u} \cdot g^{F_u+J_u} \\ &= \left(g^a\right)^{F_u+J_u} \\ &= \left(u' \prod_{i\in U} u_i\right)^a \end{aligned}$$

C returns d_u to A_I and updates list L_p.

Signcrypt Query

A_I submits a Signcrypt query for plaintext M with user identity ID_1 and ID_2. Suppose that A_I has performed PKG query before this query. We can get ID_1's partial private key d_1 and secret key x_1 with his identity by PPKE query and Corruption query respectively. C randomly selects a number k to compute $S^* = g^k$, then runs the *Signcrypt* algorithm to get R and T, where $T = d_1 \cdot \left(m' \prod_{j\in \bar{M}} m_j\right)^{k+x_1}$. C finally sends $\sigma = (R, S, T)$ to A_I.

Unsigncrypt Query

Adversary A_I submits a Unsigncrypt query for ciphertext σ with user identity ID_1 and ID_2. During this phase, C firstly verifies the validity of σ. If it is invalid, C returns character "$\perp$". Otherwise, C performs PPKE query and Corruption query with ID_2, finally executes *Unsigncrypt* algorithm and returns message M to A_I.

After the query phase, A_I chooses message M^*, ID_1^* and ID_2^* to generate a signcryption message $\sigma^* = (R^*, S^*, T^*)$, where $S^* = g^b$. C executes the *Unsigncrypt* algorithm. If *Unsigncrypt*$(\sigma^*,x_2^*,d_2^*, ID_1^*,ID_2^*) \neq \perp$, C returns the decrypted message M^* to A_I. In the end, C computes and outputs the candidate answer $g^{ab} = T^* / \left[d_1^* \cdot \left(m' \prod_{j\in \bar{M}} m_j^*\right)^{x_1^*}\right] = T^* / (d_1^* \cdot g^{ax_1^*})$ to the CDH problem.

Theorem 4 *The proposed CLSC scheme is unforgeable against the adversary A_{II} in the standard model assuming the CDH problem is hard.*

Proof Challenger C receives a CDH instance (g, g^a, g^b), where $a, b \in Z_p^*$, and he is required to compute g^{ab}. C treats A_{II} as a partner and replies the queries of A_{II} as follows.□

Setup Query

The same as the proof of Theorem 3.

PKG Query

On input of *params* and identity ID_u, C randomly selects i_b from $\{1,2,\ldots,q_p\}$, where q_p is the number of PKG query which can be performed. For the *i-th* query, C executes the following judgments:

1. If $i = i_b$, let $x_u = x_B = \perp$, $ID_u = ID_B$, $PK_u = PK_B = g^b$. Preserve $(ID_B, \perp, PK_B)$ into list L_u.
2. For other PPKE queries, C randomly selects a number $x_u \in Z_p^*$ to compute $PK_u = g^{x_u}$ and preserves (ID_u, x_u, PK_u) into list L_u.

C sends PK_u to A_{II} and updates L_u.

Corruption Query

The same as the proof of Theorem 2.

Signcrypt Query

A_{II} submits a Signcrypt query for plaintext M with user identity ID_1 and ID_2. Suppose that A_{II} has performed PKG query with ID_1 and ID_2 before this query. If $ID_1 = ID_B$, C stops the challenge. Otherwise, C gets ID_1's secret key x_1 with his identity by Corruption query, then C randomly selects a number $k \in Z_p^*$ to compute $S = g^k$ and executes the *Signcrypt* algorithm to get R and T, where $T = d_1 \cdot (m' \prod_{j \in \bar{M}} m_j)^{k+x_1}$, finally C sends $\sigma = (R, S, T)$ to A_{II}.

Unsigncrypt Query

While input ID_1, ID_2 and σ, if $ID_2 = ID_B$, C stops the challenge. Otherwise, C verifies the validity of σ. If it is invalid, C returns character "$\perp$". Otherwise, C performs PPKE query and Corruption query with ID_2, and executes *Unsigncrypt* algorithm to return message M to A_{II}.

After the query phase, A_{II} chooses message M^*, ID_1^* and ID_2^* to generate a signcryption message $\sigma^* = (R^*, S^*, T^*)$. If $ID_2^* \neq ID_B$, C terminates the simulation. Otherwise, C executes the *Unsigncrypt* algorithm and returns the decrypted message M^* to A_{II}. Finally, C computes and outputs the candidate answer $g^{ab} = T^* / \left[d_1^* \cdot (m' \prod_{j \in \bar{M}} m_j^*)^k \right] = T^* / \left(d_1^* \cdot g^{ak} \right)$ to the CDH problem.

In terms of Theorems 3 and 4, we believe that there exist algorithms can resolve the CDH problem with non-negligible advantages if an adversary $A_{i(i=I,II)}$ generate a valid signcryption message σ^* by analyzing ciphertext. To date, no satisfactory algorithms have been found that resolve the CDH problem in probabilistic polynomial time. So our scheme has the unforgeability against adaptive chosen message attack.

5.2.3 KSSTIS

For our scheme, assuming that at the $j-th$ communication, the private temporary information k and signcryption $\sigma = (R, S, T)$ is leaked. For adversary A_I, he can not obtain the related information about private key (d_A, x_A) or (d_B, x_B). A_I cannot compute $\hat{e}(d_A, u' \prod_{i \in U_B} u_i)$ or $\hat{e}(d_B, u' \prod_{i \in U_A} u_i)$ under the assumption of CBDH problem and cannot compute $PK_B^{x_A}$ or $PK_A^{x_B}$ under the assumption of CDH problem. So, it is hard to obtain the message $M = R / PK_B^{x_A} \hat{e}(d_A, u' \prod_{i \in U_B} u_i) \cdot \hat{e}(PK_B^k, u' \prod_{i \in U_B} u_i)$ for A_I. For adversary A_{II}, in our scheme, he can obtain the partial private key d_A or d_B, and then he can compute $\hat{e}(d_A, u' \prod_{i \in U_B} u_i)$ or $\hat{e}(d_B, u' \prod_{i \in U_A} u_i)$. But A_{II} cannot compute $PK_B^{x_A}$ or $PK_A^{x_B}$ without x_A or x_B under the assumption of CDH problem. So, it is also hard to compute M for A_{II}. Hence,

our scheme can achieve the KSSTIS attribute. But in Xiong's scheme [13], when the adversary obtains the private temporary information k of the $j-th$ communication, he can obtain $\sigma_1 = m \cdot upk_{R,1}^{k}$ easily, and it is easy for the adversary to obtain the message $m = \sigma_1 / upk_{R,1}^{k}$ with the public key $upk_{R,1}$. The same situation happens in Cheng's scheme [14]. When the adversary obtains the private temporary information s of the $j-th$ communication, he can obtain $R_1 = \phi(M||R) \cdot \hat{e}(pk_{R,2}, pk_{R,3})^{s}$, $pk_{R,2}$ and $pk_{R,3}$ easily, and the message $M||R = \phi^{-1}(R_1/\hat{e}(pk_{R,2}, pk_{R,3})^{s})$ can be easily obtained. For Zhou's scheme [15], when the adversary obtains the ephemeral key $r_m, r_{m'}$ of $j-th$ communication, he can obtain $\sigma_1 = \phi(M||T) \cdot \hat{e}(UPK_{r,1}, UPK_{r,1})^{r_m} \cdot \hat{e}(g_1, g_1)^{r_{m'}}$, $UPK_{r,1}$ and g_1 easily, and the message $M||T = \phi^{-1}(\sigma_1/\hat{e}(UPK_{r,1}, UPK_{r,1})^{r_m} \cdot \hat{e}(g_1, g_1)^{r_{m'}})$ can be easily obtained.

5.2.4 Malicious-but-Passive KGC Attack

The malicious-but-passive KGC is malicious at the setup stage and may compute master public/secret key pair maliciously so that he can carry out the attack more easily [16]. The paper [16] pointed that certificateless schemes using the same key structure as [1] are not secure by giving this attack, where the key structure consists of the user private key $usk_{ID} = Q_{ID}^{sx}(Q_{ID} = H(ID)$, s is the system master secret key and x is the user secret key) and user public key $upk_{ID} = g^{sx}$ (g is a generator of G_1). The malicious-but-passive KGC selects a random number α and computes $g = Q_{ID}^{\alpha}$, then he can obtain the user private key $usk_{ID} = (upk_{ID})^{\alpha^{-1}}$. For our scheme, the key structure consists of the user private key $usk_U = (x_u, (u' \prod_{i \in U} u_i)^{\alpha})$($\alpha$ is the system master secret key and x_u is the user secret key) and user public key $upk_{ID} = g^{x_u}$. The malicious-but-passive KGC selects a random number ς and computes $g = (u' \prod_{i \in U} u_i)^{\varsigma}$, then he can obtain the value $(u' \prod_{i \in U} u_i)^{x_u}$, but he cannot computer the user secret key x_u, so our scheme can resist the malicious-but-passive KGC attack.

6 Performance Analysis

In recent years, many signcryption schemes have been proposed in the standard model. In this part, we analyze the efficiency and performance by comparing our scheme with schemes [13–15]. The efficiency of signcryption scheme in the standard model relies on the amount of exponentiation computation and the number of pairing operations. So, we mainly focused on these two operations. In addition, public key length, private key length and ciphertext length also are considered in this part. Let Exp_{G1}, Exp_{G2} denote the exponentiation computation in group G_1, G_2 respectively, where the computation cost of Exp_{G1} is the same as that of Exp_{G2}. Let Pi denotes the pairing operation. $|G_1|$ and $|G_2|$ denote the length of a group element. The comparisons are shown in Table 1.

We can see from Table 1 that our scheme is more efficient in terms of public key length, private key length and ciphertext length. What's the most important, we use less exponentiation computation and pairing operation. It should be noted that we put more calculations in offline phase, for instance, we can calculate $\hat{e}(P_{pub}, u' \prod_{i \in U} u_i)$ in advance and store it together with system parameters, which can reduce the calculation costs in communication phase. In addition to efficiency improvement, our scheme also enhances the security, since the scheme achieves confidentiality, unforgeability and KSSTIS attribute.

Table 1 Comparisons of different schemes in the standard model

Scheme	KSSTIS	Computation cost			Public key length	Private key length	Ciphertext length
		Signcryption	Unsigncryption	Offline computation			
[13]	N	$5Exp_{G_1}$	$1Exp_{G_1} + 5Pi$	$4Exp_{G_1} + 2Exp_{G_2} + 8Pi$	$\|2G_1\| + \|1G_2\|$	$\|2G_1\| + \|2Z_p\|$	$\|4G_1\| + \|1G_2\|$
[14]	N	$3Exp_{G_1} + 1Exp_{G_2}$	$5Pi$	$10Pi$	$\|3G_1\|$	$\|2G_1\| + \|1Z_p\|$	$\|4G_1\| + \|1G_2\|$
[15]	N	$3Exp_{G_1} + 3Exp_{G_2}$	$2Exp_{G_1} + 5Pi$	$2Exp_{G_1} + 4Pi$	$\|2G_1\|$	$\|2G_1\| + \|1Z_p\|$	$\|4G_1\| + \|2G_2\|$
Our scheme	Y	$2Exp_{G_1} + 1Exp_{G_2}$	$1Exp_{G_1} + 3Pi$	$2Exp_{G_1} + 5Pi$	$\|1G_1\|$	$\|1G_1\| + \|1Z_p\|$	$\|2G_1\| + \|1G_2\|$

7 Conclusions

In this paper, we have proposed a provably secure certificateless signcryption in the standard model. With the help of DBDH and CDH hard problems, our scheme has the ability of indistinguishability against adaptive chosen ciphertext attack and existential unforgeability against adaptive chosen message attack. Moreover, our scheme satisfies known session-specific temporary information security, which most of signcryption schemes in the standard model cannot achieve this security attribute. Performance analysis shows that our scheme is more efficient than other schemes in terms of computation and security. Thus, our scheme is more suitable for resource-constrained wireless communication networks.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grant (Nos. 61662046 and 61601215); and the research project of Jiangxi Province under Grant (Nos. 20171BCB23014 and 20142BBE50019).

References

1. Al-Riyami, S. S., & Paterson, K.G. (2003). Certificateless public key cryptography. In *Advances in cryptography-ASIACRYPT 2003* (pp. 452–473). Berlin, Germany.
2. Barbosa, M., & Farshim, P. (2008). Certificateless signcryption. *Cryptology ePrint Archive*, Retrieved from http://eprint.iacr.org/2008/143.pdf.
3. Luo, M., Wang, S. Q., & Hu, J. (2016). A more efficient and secure broadcast signcryption scheme using certificateless public-key cryptography for resource-constrained networks. *Journal of Internet Technology, 17*(1), 81–89.
4. Zhou, Y. W., Yang, B., & Zhang, W. J. (2016). Provably secure and efficient leakage-resilient certificateless signcryption scheme without bilinear pairing. *Discrete Applied Mathematics, 204,* 185–202.
5. Canetti, R., Goldreich, O., & Halevi, S. (2004). The random oracle methodology, revisited. *Journal of the ACM, 51*(4), 557–594.
6. Paterson, K. G., & Schuldt, J. C. N. (2006). Efficient identity-based signatures secure in the standard model. In *IACR ePrint Archive*, http://eprint.iacr.org/2006/080
7. Yu, Y., Yang, B., Sun, Y., & Zhu, S. (2009). Identity based signcryption scheme without random oracles. *Computer Standards & Interfaces, 31*(1), 56–62.
8. Li, X., Qian, H., Weng, J., & Yu, Y. (2013). Fully secure identity-based signcryption scheme with shorter signcryptext in the standard model. *Mathematical and Computer Modelling, 57*(3), 503–511.
9. Liu, Z., Hu, Y., Zhang, X., & Ma, H. (2010). Certificateless signcryption scheme in the standard model. *Information Sciences, 180*(3), 452–464.
10. Miao, S., Zhang, F., Li, S., & Mu, Y. (2013). On security of a certificateless signcryption scheme. *Information Sciences, 232,* 475–481.
11. Weng, J., Yao, G., Deng, R. H., & Li, X. (2011). Cryptanalysis of a certificateless signcryption scheme in the standard model. *Information Sciences, 181*(3), 661–667.
12. Jin, Z., Wen, Q., & Zhang, H. (2010). A supplement to Liu et al.'s certificateless signcryption scheme in the standard model. *Cryptology ePrint Archive*, Retrieved from http://eprint.iacr.org/2010/252.pdf.
13. Xiong, H. (2014). Toward Certificateless Signcryption Scheme Without Random Oracles. *Cryptology ePrint Archive*, Retrieved from: http://eprint.iacr.org/2014/162.pdf.
14. Cheng, L., & Wen, Q. Y. (2015). An improved certificateless signcryption in the standard model. *International Journal of Network Security, 17*(5), 597–606.
15. Zhou, C. X., Gao, G. Y., & Cui, Z. M. (2017). Certificateless signcryption in the standard model. *Wireless Personal Communications, 92*(2), 495–513.
16. Au, M. H., Chen, J., Liu, J. K., Mu, Y., Wong, D. S., & Yang, G. M. (2007). Malicious KGC attacks in certificateless cryptography. In *Proceedings of ASIACCS'2007* (pp. 302–311). New York: ACM.

Ming Luo received the B.E. and Ph.D. degree from Northeastern University, Shenyang, China in 2004 and 2010, respectively. Now he has been served as an associate professor in the School of Software, Nanchang University, Nanchang, China. He has won lots of scholarships in China and was supported by the National Natural Science Foundation of China, the National High-Tech Research and Development Plan of China and the Science and Technology Supporting Program of Jiangxi Province. His research interests are networks security, information security and cryptography.

Yuwei Wan received the B.E. degree form the College of software, Nanchang University in July 2014. She is currently pursuing her M.E degree from the College of software, Nanchang University. Her current research interests include information security and cryptography.