Cryptanalysis of Certificateless Signcryption Schemes and an Efficient Construction Without Pairing

S. Sharmila Deva Selvi, S. Sree Vivek * and C. Pandu Rangan*

Theoretical Computer Science Lab,
Department of Computer Science and Engineering,
Indian Institute of Technology Madras, India.
{sharmila,svivek,prangan}@cse.iitm.ac.in

Abstract. Certificateless cryptography introduced by Al-Riyami and Paterson eliminates the key escrow problem inherent in identity based cryptosystems. Even though building practical identity based signcryption schemes without bilinear pairing are considered to be almost impossible, it will be interesting to explore possibilities of constructing such systems in other settings like certificateless cryptography. Often for practical systems, bilinear pairings are considered to induce computational overhead. Signcryption is a powerful primitive that offers both confidentiality and authenticity to noteworthy messages. Though some prior attempts were made for designing certificateless signcryption schemes, almost all the known ones have security weaknesses. Specifically, in this paper we demonstrate the security weakness of the schemes in [4], [2] and [14]. We also present the first provably secure certificateless signcryption scheme without bilinear pairing and prove it in the random oracle model.

Keywords. Certificateless Signcryption, Provable Security, Pairing-free Cryptosystem, Random Oracle model, Cryptanalysis.

1 Introduction

Traditional public key infrastructure (PKI) based cryptosystems allow any user to choose their own private key and the corresponding public key. The public key is submitted to a certification authority (CA), which verifies the users identity and issues a certificate linking the users identity and the public key. Thus, PKI based systems need digital certificate management that is too cumbersome to maintain. Shamir [11] introduced the notion of identity based cryptography (IBC) to reduce the burden on the CA. In IBC, the private key of a user is not chosen by the user, rather it is issued by a trusted authority called the private key generator (PKG) or the trust authority (TA) and the public keys are generated by arbitrary strings representing the users identities and thus avoiding the need for certificates altogether. IBC suffers from an inherent issue called the key escrow problem, i.e. since the PKG is responsible for the generation of the private keys of all the users in the system, it has the ability to recover confidential information meant for any user or sign instead of a legitimate user. Certificateless cryptography (CLC) was introduced by Al-Riyami and Paterson [1] to address the key escrow problem, while avoiding the use of certificates and the need for a CA. The principle behind CLC is to partition private keys into two components: an identity based partial private key (known to the PKG) and a non-certified private key (which is unknown to the PKG). This technique efficiently combines the best features of IBC and PKI. A number of certificateless encryption and signature schemes derived from identity based encryption and signature schemes have been successfully constructed and were proven secure under various assumptions.

Signcryption which was proposed by Zheng [15] is a cryptographic primitive that provides authentication and confidentiality simultaneously, at a lower computational cost and communication overhead than signing and encrypting the message independently. A secure signcryption scheme should provide confidentiality, authentication, non-repudiation and should provide insider security too, i.e. even if the sender's private key is compromised, an adversary should not be able to unsigncrypt the message and even with the receiver's private key, a forger should not be able to generate a fresh signcryption (As if generated by the same sender).

* Work supported by Project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation sponsored by Department of Information Technology, Government of India

All the initial constructs for certificateless cryptosystem were based on bilinear pairing [3, 6, 12, 13, 8, 9]. The first certificateless cryptosystem without using bilinear pairing was proposed in the context of encryption by Beak et al. [3]. In general, certificateless cryptosystem is prone to key replacement attack because the public keys are not certified and anyone can replace the public key of any legitimate user in the system. The challenging task in the design of certificateless cryptosystem is to come up with a scheme which is secure even if the public key of the user is replaced. The excellent survey by Dent [7] gives a comprehensive overview of the design of provably secure certificateless encryption schemes.

To the best of our knowledge, there exist four ([4], [2], [14] and [5]) certificateless signcryption schemes (CLSC) in the literature. Among these four, [4], [2] and [14] are pairing based and [5] uses pairing for public key verification alone. In this paper, we show the security weaknesses in [4], [2] and [14]. We also present a provably secure certificateless signcryption scheme without pairing. Our scheme is the first provably secure certificateless signcryption scheme without pairing. The newly proposed CLSC scheme uses a key construct similar to that of [13] but uses a completely different approach for encryption. Any signcryption scheme is strongly secure if attacks by the insider is considered. Our security model considers insider security and we have proved the security of our scheme in the random oracle model. It is to be noted that signcryption schemes are not directly obtained by combining a digital signature scheme and an encryption schemes. The security requirements for signcryption schemes are entirely different from encryption and digital signatures. The notion of insider security comes into picture when we talk about signcryption. This is because the private information of the sender and the public information of the receiver is involved in signcryption schemes. Thus, the certificateless signcryption scheme presented here is not a trivial extension of a signature scheme clubbed with an encryption scheme.

2 Preliminaries

In this section, we give the definition for the computational assumptions, which we have used to prove our scheme.

A. Discrete Logarithm Problem (DLP)) Let p, q be two primes such that $q|(p-1)$, given $\langle g, g^a\rangle \in \{\mathbb{Z}_p^*\}^2$ for unknown $a \in \mathbb{Z}_q^*$, the DL problem in $\mathbb{Z}_p^*$ is to find a.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the DL problem in $\mathbb{Z}_p^*$ is defined as

$$Adv_{\mathcal{A}}^{DL} = Pr\left[\mathcal{A}(g, g^a) = a \mid a \in \mathbb{Z}_q^*\right]$$

The *DL Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{DL}$ is negligibly small.

B. Computational Diffie-Hellman Problem (CDHP) Given $\langle g, g^a, g^b\rangle \in \{\mathbb{Z}_p^*\}^3$ for unknown $a, b \in \mathbb{Z}_q^*$, the CDH problem in $\mathbb{Z}_p^*$ is to compute g^{ab}.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{Z}_p^*$ is defined as

$$Adv_{\mathcal{A}}^{CDH} = Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab} \mid a, b \in \mathbb{Z}_q^*\right]$$

The *CDH Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{CDH}$ is negligibly small.

2.1 Framework of CLSC

A certificateless signcryption scheme is defined by the following seven probabilistic polynomial-time algorithms:

- **CLSC.Setup:** This algorithm takes the security parameter 1^κ as input and outputs the master private key *msk*, the master public key *mpk* and the system public parameters *params*. It is run by the KGC in to initialize the system.

- **CLSC.PartialPrivateKeyExtract:** This algorithm takes the master public key mpk, the master private key msk and an identity $ID_A \in \{0,1\}^*$ of a user U_A as input. It outputs the partial private key d_A of U_A. This algorithm is run by the KGC once for each user and the corresponding partial private key is sent to U_A through a secure and authenticated channel.
- **CLSC.SetSecretValue:** This algorithm is run by every user independently and the value produced is called the secret value. Specifically, when user U_A runs the algorithm, the output generated is denoted as y_A and y_A is maintained as a secret value by U_A. (Note that y_A is not known to the KGC)
- **CLSC.SetPrivateKey:** This algorithm is run by the user U_A. It takes the master public key mpk, the user identity ID_A, partial private key d_A and the secret value y_A as input and produces the full private key s_A for U_A. This algorithm is run once by each user.
- **CLSC.SetPublicKey:** This algorithm is run by the user U_A. It takes the master public key mpk, the user's identity ID_A, the partial private key d_A and the secret value y_A as input and outputs the public key PK_A of U_A. It is run once by the user and the resulting public key is widely and freely distributed.
- **CLSC.Signcrypt:** In order to generate a signcryption of message m for the receiver U_B, the sender U_A provides the system public parameters *params*, sender identity ID_A, receiver identity ID_B, the public keys PK_A and PK_B of the sender and the receiver, the sender's full private key s_A and the message $m \in \mathcal{M}$ as input to this algorithm. The output is a signcryption $c \in \mathcal{CT}$. (Note that $\mathcal{M}$ is the message space and $\mathcal{CT}$ is the ciphertext space).
- **CLSC.Unsigncrypt:** The user U_B provides the system public parameters *params*, the sender identity ID_A, public key PK_A, the receiver identity ID_B, public key PK_B and the full private key s_B along with the signcryption $c \in \mathcal{CT}$ as input to this algorithm. The algorithm returns the message $m \in \mathcal{M}$, if c is a valid signcryption of m from U_A to U_B. The algorithm outputs "*Invalid*", otherwise.

2.2 Security Model of CLSC

The confidentiality proof of any CLSC scheme can be viewed as an interactive game, namely IND-CLSC-CCA2 between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. Si,ilarly, the unforgeability proof of CLSC can be viewed as an interactive game namely EUF-CLSC-CMA, between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$. In both the IND-CLSC-CCA2 and EUF-CLSC-CMA games, $\mathcal{A}$ and $\mathcal{F}$ are given access to some or all of the following six oracles (depending on their type). These oracles are simulated by $\mathcal{C}$:

- **Partial Private Key Extract of ID_A:** $\mathcal{C}$ responds by returning the partial private key d_A of the user U_A to $\mathcal{A}$.
- **Request Secret Value of ID_A:** If U_A's public key has not been replaced by $\mathcal{A}$ then $\mathcal{C}$ returns the user secret value y_A to $\mathcal{A}$. If U_A's public key was replaced by $\mathcal{A}$, then $\mathcal{C}$ returns nothing to $\mathcal{A}$.
- **Request Public Key of ID_A:** $\mathcal{C}$ responds by returning the current public key PK_A of user U_A to $\mathcal{A}$. (Because public keys are viable to change, $\mathcal{C}$ returns the current public key it has stored.)
- **Replace Public Key of ID_A:** The public key PK_A for a user U_A can be replaced with any value PK'_A provided by $\mathcal{A}$. On getting PK'_A from $\mathcal{A}$, $\mathcal{C}$ replaces the public key PK_A of ID_A with PK'_A. At any given time the current value of the user's public key is used by $\mathcal{C}$ in its computations or responses.
- **Signcryption of message m with ID_A as sender and ID_B as receiver:** $\mathcal{C}$ responds with the signcryption c on message m with ID_A as the sender and ID_B as the receiver. Note that even if $\mathcal{C}$ does not know the sender's private key, $\mathcal{C}$ should be able to produce a valid ciphertext and this is a strong property of the security model also $\mathcal{C}$ uses the current public keys of ID_A as well as ID_B to perform the signcryption.
- **Unsigncryption of ciphertext c with ID_A as sender and ID_B as receiver:** An unsigncryption query for ciphertext c and user U_A as the sender and U_B as the receiver is answered by $\mathcal{C}$, by first decrypting c and then returning the corresponding message m. $\mathcal{C}$ should be able to properly unsigncrypt ciphertexts, even for those users whose public keys have been replaced or if the receiver private key is not known to $\mathcal{C}$. This is a strong requirement of the security model. (Note that, $\mathcal{C}$ may not know the correct private key of the user whose public key is replaced. Still $\mathcal{C}$ can unsigncypt c by getting the corresponding secret value from $\mathcal{A}$.)

For any certificateless signcryption scheme two types of attacks are possible. They are referred as Type-I and Type-II attacks in the literature. Under each type of attack, it is required to establish the confidentiality and unforgeability of the scheme. The attack by a third party, (i.e. anyone except the legitimate receiver or the KGC) who is trying to break the security of the system is modeled by Type-I attack. The confidentiality of CLSC under Type-I attack is established through an interactive game between the adversary $\mathcal{A}_I$ and the challenger $\mathcal{C}$, and an interactive game between the forger $\mathcal{F}_I$ and the challenger $\mathcal{C}$ establishes the unforgeability under Type-I attack. The attack by a honest-but-curious KGC, who tries to break the security of the scheme is modelled by a Type-II attack. The confidentiality of the scheme under Type-II attack is established through an interactive game between the adversary $\mathcal{A}_{II}$ and the challenger $\mathcal{C}$. The unforgeability under Type-II attack is established through the game between the forger $\mathcal{F}_{II}$ and the challenger $\mathcal{C}$.

Summary of Constraints: In summary, the security model distinguishes the two types of adversaries (resp. forgers), namely Type-I and Type-II with the following constraints.

– Type-I adversary $\mathcal{A}_I$ (resp. forger $\mathcal{F}_I$) is allowed to replace the public keys of users at will but does not have access to the master private key *msk*.
– Type-II adversary $\mathcal{A}_{II}$ (resp. forger $\mathcal{F}_{II}$) is equipped with the master private key *msk* but is not allowed to replace public keys of any of the users.

Confidentiality: The security model to prove the confidentiality of a CLSC scheme with respect to Type-I adversary $\mathcal{A}_I$ (IND-CLSC-CCA2-I) and Type-II adversary $\mathcal{A}_{II}$ (IND-CLSC-CCA2-II) are given below:

IND-CLSC-CCA2-I game for Type-I Adversary: A certificateless signcryption (CLSC) scheme is IND-CLSC-CCA2-I secure if no probabilistic polynomial time adversary $\mathcal{A}_I$ has non-negligible advantage in winning the IND-CLSC-CCA2-I game. $\mathcal{A}_I$ is given access to all the six oracles defined above. It is to be noted that $\mathcal{A}_I$ does not have access to the master private key *msk*. IND-CLSC-CCA2-I game played between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_I$ is defined below:

Setup: The challenger $\mathcal{C}$ runs the setup algorithm to generate the system public parameters *params* and the master private key *msk*. $\mathcal{C}$ gives *params* to $\mathcal{A}_I$ while keeping *msk* secret. $\mathcal{A}_I$ interacts with $\mathcal{C}$ in two phases:

Phase I: $\mathcal{A}_I$ is given access to all the six oracles described above. $\mathcal{A}_I$ adaptively queries (adaptively means the current query may depend on the responses to the previous queries) the oracles consistent with the conditions for Type-I adversary (Described in the ***Summary of Constraints*** above).

Challenge: $\mathcal{A}_I$ generates two messages m_0, m_1 of equal length, an arbitrary sender identity ID_A and a receiver identity ID_B, which satisfies the following constraints.

– $\mathcal{A}_I$ can access the full private key of the sender ID_A.
– $\mathcal{A}_I$ has not queried the **Partial Private Key** corresponding to the receiver ID_B.

$\mathcal{A}_I$ sends m_0, m_1, ID_A and ID_B to $\mathcal{C}$. $\mathcal{C}$ randomly chooses a bit $b \in_R \{0, 1\}$ and computes a signcryption c^* with ID_A as the sender and ID_B as the receiver. Now, c^* is sent to $\mathcal{A}_I$ as the challenge signcryption.

Phase II: $\mathcal{A}_I$ adaptively queries the oracles consistent with the constraints that $\mathcal{A}_I$ should not query the partial private key of ID_B and $\mathcal{A}_I$ should not query for the *Unsigncryption* on c^* with ID_A as sender and ID_B as receiver.

Guess: $\mathcal{A}_I$ outputs a bit b' at the end of the game. $\mathcal{A}_I$ wins the IND-CLSC-CCA2-I game if $b' = b$. The advantage of $\mathcal{A}_I$ is defined as:

$$Adv_{\mathcal{A}_I}^{IND-CLSC-CCA2-I} = |2Pr[b = b'] - 1|$$

IND-CLSC-CCA2-II game for Type-II Adversary: A certificateless signcryption scheme (CLSC) is IND-CLSC-CCA2-II secure if no probabilistic polynomial time adversary $\mathcal{A}_{II}$ has non-negligible advantage in winning the IND-CLSC-CCA2-II game. $\mathcal{A}_{II}$ is given access to all the six oracles. The IND-CLSC-CCA2-II game played between $\mathcal{C}$ and the adversary $\mathcal{A}_{II}$ is defined below:

Setup: The challenger $\mathcal{C}$ runs the setup algorithm to generate the system public parameters *params* and the master private key *msk*. $\mathcal{C}$ gives both *params* and *msk* to $\mathcal{A}_{II}$. $\mathcal{C}$ interacts with $\mathcal{A}_{II}$ in two phases:

Phase I: This phase is similar to Type-I confidentiality game IND-CLSC-CMA-I.
Challenge: Same as Type-I but with the restrictions that:

1. $\mathcal{A}_{II}$ should not have queried the private key of the receiver ID_B in Phase I.
2. $\mathcal{A}_{II}$ has not replaced public key of ID_B in Phase I.

Phase II: Same as Type-I but with the restrictions that,

- $\mathcal{A}_{II}$ cannot extract the private key of ID_B.
- $\mathcal{A}_{II}$ should not replace the receiver ID_B's public key.
- Unsigncryption query on $\langle c^*, ID_A, ID_B \rangle$ is not allowed.

Guess: Same as Type-I confidentiality game IND-CLSC-CMA-I.
The advantage of $\mathcal{A}_{II}$ is defined as:

$$Adv_{\mathcal{A}_{II}}^{IND-CLSC-CCA2-II} = |2Pr[b = b'] - 1|$$

<u>**Unforgeability:**</u> The security model to prove the unforgeability of a CLSC scheme with respect to Type-I forger $\mathcal{F}_I$ (EUF-CLSC-CMA-I) and Type-II forger $\mathcal{F}_{II}$ (EUF-CLSC-CMA-II) are given below:

EUF-CLSC-CMA-I game for Type-I Forger: A certificateless signcryption scheme CLSC is Type-I, EUF-CLSC-CMA secure if no probabilistic polynomial-time forger $\mathcal{F}_I$ has non-negligible advantage in winning the EUF-CLSC-CMA-I game. A Type-I forger $\mathcal{F}_I$ is given access to all the six oracles defined above. The EUF-CLSC-CMA-I game played between the challenger $\mathcal{C}$ and the forger $\mathcal{F}_I$ is defined below:

Setup: $\mathcal{C}$ runs the setup algorithm to generate the master private key msk and public parameters $params$. $\mathcal{C}$ gives $params$ to $\mathcal{F}_I$ while keeping msk secret.

Training Phase: $\mathcal{F}_I$ is given access to all the six oracles. $\mathcal{F}_I$ adaptively queries the oracles consistent with the constraints for Type-I forger (Stated in the ***Summary of Constraints***).

Forgery: $\mathcal{F}_I$ outputs a signcryption c^* and a sender identity ID_A, for which $\mathcal{F}_I$ has not queried the partial private key. $\mathcal{F}_I$ wins the EUF-CLSC-CMA-I game if c^* is a valid signcryption with ID_A as the sender and ID_B as the receiver, also c^* was not the output of any signcrypt query on the corresponding message m with ID_A as the sender and ID_B as the receiver.

EUF-CLSC-CMA-II game for Type-II Forger: A certificateless signcryption scheme is Type-II, EUF-CLSC-CMA secure if no probabilistic polynomial-time forger $\mathcal{F}_{II}$ has non-negligible advantage in winning the EUF-CLSC-CMA-II game. A Type-II forger is given access to all the six oracles. EUF-CLSC-CMA-II game played between the challenger $\mathcal{C}$ and the forger $\mathcal{F}_{II}$ is same as EUF-CLSC-CMA-I with the constraints for Type-II Forger (Stated in the ***Summary of Constraints***).

3 Certificateless Signcryption Scheme of Barbosa et al.

In this section, we give the review and attack of the certificateless signcryption scheme by Barbosa et al. [4].

3.1 Review of Barbosa et al. Certificateless signcryption scheme

This scheme uses a symmetric bilinear group description $\varGamma$ which is defined with two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of same order q and an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. The four cryptographic hash functions used in the scheme are : $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^n$, $H_3 : \{0,1\}^* \rightarrow G_1$, $H_4 : \{0,1\}^* \rightarrow G_1$. Here, n is the maximum number of bits in a message. The master secret key s is selected uniformly at random from Z_p, and the master public key $P_{pub} = sP$. The public parameters of the system are $params$= $\langle \varGamma, P, P_{pub}, q, n, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_1, H_2, H_3, H_4 \rangle$.

The partial private key extraction algorithm on input (ID, s) returns $D = sH_1(ID) = sQ$. The user key generation algorithm returns a random element $x \in Z_p$ as the secret value, and $PK = xP$ as the public key of user with identity ID. The full private key of user with identity ID is $S = (x, D)$. Message, ciphertext and randomness spaces are $\{0,1\}^\kappa$, $G_1 \times \{0,1\}^\kappa \times G_1$ and Z_p respectively.

Signcrypt$(m, S_S = (x_S, D_S), ID_S, PK_S, ID_R, PK_R, P_{pub})$

- Choose $r \in Z_p$.
- Compute $U = rP$ and $T = \hat{e}(P_{pub}, Q_R)^r$
- Compute $h = H_2(U, T, rPK_R, ID_R, PK_R)$
- Compute $V = m \oplus h$
- Compute $H = H_3(U, V, ID_S, PK_S)$ and $H' = H_4(U, V, ID_S, PK_S)$

– Compute $W = D_S + rH + x_S H'$
– Set $c = (U, V, W)$
– Return the signcryption c of message m from ID_S to ID_R.

Unsigncrypt$(c, S_R = (x_R, D_R), ID_R, PK_R, ID_S, PK_S, P_{pub})$

– The ciphertext c is of the form (U, V, W).
– $H = H_3(U, V, ID_S, PK_S)$ and $H' = H_4(U, V, ID_S, PK_S)$
– If the check $e(P_{pub}, Q_S)e(U, H)e(PK_S, H') \stackrel{?}{=} e(P, W)$ fails, return "$Invalid$".
– Compute $T = \hat{e}(D_R, U)$
– Compute $h = H_2(U, T, x_R U, ID_R, PK_R)$
– Retrieve $m = V \oplus h$
– Return the message m.

Note: The certificateless signcryption scheme uses an Encrypt-then-Sign approach. A common randomness is shared between the signature and encryption components in the scheme to bind them together.

3.2 Attack on Barbosa et al. Certificateless signcryption scheme

The scheme proposed by Barbosa et al. in [4] is existentially forgeable. The scheme uses the Encrypt-then-Sign approach with public verifiability of ciphertext. The intuition behind the attack: for any signcryption scheme following the Encrypt-then-Sign approach, the identity of the sender should be bound to the encryption and the identity of the receiver should be bound to the signature. In [4], the authors have achieved this binding by using a common randomness for encryption and signature independently but they failed to bind the receiver to the signature. This led to the attack on existential unforgeability of [4]. The attack is shown below.

– During the unforgeability game (Both type-1 and type-2), the forger requests a signcryption on a message m from ID_S^* to a arbitrary user with identity ID_A.
– Let the signcryption of m from ID_S^* to ID_A be $c = (U, V, W)$.
– Now, the forger submits $c^* = (U, V, W)$ as a signcryption from user ID_S^* to ID_R^*, where ID_S^* is the target sender identity for which the forger is not allowed to know the private key (partial private key for Type-I and user private key for Type-II forgers respectively) and ID_R^* is the new receiver identity. Note that c^* is a valid signcryption of some random message $m^* = m \oplus h \oplus h^*$ where $h^* = H_2(U, T^*, x_R^* U, ID_R^*, PK_R^*)$ and $T^* = \hat{e}(D_R^*, U)$. Here $h = H_2(U, T, x_A U, ID_A, PK_A)$ is the key used for encrypting the message m from ID_S^* to ID_A during signcryption.
– The signature W will pass the verification because none of the components of the H and H' are altered. The correctness of the signcryption is straight forward as follows.

$$e(P_{pub}, Q_S)e(U, H)e(PK_S, H') = e(P, W)$$

where $H = H_3(U, V, ID_S, PK_S)$ and $H' = H_4(U, V, ID_S, PK_S)$

So the challenger will accept c^* as a valid forgery on message $m^* = m \oplus h \oplus h^*$.

4 Certificateless Signcryption Scheme of Diego et al.

In this section, we give the review and attack of the certificateless signcryption scheme by Diego et al. [2].

4.1 Overview of the Scheme

Diego et al.'s CLSC scheme [2] consists of five algorithms namely: *Setup*, *Extract*, *Keygen*, *Signcrypt* and *Unsigncrypt*, which we describe below.

– ***Setup.*** Let κ be the security parameter. The KGC performs the following to set up the system.
  - The KGC selects cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of same order q with generators $P \in_R \mathbb{G}_1$ and $Q \in_R \mathbb{G}_2$.
  - Selects the master secret key $s \in_R \mathbb{Z}_q^*$ and the master public key is set to be $P_{pub} = sP$.
  - Selects an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

- Computes $g = \hat{e}(P, Q)$.
- Selects three hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*, H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n, H_3 : \{0,1\}^n \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$, Here n is the length of the message.
- The public parameters of the scheme are set to be $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g, P, Q, P_{pub}, H_1, H_2, H_3 \rangle$.

– ***Extract.*** Here, ID_A is the identity of the user U_A, the KGC computes the partial private key of user U_A as follows.
  - Computes the hash value $y_A = H_1(ID_A)$ and the partial private key $D_A = (y_A + s)^{-1}Q \in \mathbb{G}_2$.
  - The KGC sends D_A to the user U_A via a secure authenticated channel.

– ***Keygen.*** User U_A computes the full private key by performing the following steps:
  - Chooses $x_A \in_R \mathbb{Z}_q^*$ as the secret value.
  - Computes the full private key $S_A = x_A^{-1} D_A \in \mathbb{G}_2$.
  - Computes the public key as $P_A = x_A(y_A P + P_{pub}) \in \mathbb{G}_1$.
  - It is to be noted that $\hat{e}(P_A, S_A) = g$.

– ***Signcrypt.*** Inorder to signcrypt the message m to the receiver U_B, the sender U_A does the following:
  - Chooses $r \in_R \mathbb{Z}_q^*$, computes $u = r^{-1}$ and $U = g^u$.
  - Computes $c = m \oplus H_2(U)$, $R = rP_A$ and $S = uP_B$.
  - Computes $h = H_3(c, R, S)$ and $T = (r + h)^{-1} S_A$.

  Finally, the sender outputs the signcryption on message m as $\sigma = \langle c, R, S, T \rangle$.

– ***Unsigncrypt.*** Inorder to unsigncrypt a ciphertext σ, the receiver U_B does the following:
  - Computes $h' = H_3(c, R, S)$.
  - Computes $U' = \hat{e}(S, S_B)$.
  - Recovers the message as $m' = c \oplus H_2(U')$.
  - Checks whether $\hat{e}(R + h' P_A, T) \stackrel{?}{=} g$.

  If the check holds, then accepts m' as the message, otherwise outputs $Invalid$.

4.2 Attack on the CLSC Scheme by Diego et al.

Type-I Forgeability: The Type-I adversary who is capable of replacing the public keys of all users and is not allowed to know the master private key can forge a valid signcryption on any message m, from any legitimate user U_A to U_B by performing the following:

– Let ID_A be the identity of user U_A.
– The adversary chooses $r \in_R \mathbb{Z}_q^*$, computes $u = r^{-1}$.
– Computes $U = g^u$ and sets $c = m \oplus H_2(U)$.
– Set $T = uQ$, $R = rP - P$ and $S = uP_B$.
– Compute $h = H_3(c, R, S)$.
– Set $P_A = h^{-1}P$.

Finally, the forger outputs the signcryption on message m as $\sigma = (c, R, S, T)$ which is a valid signcryption on m from U_A to U_B.

Correctness: The signcryption σ, which is produced as forgery passes the verification test as shown below,

$$\begin{aligned} \hat{e}(R + hP_A, T) &= \hat{e}(rP - P + hh^{-1}P, uQ) \\ &= \hat{e}(rP, uQ)\hat{e}(-P + P, uQ) \\ &= \hat{e}(P, Q)\hat{e}(-P, uQ)\hat{e}(P, uQ) \\ &= \hat{e}(P, Q) \\ &= g \end{aligned}$$

This proves that the forgery generated is valid.

Type-I and Type-II Attacks on Confidentiality:

- Let $\sigma^* = (c^*, R^*, S^*, T^*)$ be the challenge signcryption on message m_b, $b \in \{0,1\}$ with ID_A as the sender and ID_B as the receiver.
- The adversary is capable of generating a new signcryption σ' on the message m_b (The message is same as in σ^*) with ID_C as sender and ID_B as receiver (Note that the adversary knows the private key of ID_C).
- σ' is obtained by the adversary by performing the following:
  - Sets $c' = c^*$.
  - Computes $R' = r' P_C$, where $r' \in_R \mathbb{Z}_q^*$.
  - Set $S' = S^*$.
  - Computes $h' = H_3(c', R', S')$
  - Set $T' = (r' + h')^{-1} S_C$
  - The signcryption corresponding to this change is $\sigma' = \langle c', R', S', T' \rangle$.
- Now, the adversary can query the unsigncryption oracle for the unsigncryption of σ' (Note that this query is valid because σ' is different from the challenge signcryption σ^*).
- The unsigncryption oracle will give back the message m_b since the key used in both σ^* and σ' are the same i.e., $U' = \hat{e}(S', S_B) = \hat{e}(S^*, S_B) = U^*$ and note that $S' = S^*$. Hence , $c' \oplus H_2(U') = c^* \oplus H_2(U^*) = m_b$.
- Therefore, designcryption of σ' outputs the message m_b, which is used for generating the challenge ciphertext σ^*. Thus the adversary can determine whether $m_b \stackrel{?}{=} (m_0 \text{ or } m_1)$ breaking the indistinguishability of the scheme. This attack can be performed by both Type-I and Type-II adversaties because the adversary does not require the master private key or even does not want to replace the public key.

5 Certificateless Signcryption Scheme of Chen-Huang et al.

In this section, we present the review and attack of the certificateless signcryption scheme by Chen-Huang et al. [14].

5.1 Overview of the Scheme

The CLSC scheme of Chen-Huang et al. [14] consists of the following four algorithms.

- ***Setup.*** Given κ as the security parameter, the KGC does the following to setup the system parameters.
  - The KGC selects $\mathbb{G}_1$, $\mathbb{G}_2$ of same prime order q with a generator $P \in_R \mathbb{G}_1$.
  - Selects the master secret key $s \in_R \mathbb{Z}_q^*$ and the master public key is set to be $P_{pub} = sP$.
  - Selects an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
  - Selects three cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1, H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*, H_3 : \{0,1\}^* \rightarrow \{0,1\}^n$, where n is the size of the message.
  - Computes $T = \hat{e}(P, P)$.
  - The public parameters of the scheme are set to be $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, T, H_1, H_2, H_3 \rangle$.
- ***Keygen.*** Let, ID_A be the identity of the user U_A. The KGC computes the partial private key of user U_A as follows.
  - Computes $Q_A = H_1(ID_A)$ and the partial private key $D_A = sQ_A \in \mathbb{G}_2$.
  - The KGC sends D_A to the user U_i via a secure authenticated channel.

  On receiving the partial private key D_A, user U_A computes his full private key by performing the following steps:
  - U_A chooses $x_A \in_R \mathbb{Z}_q^*$ as the secret value.
  - Sets the full private key $S_A = \langle x_A, D_A \rangle$.
  - The corresponding public key is $P_A = T^{x_A} \in \mathbb{G}_2$.
- ***Signcrypt.*** Inorder to signcrypt the message m of length n to the receiver U_B, the sender U_A does the following:
  - Chooses $r, r_1, r_2 \in_R \mathbb{Z}_q^*$, computes $R_1 = T^{r_1}$ and $R_2 = T^{r_2}$.
  - Computes $h = H_2(m \| R_1 \| R_2 \| P_A \| P_B)$.
  - Computes $U = r_1 P - hS_A$ and $u = r_2 - x_A h$.

  - Computes $K = \hat{e}(S_A, Q_B)^r T_B^{x_A}$ and $W = rQ_A$.
  - Computes $c = H_3(K) \oplus m$

  Finally, the sender outputs the signcryption on message m as $\sigma = (c, u, h, U, W)$.

– ***Unsigncrypt.*** Inorder to unsigncrypt a ciphertext σ, the receiver U_B does the following:
  - Computes $K' = \hat{e}(S_B, W) T_A^{x_B}$.
  - Retrieves the message as $m' = c \oplus H_3(K')$.
  - Checks whether $h \stackrel{?}{=} H_2(m' \| \hat{e}(U, P)\hat{e}(Q_A, P_{pub})^h) \| T^u P_A^h \| P_A \| P_B)$.

  If the check holds, then accepts m' as the message, otherwise outputs $Invalid$.

5.2 Attack on the CLSC Scheme by Chen-Huang et al.

In this section, we show that the certificateless signcryption scheme by Chen-Huang et al. does not provide confidentiality as well as unforgeability with respect to both Type-I and Type-II attacks.

Attack on Type-I and Type-II Confidentiality: The following attack is possible because the adversary is capable of altering the challenge signcryption without altering the message in it and is allowed to obtain the unsigncryption of the newly formed signcryption, which yields the message signcrypted in σ^*. We explain the attack in detail now. On getting the challenge signcryption $\sigma^* = \langle c^*, u^*, h^*, U^*, W^* \rangle$, ($\sigma^*$ is the signcryption of either message m_0 or m_1 from user U_A to U_B) the adversary (Type-I and Type-II) is capable of generating a new ciphertext $\sigma' = \langle c', u', h', U', W' \rangle$ (signcryption of m_0 from user U_C to U_B) as follows:

– Replace the public key of user U_C with the public key of user U_A.
– Sets $c' = c^*$ and $W' = W^*$.
– Chooses $r_1, r_2 \in_R \mathbb{Z}_q^*$, computes $R_1 = T^{r_1}$ and $R_2 = T^{r_2}$.
– Computes $h' = H_2(m_0 \| R_1 \| R_2 \| P_C \| P_B)$.
– Computes $U' = r_1 P - h' S_C$ and $u = r_2 - x_C h'$.
– Gets the unsigncryption of σ'.
– If $Unsigncrypt(\sigma')$ = "m_0" then the adversary outputs that σ^* is the signcryption of m_0 (i.e. $b' = 0$).
– If $Unsigncrypt(\sigma')$ = "$Invalid$" then the adversary outputs m_1 (i.e. $b' = 1$).

Note: This attack can be done by both Type-I and Type-II adversaries.

6 Certificateless Signcryption Without Pairing

– **CLSC.Setup(1^κ):** The KGC takes the security parameter 1^κ as input and performs the following for setting up the system:
  - Chooses two big prime numbers p and q such that $q|(p-1)$.
  - Selects an element $g \in_R \mathbb{Z}_p^*$ with order q.
  - Chooses a master private key $s \in_R \mathbb{Z}_q^*$ and computes the master public key g_{pub}=g^s.
  - Chooses five cryptographic hash functions $H_1 : \{0,1\}^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_q^*$, $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $H_4 : \{0,1\}^* \rightarrow |\mathcal{M}| \times \mathbb{Z}_q^* \times \mathbb{Z}_q^*$, $H_5 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, here $\mathcal{M}$ is the message space.

  The public parameters of the system, $params = \langle p, q, g, g_{pub}, H_1, H_2, H_3, H_4, H_5 \rangle$.

– **CLSC.PartialPrivateKeyExtract:** Given an identity, say ID_A of a user U_A, the KGC performs the following to generate the partial private key corresponding to ID_A:
  - Chooses $x_{A0}, x_{A1} \in_R \mathbb{Z}_q^*$.
  - Computes $X_{A0} = g^{x_{A0}}$ and $X_{A1} = g^{x_{A1}}$.
  - Computes $q_{A0} = H_1(ID_A, X_{A0})$ and $q_{A1} = H_2(ID_A, X_{A0}, X_{A1})$.
  - Computes $d_{A0} = x_{A0} + sq_{A0}$ and $d_{A1} = x_{A1} + sq_{A1}$.

  Returns $d_A = \langle d_{A0}, d_{A1} \rangle$ and $X_A = \langle X_{A0}, X_{A1} \rangle$, the partial private keys securely to user U_A.

  Note: It should be noted that the partial private key of a user is a Schnorr signature on the user's identity, signed by the KGC using the master private key.

– **CLSC.SetSecretValue:** The user U_A chooses an element $y_A \in_R \mathbb{Z}_q^*$ and keeps it as his secret value.
– **CLSC.SetPrivateKey:** The user U_A sets his full private key $s_A = \langle y_A, d_{A0} \rangle$.

– **CLSC.SetPublicKey:** The user U_A computes $Y_A = g^{y_A}$ and sets his public key as $PK_A = \langle d_{A1},\ X_{A0}, X_{A1}, Y_A \rangle$. The resulting public key is distributed widely and freely.

– **CLSC.Signcrypt:** The sender U_A signcrypts a message m to a receiver U_B by performing the following:
  - Chooses $r_1, r_2 \in_R \mathbb{Z}_q^*$, computes $c_1 = g^{r_1}$ and $c_2 = g^{r_2}$.
  - Computes $k_1 = (Y_B)^{r_1}$ and $k_2 = (X_{B0}.(g_{pub})^{q_{B0}})^{r_1}$.
  - Computes $d = H_3(m, c_2, ID_A, ID_B, PK_A)$ and $e = H_5(m, c_2, ID_A, ID_B, PK_A)$.
  - Computes $v = (d.d_{A0} + e.y_A) + r_2$.
  - Computes $c_3 = H_4(k_1, k_2, ID_A, ID_B) \oplus (m \| r_1 \| v)$.

  Now $c = \langle c_1, c_2, c_3 \rangle$ is the signcryption on message m to user U_B

– **CLSC.Unsigncrypt:** To unsigncrypt a signcryption $c = \langle c_1, c_2, c_3 \rangle$ from sender U_A, the receiver U_B does the following:
  - Computes $k_1' = (c_1)^{y_B}$ and $k_2' = (c_1)^{d_{B0}}$.
  - Computes $(m' \| r_1' \| v') = c_3 \oplus H_4(k_1', k_2', ID_A, ID_B)$.
  - Checks whether $g^{r_1'} \stackrel{?}{=} c_1$.
  - If so computes $d' = H_3(m', c_2, ID_A, ID_B, PK_A)$ and $e' = H_5(m', c_2, ID_A, ID_B, PK_A)$.
  - Checks whether $g^{v'} \stackrel{?}{=} ((g_{pub})^{q_{A0}}.X_{A0})^{d'}.(Y_A)^{e'}.c_2$.

  If both the checks hold, m' is output as the unsigncrypted message else outputs "$Invalid$".

Correctness: The correctness of the verification test $g^{r_1'} \stackrel{?}{=} c_1$ is straight forward. The second check also passes the verification if the signcryption is formed in a legitimate way which is shown below.

$$\begin{aligned} g^{v'} &= g^{(d_{A0}d' + y_A e') + r_2} \\ &= g^{(x_{A0}d' + s q_{A0} d' + y_A e') + r_2} \\ &= g^{x_{A0}d'}.g^{s q_{A0} d'}.g^{y_A e'}.g^{r_2} \\ &= ((g_{pub})^{q_{A0}}.X_{A0})^{d'}.(Y_A)^{e'}.c_2 \end{aligned}$$

7 Security of CLSC Scheme

In this section, we provide the formal proof for the unforgeability and confidentiality of the CLSC scheme.

7.1 Type-I Unforgeability

Theorem 1. *If an EUF-CLSC-CMA-I forger $\mathcal{F}_I$ has advantage ϵ_0 against CLSC scheme, asking q_{H_i} $(i = 1, 2, 3, 4, 5)$ hash queries to random oracles H_i $(i = 1, 2, 3, 4, 5)$, q_{sc} signcryption queries, q_{us} unsigncryption queries, q_{pkr} extract secret value queries, q_{ppk} partial private key extract queries, q_{pk} public key request queries and q_{rpk} public key replacement queries, then there exist an algorithm $\mathcal{C}$ that solves the DL problem with advantage*

$$\epsilon \geq \frac{1}{9}.\left(\frac{\epsilon'.(1-\alpha)^{q_{ppk}}.\left(1 - \frac{q_{ppk}}{q_{pk}}\right)}{q_{pk}} - \frac{q_{us}}{q} - \frac{q_{sc}.\left(q_{H_3} + q_{H_5} + q_{sc}\right)}{2^{\kappa}} \right)$$

where, α is the advantage of an adversary breaking the Schnorr signature scheme.

7.2 Type-II Unforgeability

Theorem 2. *If an EUF-CLSC-CMA-II forger $\mathcal{F}_{II}$ has advantage ϵ_0 against CLSC scheme, asking q_{H_i} $(i = 1,\ 2,\ 3,\ 4,\ 5)$ hash queries to random oracles H_i $(i = 1,\ 2,\ 3,\ 4,\ 5)$, q_{sc} signcryption queries, q_{us} unsigncryption queries, q_{pkr} extract secret value queries, q_{ppk} partial private key extract queries, q_{pk} public key request queries and q_{rpk} public key replacement queries, then there exist an algorithm $\mathcal{C}$ that solves the DL problem with advantage*

$$\epsilon \geq \frac{1}{9}.\left(\frac{\epsilon'.\left(1 - \frac{q_{ppk}}{q_{pk}}\right).\left(1 - \frac{q_{rpk}}{q_{pk}}\right)}{q_{pk}} - \frac{q_{us}}{q} - \frac{q_{sc}.\left(q_{H_3} + q_{H_5} + q_{sc}\right)}{2^{\kappa}} \right)$$

7.3 Type-I Confidentiality

Theorem 3. *If an EUF-CLSC-CCA2-I adversary $\mathcal{A}_I$ has advantage ϵ against CLSC scheme, asking q_{H_i} $(i = 1, 2, 3, 4, 5)$ hash queries to random oracles H_i $(i = 1, 2, 3, 4, 5)$, q_{sc} signcryption queries, q_{us} unsigncryption queries, q_{pkr} extract secret value queries, q_{ppk} partial private key extract queries, q_{pk} public key request queries and q_{rpk} public key replacement queries, then there exist an algorithm $\mathcal{C}$ that solves the CDH problem with advantage*

$$\epsilon' \geq \frac{1}{q_4^{'}}\left(\epsilon.(1-\alpha)^{q_{ppk}}\left(1-\frac{q_{ppk}}{q_{pk}}\right)-\left(\frac{q_{sc}.\left(q_{H_3}+q_{H_5}+q_{sc}\right)}{2^{\kappa}}\right)-\frac{q_{us}}{q}\right)$$

where, $q_4^{'}$ is the number of tuples in the L_{H_4} list having $\langle ID_A, ID_\gamma\rangle$ and α is the advantage of an adversary in breaking the Schnorr signature scheme.

7.4 Type-II Confidentiality

Theorem 4. *If an EUF-CLSC-CCA2-II adversary $\mathcal{A}_{II}$ has advantage ϵ against CLSC scheme, asking q_{H_i} $(i = 1, 2, 3, 4, 5)$ hash queries to random oracles H_i $(i = 1, 2, 3, 4, 5)$, q_{sc} signcryption queries, q_{us} unsigncryption queries, q_{pkr} extract secret value queries, q_{ppk} partial private key extract queries, q_{pk} public key request queries and q_{rpk} public key replacement queries, then there exist an algorithm $\mathcal{C}$ that solves the CDH problem with advantage*

$$\epsilon' \geq \frac{1}{q_4^{'}}\left(.\epsilon.(1-\alpha)^{q_{ppk}}\left(1-\frac{q_{ppk}}{q_{pk}}\right)-\left(\frac{q_{sc}.\left(q_{H_3}+q_{H_5}+q_{sc}\right)}{2^{\kappa}}\right)-\frac{q_{us}}{q}\right)$$

where $q_4^{'}$ is the number of tuples in the L_{H_4} list having $\langle ID_A, ID_\gamma\rangle$.

Note: Security proofs will be available soon.

8 Conclusion

In this work, we have showed the security weakness in three existing certificateless signcryption schemes that appear in [4], [2] and [14]. We have also presented the first pairing free certificateless signcryption scheme in the random oracle model. The proposed scheme is more efficient since the scheme evades bilinear pairing. We have proved the security of the scheme with the strongest security notion for signcryption schemes, namely insider security. We leave it as an open problem to construct certificateless signcryption scheme without pairing in the standard model. As a concluding remark we present the complexity figure of the new CLSC scheme in the following table.

Scheme	Signcrypt	Unsigncrypt
CLSC	5 *EXP*	7 *EXP*

Table-1: Complexity figure for CLSC

EXP - Exponentiation in group $\mathbb{G}$

References

1. Sattam S. Al-Riyami and Kenneth G. Paterson. Certificateless public key cryptography. In *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473. Springer, 2003.
2. Diego Aranha, Rafael Castro, Julio Lopez, and Ricardo Dahab. Efficient certificateless signcryption. http://sbseg2008.inf.ufrgs.br/proceedings/data/pdf/st03_01_resumo.pdf.
3. Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Certificateless public key encryption without pairing. In *Information Security - ISC 2005*, volume 3650 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 2005.
4. Manuel Barbosa and Pooya Farshim. Certificateless signcryption. In *ACM Symposium on Information, Computer and Communications Security - ASIACCS 2008*, pages 369–372. ACM, 2008.
5. Paulo S. L. M. Barreto, Alexandre Machado Deusajute, Eduardo de Souza Cruz, Geovandro C. F. Pereira, and Rodrigo Rodrigues da Silva. Toward efficient certificateless signcryption from (and without) bilinear pairings. http://sbseg2008.inf.ufrgs.br/proceedings/data/pdf/st03_03_artigo.pdf.

6. Zhaohui Cheng and Richard Comley. Efficient certificateless public key encryption. Cryptology ePrint Archive, Report 2005/012, 2005. http://eprint.iacr.org/.
7. Alexander W. Dent. A survey of certificateless encryption schemes and security models. *International Journal of Information Security*, Volume-7(5):349–377, 2008.
8. Joseph K Liu, Man Ho Au, and Willy Susilo. Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model: extended abstract. In *ASIACCS 2007, Proceedings of the 2nd ACM symposium on Information, Computer and Communications Security*, pages 273–283. ACM, 2007.
9. Jong Hwan Park, Kyu Young Choi, Jung Yeon Hwang, and Dong Hoon Lee. Certificateless public key encryption in the selective-id security model (without random oracles). In *Pairing-Based Cryptography - Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 60–82. Springer, 2007.
10. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
11. Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, CRYPTO - 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
12. Yijuan Shi and Jianhua Li. Provable efficient certificateless public key encryption. Cryptology ePrint Archive, Report 2005/287, 2005. http://eprint.iacr.org/.
13. Yinxia Sun, Futai Zhang, and Joonsang Baek. Strongly secure certificateless public key encryption without pairing. In *Cryptology and Network Security - CANS 2007*, volume 4856 of *Lecture Notes in Computer Science*, pages 194–208. Springer, 2007.
14. Chenhuang Wu and Zhixiong Chen. A new efficient certificateless signcryption scheme. In *IEEE, International Symposium on Information Science and Engieering, 2008. ISISE '08.*, volume 1, pages 661–664, 2008.
15. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) $<<$ cost(signature) + cost(encryption). In *Advances in Cryptology, CRYPTO - 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 1997.