Analysis and Improvement of a Signcryption Scheme with Key Privacy

Guomin Yang, Duncan S. Wong[*], and Xiaotie Deng

Department of Computer Science
City University of Hong Kong
Hong Kong, China
{csyanggm,duncan,deng}@cs.cityu.edu.hk

Abstract. In PKC'04, a signcryption scheme with key privacy was proposed by Libert and Quisquater. Along with the scheme, some security models were defined with regard to the signcryption versions of confidentiality, existential unforgeability and ciphertext anonymity (or key privacy). The security of their scheme was also claimed under these models. In this paper, we show that their scheme cannot achieve the claimed security by demonstrating an insider attack which shows that their scheme is not semantically secure against chosen ciphertext attack (not even secure against chosen plaintext attack) or ciphertext anonymous. We further propose a revised version of their signcryption scheme and show its security under the assumption that the gap Diffie-Hellman problem is hard. Our revised scheme supports parallel processing that can help reduce the computation time of both signcryption and de-signcryption operations.

Keywords: Signcryption, Key Privacy, Ciphertext Anonymity, Bilinear Pairings, Gap Diffie-Hellman Groups

1 Introduction

Signcryption, introduced by Zheng in 1997 [17], is a public key primitive which has the ingredients of both digital signature and data encryption. A signcryption scheme allows a sender to simultaneously sign and encrypt a message for a receiver in such a way that it takes less computation time and has lower message expansion rate than that of performing signature generation and then encryption separately, which is referred to as signature-then-encryption procedure [17]. The performance advantage of signcryption over the signature-then-encryption procedure makes signcryption attractive to providing secure and authenticated message delivery for resource constrained devices such as low-power mobile units, smart cards, and emerging sensors.

[*] The work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 9040904 (RGC Ref. No. CityU 1161/04E)).

J. Zhou et al. (Eds.): ISC 2005, LNCS 3650, pp. 218–232, 2005.

A number of signcryption schemes were proposed after Zheng's work [2, 13, 15, 16, 12, 9, 11]. In 2002, formal security proofs for Zheng's schemes were given by Baek et al. [3]. In their paper, they defined a notion similar to semantic security against adaptive chosen ciphertext attack (IND-CCA2) [14] for message confidentiality and a notion similar to existential unforgeability against chosen message attack (EUF-CMA) [10] for signature unforgeability.

In [1], An et al. described a new security notion called 'Insider Security'.[1] The notion of 'Insider Security' is to allow an adversary to have access to the sender's private key besides the public keys of the sender and the receiver. If a signcryption scheme is 'Insider Secure', then this adversary should not be able to obtain the message of a signcryption from the sender. Instead, it is similar to the requirement for the conventional signature-then-encryption procedure that only the one who has the receiver's private key can open a signcryption. In some cases, it becomes important for ensuring 'Insider Security'. For example, if an adversary happens to steal the sender's private key, then we do not want all previous (and future) signcrypted ciphertexts from the honest sender being compromised by the adversary.

In [9], more security notions for signcryption schemes have been defined under the identity-based setting. One of which is "Ciphertext Anonymity". It captures the property that the ciphertext must contain no information in the clear that identifies the sender or recipient of the message. This can be considered as an extension to the notion of "Key-Privacy" defined by Bellare et al. [4] for public key encryption.

In [11], a new signcryption scheme claiming to have ciphertext anonymity (or key privacy) was proposed. Along with the scheme, some security models were also defined with regard to the signcryption versions of confidentiality, existential unforgeability and ciphertext anonymity (or key privacy). In particular, these models captured the notions of IND-CCA2, EUF-CMA, Insider Security and Ciphertext Anonymity. The security of their scheme was also claimed under these models.

However, we find that their scheme cannot achieved the claimed security. In this paper, we demonstrate an insider attack which shows that their scheme is not semantically secure against adaptive chosen ciphertext attack (not even secure against chosen plaintext attack). The same attacking technique also compromises its ciphertext anonymity.

We further propose a revised/improved version of their scheme and show its security under the assumption that the gap Diffie-Hellman problem is hard. Our improved scheme supports parallel processing which can help reduce the computation time of both signcryption and de-signcryption operations.

Organization. In the rest of the paper, we first give the definition and security models of a signcryption scheme with key privacy in Sec. 2. It is then followed by the description of the Libert-Quisquater scheme in Sec. 3. We show that their

[1] The original paper of An et al. [1] only presents the insider attack against the integrity of a signcryption. The idea has later been extended to confidentiality and other security properties [9, 11].

scheme is not semantically secure against chosen plaintext attack and hence not secure against chosen ciphertext attack. We also show that key privacy is not achieved either. In Sec. 4, a modification of their scheme is described. Security and performance analyses are also given. We conclude the paper in Sec. 5.

2 The Definition and Security Models of a Signcryption Scheme with Key Privacy

A signcryption scheme is a quadruple of probabilistic polynomial time (PPT) algorithms (**Keygen**, **Signcrypt**, **De-signcrypt**, **Verify**).

$(sk, pk) \leftarrow \textbf{Keygen}(1^k)$ is the key generation algorithm which takes a security parameter k and generates a private/public key pair (sk, pk).

$\sigma \leftarrow \textbf{Signcrypt}(1^k, m, sk_U, pk_R)$ takes as inputs a security parameter k, a message m, a private key sk_U and a public key pk_R, outputs a ciphertext σ. m is drawn from a message space M which is defined as $\{0,1\}^n$ where n is some polynomial in k.

$(m, s, pk_U)/\texttt{reject} \leftarrow \textbf{De-signcrypt}(1^k, \sigma, sk_R)$ takes as inputs a security parameter k, a ciphertext σ and a private key sk_R, outputs either a triple (m, s, pk_U) where m is a message, s is a signature and pk_U is a public key, or `reject` which indicates the failure of de-signcryption.

$\texttt{true}/\texttt{false} \leftarrow \textbf{Verify}(1^k, m, s, pk_U)$ takes as inputs a security parameter k, a message m, a signature s and a public key pk_U, outputs `true` for a valid signature or `false` for an invalid signature.

For simplicity, we omit the notation of 1^k from the inputs of **Signcrypt**, **De-signcrypt** and **Verify** in the rest of this paper.

Note that the specification above requires the corresponding signcryption scheme to support the "unwrapping" option which was introduced in [12]. The "unwrapping" option allows the receiver of a ciphertext to release the message and derive the embedded sender's signature from the ciphertext for public verification. Early schemes such as [17] do not support the "unwrapping" option and therefore not satisfy this definition.

Definition 1 (Completeness). *For any $m \in M$, $(sk_U, pk_U) \leftarrow \textbf{Keygen}(1^k)$ and $(sk_R, pk_R) \leftarrow \textbf{Keygen}(1^k)$ such that $sk_U \neq sk_R$, we have*

$$(m, s, pk_U) \leftarrow \textbf{\textit{De-signcrypt}}(\textbf{\textit{Signcrypt}}(m, sk_U, pk_R), sk_R)$$

and $\texttt{true} \leftarrow \textbf{Verify}(m, s, pk_U)$.

Informally, we consider a secure signcryption scheme with key privacy to be semantically secure against adaptive chosen ciphertext attack, existentially unforgeable against chosen message attack, and anonymous in the sense that a ciphertext should contain no information in the clear that identifies the author or the recipient of the message and yet be decipherable by the intended recipient without that information. We capture these notions in the following definitions. They are similar to those defined by Libert and Quisquater [11].

Definition 2 (Confidentiality). *A signcryption scheme is semantically secure against chosen ciphertext insider attack (SC-IND-CCA) if no PPT adversary has a non-negligible advantage in the following game:*

1. *The challenger runs* ***Keygen*** *to generate a key pair* (sk_U, pk_U). sk_U *is kept secret while* pk_U *is given to adversary* $\mathcal{A}$.
2. *In the first stage,* $\mathcal{A}$ *makes a number of queries to the following oracles:*
   (a) *Signcryption oracle:* $\mathcal{A}$ *prepares a message* $m \in M$ *and a public key* pk_R*, and queries the signcryption oracle (simulated by the challenger) for the result of* ***Signcrypt***(m, sk_U, pk_R)*. The result is returned if* $pk_R \neq pk_U$ *and* pk_R *is valid in the sense that* pk_R *is in the range of* ***Keygen*** *with respect to the security parameter. Otherwise, a symbol '*$\perp$*' is returned for rejection.*
   (b) *De-signcryption oracle:* $\mathcal{A}$ *produces a ciphertext* σ *and queries for the result of* ***De-signcrypt***(σ, sk_U)*. The result is made of a message, a signature and the sender's public key if the de-signcryption is successful and the signature is valid under the recovered sender's public key. Otherwise, a symbol '*$\perp$*' is returned for rejection.*

   These queries can be asked adaptively: each query may depend on the answers of previous ones.
3. $\mathcal{A}$ *produces two plaintexts* $m_0, m_1 \in M$ *of equal length and a valid private key* sk_S *such that* sk_S *is in the range of* ***Keygen*** *with respect to the security parameter. The challenger flips a coin* $\check{b} \overset{R}{\leftarrow} \{0,1\}$ *and computes a signcryption* $\sigma^* =$ ***Signcrypt***$(m_{\check{b}}, sk_S, pk_U)$ *of* $m_{\check{b}}$ *with the sender's private key* sk_S *under the receiver's public key* pk_U. σ^* *is sent to* $\mathcal{A}$ *as a challenge ciphertext.*
4. $\mathcal{A}$ *makes a number of new queries as in the first stage with the restriction that it cannot query the de-signcryption oracle with* σ^*.
5. *At the end of the game,* $\mathcal{A}$ *outputs a bit* b' *and wins if* $b' = \check{b}$.

$\mathcal{A}$*'s advantage is defined as* $Adv^{ind\text{–}cca}(\mathcal{A}) = \Pr[b' = \check{b}] - \frac{1}{2}$ *and the probability that* $b' = \check{b}$ *is called the probability that* $\mathcal{A}$ *wins the game.*

The definition above captures the advantage of an active adversary over an eavesdropper. That is, the adversary knows and has the full control of the signing key. This also gives us insider-security for confidentiality [1].

Definition 3 (Unforgeability). *A signcryption scheme is existentially unforgeable against chosen-message insider attack (SC-EUF-CMA) if no PPT forger has a non-negligible advantage in the following game:*

1. *The challenger runs* ***Keygen*** *to generate a key pair* (sk_U, pk_U). sk_U *is kept secret while* pk_U *is given to forger* $\mathcal{F}$.
2. *The forger* $\mathcal{F}$ *adaptively makes a number of queries to the signcryption oracle and the de-signcryption oracle as in the confidentiality game.*
3. $\mathcal{F}$ *produces a ciphertext* σ *and a valid key pair* (sk_R, pk_R) *in the sense that the key pair is in the range of* ***Keygen*** *and wins the game if*

(a) ***De-signcrypt***(σ, sk_R) *returns a tuple* (m, s, pk_U) *such that* $\texttt{true} \leftarrow$ ***Verify***(m, s, pk_U)*, and*
(b) σ *is not the output of the signcryption oracle.*

We allow the forger to have the full control of the de-signcryption key pair (sk_R, pk_R). This also captures the notion of insider-security for unforgeability.

Definition 4 (Ciphertext Anonymity). *A signcryption scheme is ciphertext anonymous against chosen-ciphertext insider attack (SC-ANON-CCA) if no PPT distinguisher has a non-negligible advantage in the following game:*

1. *The challenger generates two distinct public key pairs* $(sk_{R,0}, pk_{R,0})$ *and* $(sk_{R,1}, pk_{R,1})$ *using* ***Keygen****, and gives* $pk_{R,0}$ *and* $pk_{R,1}$ *to the distinguisher* $\mathcal{D}$*.*
2. *In the first stage,* $\mathcal{D}$ *adaptively makes a number of queries in the form of* ***Signcrypt***$(m, sk_{R,c}, pk_R)$ *or* ***De-signcrypt***$(\sigma, sk_{R,c})$*, for* $c = 0$ *or* $c = 1$*.* pk_R *is some arbitrary but valid recipient key such that* $pk_R \neq pk_{R,c}$*.*
3. *After completing the first stage,* $\mathcal{D}$ *outputs two valid and distinct private keys* $sk_{S,0}$ *and* $sk_{S,1}$*, and a plaintext* $m \in M$*.*
4. *The challenger then flips two coins* $b, b' \xleftarrow{R} \{0, 1\}$ *and computes a challenge ciphertext* $\sigma =$ ***Signcrypt***$(m, sk_{S,b}, pk_{R,b'})$ *and sends it to* $\mathcal{D}$*.*
5. $\mathcal{D}$ *adaptively makes a number of new queries as above with the restriction that it is not allowed to ask the de-signcryption oracle of the challenge ciphertext* σ*.*
6. *At the end of the game,* $\mathcal{D}$ *outputs bits* d, d' *and wins the game if* $(d, d') = (b, b')$*.*

$\mathcal{D}$*'s advantage is defined as* $Adv^{anon-cca}(\mathcal{D}) = \Pr[(d, d') = (b, b')] - \frac{1}{4}$*.*

The ciphertext anonymity definition above follows that of Libert and Quisquater in [11, Def. 4], which is considered to be an extension of the "Key-Privacy" notion of public key encryption [4]. We only consider this definition for key privacy in this paper rather than also considering an additional one called key invisibility [11, Def. 5]. We believe that the definition above is more intuitive. With only a few differences, one can also consider it as a non-identity based version of Boyen's definition [9] of ciphertext anonymity in the identity-based setting.

3 Security Analysis of the Libert-Quisquater Scheme

3.1 Preliminaries

Bilinear Pairings. Let k be a system-wide security parameter. Let q be a k-bit prime. Let G_1 be an additive cyclic group of order q and G_2 be a multiplicative cyclic group of the same order. Let P be a generator of G_1. A bilinear map is defined as $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1. *Bilinear*: For all $U, V \in G_1$, and $a, b \in \mathbb{Z}$, we have $e(aU, bV) = e(U, V)^{ab}$.
2. *Non-degenerate*: $e(P, P) \neq 1$.
3. *Computable*: there is an efficient algorithm to compute $e(U, V)$ for any $U, V \in G_1$.

Modified pairings [7] obtained from the Weil or the Tate pairing provide admissible maps of this kind.

The Gap Diffie-Hellman Problem. The Decisional Diffie-Hellman problem (DDH) [6] in G_1 is to distinguish between the distributions of $\langle P, aP, bP, abP\rangle$ and $\langle P, aP, bP, cP\rangle$ where a, b, c are random in $\mathbb{Z}_q$. The Computational Diffie-Hellman problem (CDH) in G_1 is to compute abP from $\langle P, aP, bP\rangle$ where a, b are random in $\mathbb{Z}_q$.

The Gap Diffie-Hellman problem (GDH) is to solve a given random instance $\langle P, aP, bP\rangle$ of the CDH problem with the help of a DDH oracle. The DDH oracle can be implemented through a bilinear map since it suffices to check if the equation $e(P, cP) = e(aP, bP)$ holds for determining if $cP = abP$.

3.2 Libert-Quisquater Signcryption Scheme

Suppose each element in G_1 can distinctly be represented using ℓ bits. Let $H_1 : \{0,1\}^{n+2\ell} \rightarrow G_1$, $H_2 : G_1^3 \rightarrow \{0,1\}^{\ell}$ and $H_3 : \{0,1\}^{\ell} \rightarrow \{0,1\}^{n+\ell}$ be cryptographic hash functions where n denotes the length of a plaintext in binary representation and is some polynomial in k. For security analysis, all hash functions are viewed as random oracles [5]. Also assume that the discrete logarithm of the output of H_1 for any input is hard to compute. The Libert-Quisquater signcryption scheme [11] is reviewed as follows.

Keygen: A private key is generated by picking a random $x_u \leftarrow \mathbb{Z}_q$ and the corresponding public key is computed as $Y_u = x_u P$. In the following, the sender and the receiver are denoted by $u = S$ and $u = R$, and their public key pairs are denoted by (x_S, Y_S) and (x_R, Y_R), respectively.

Signcrypt: To signcrypt a message $m \in \{0,1\}^n$ for receiver R, sender S carries out the following steps:

1. Pick a random $r \leftarrow \mathbb{Z}_q$ and compute $U = rP$.
2. Compute $V = x_S H_1(m, U, Y_R)$.
3. Compute $W = V \oplus H_2(U, Y_R, rY_R)$ and $Z = (m \| Y_S) \oplus H_3(V)$.

The ciphertext is $\sigma = \langle U, W, Z\rangle$.

De-signcrypt: When a ciphertext $\sigma = \langle U, W, Z\rangle$ is received, receiver R performs the following steps:

1. Compute $V = W \oplus H_2(U, Y_R, x_R U)$.
2. Compute $(m \| Y_S) = Z \oplus H_3(V)$.
3. If $Y_S \notin G_1$, outputs `reject`. Otherwise, compute $H = H_1(m, U, Y_R)$ and check if $e(Y_S, H) = e(P, V)$.
4. If the equation holds, output $\langle m, (U, Y_R, V), Y_S\rangle$; otherwise, output `reject`.

Verify: For a message-signature pair $(m,(U,Y_R,V))$ and a signing key Y_S, the algorithm checks if $e(Y_S,H_1(m,U,Y_R)) = e(P,V)$. If the condition holds, it outputs `true`. Otherwise, it outputs `false`.

The scheme can be viewed as a *sequential* composition of the short signature by [8] and some Diffie-Hellman based encryption scheme. It is called sequential because the signature component V and the 'masking' Z of the message have to be computed in sequence.

3.3 Security Analysis

In [11], it is claimed that the scheme reviewed above is semantically secure against chosen ciphertext insider attack in the model of SC-IND-CCA, existential unforgeable against chosen message insider attack in the SC-EUF-CMA model, and also provides ciphertext anonymity in the model of SC-ANON-CCA.

However, we find that the scheme is not even semantically secure against chosen plaintext attack. That is, with non-negligible advantage (in fact, our attacking technique can break the scheme with overwhelming probability), there exists a PPT adversary $\mathcal{A}$ which can win the game defined in Definition 2 even without querying any of the signcryption oracle and de-signcryption oracle. We will also show that the scheme does not provide ciphertext anonymity either. Below is the attack which compromises the scheme's confidentiality.

Attack Against Confidentiality:
Let $\mathcal{A}$ be an adversary defined in the game of Definition 2. Suppose the public key that $\mathcal{A}$ received from the game challenger is Y_R.

- In the first stage of the game, $\mathcal{A}$ does nothing. That is, $\mathcal{A}$ does not make any query to the signcryption oracle or the de-signcryption oracle.
- After completing the first stage, $\mathcal{A}$ randomly chooses $m_0 \leftarrow \{0,1\}^n$ and sets $m_1 = \overline{m}_0$. That is, m_1 is the complement of m_0. Then, $\mathcal{A}$ randomly picks a private key $x_S \leftarrow \mathbb{Z}_q$ and asks the game challenger for a challenge ciphertext.
- When $\sigma = \langle U,W,Z\rangle$ is received, $\mathcal{A}$ does the following test.

$$(m_0\|Y_S) \stackrel{?}{=} Z \oplus H_3(x_S H_1(m_0,U,Y_R))$$

- If the equation holds, $\mathcal{A}$ outputs a bit b' with value 0. Otherwise, $\mathcal{A}$ outputs 1 for b'.

It is easy to see that $\Pr[b'=0 \mid b=0] = 1$. In the case of $b=1$, let $\mathbf{E}$ be the event that $(m_0\|Y_S) = m_1\|Y_S \oplus H_3(x_S H_1(m_1,U,Y_R)) \oplus H_3(x_S H_1(m_0,U,Y_R))$, or $1^n\|0^\ell = H_3(x_S H_1(m_1,U,Y_R)) \oplus H_3(x_S H_1(m_0,U,Y_R))$. $1^n\|0^\ell$ can be viewed as the distance between the hash values of two different inputs, one involves m_1 and the other one involves m_0. As H_1 and H_3 are viewed as random oracles,

$\Pr[\mathbf{E}] \leq \max(1/2^{\ell}, 1/2^{n+\ell}) = 2^{-\ell}$. We can see that $\Pr[\mathbf{E}] = \Pr[b' = 0 \mid b = 1]$. Hence $\Pr[b' = 1 \mid b = 1] = 1 - \Pr[\mathbf{E}] \geq 1 - 2^{-\ell}$. Therefore,

$$\begin{aligned}\Pr[\mathcal{A} \text{ wins the game}] &= \Pr[b' = 0, b = 0] + \Pr[b' = 1, b = 1] \\ &\geq \frac{1}{2} \cdot 1 + \frac{1}{2}(1 - 2^{-\ell}) \\ &= 1 - 2^{-\ell-1}.\end{aligned}$$

Compromising Ciphertext Anonymity. The attacking technique described above can be extended easily to compromise the ciphertext anonymity of the scheme. For a distinguisher $\mathcal{D}$ described in Definition 4, it also does nothing in the first stage of the game. After obtaining a challenge ciphertext from the game challenger, $\mathcal{D}$ only needs to conduct several rounds of tests similar to that described in the **Attack Against Confidentiality** above. The chance for $\mathcal{D}$ of winning the game is overwhelming.

Note that in either of these attacks, the oracles of signcryption and designcryption are not queried. This also implies that the scheme is not even secure against chosen plaintext insider attack.

These attacks also show that two theorems (Theorem 1 and Theorem 3) in [11] are incorrect. The errors are due to the imprecision of the corresponding proofs. In their proof for Theorem 1, a simulator $\mathcal{B}$ is constructed to simulate the role of the challenger in the SC-IND-CCA game (Definition 2). The proof is to demonstrate that if there exists an adversary $\mathcal{A}$ which can break the SC-IND-CCA security of the scheme, then $\mathcal{B}$ can solve the CDH problem (in other words, given a random instance (aP, bP), calculate abP) with the help of $\mathcal{A}$. $\mathcal{B}$ first sets $\mathcal{A}$'s challenge public key to bP. After getting m_0, m_1 and x_S from $\mathcal{A}$, $\mathcal{B}$ produces a challenge ciphertext $\sigma = \langle U, W, Z\rangle = \langle aP, W, Z\rangle$ where $W \overset{R}{\leftarrow} \{0,1\}^{\ell}$ and $Z \overset{R}{\leftarrow} \{0,1\}^{n+\ell}$. Then the authors claimed: "...$\mathcal{A}$ will not realize the σ is not a valid signcryption for the sender's private key x_S and the public key bP unless it asks for the hash value $H_2(aP, bP, abP)$." But our attack demonstrates that $\mathcal{A}$ can easily verify whether σ is a valid ciphertext or not without querying H_2. The same problem exists in the proof for Theorem 3 in [11].

4 An Improved Signcryption Scheme

The problem of Libert-Quisquater's scheme is that one can judge whether a ciphertext is the signcryption of a specific plaintext once the signing private key of the ciphertext is known. In other words, it does not provide insider security. To solve this problem, we observe that $Z = (m\|Y_S) \oplus H_3(V)$ where V can be obtained from $V \leftarrow W \oplus H_2(U, Y_R, rY_R)$ or $V \leftarrow x_S H_1(m, U, Y_R)$. Knowing either r or x_s is sufficient to break the secrecy of the plaintext "$m\|Y_S$". In order to prevent insider attack, we modify the scheme such that the secrecy of the plaintext does not rely on x_S.

4.1 Improved Libert-Quisquater Signcryption Scheme

The public parameters are the same as the original scheme except that H_3 is modified to $H_3 : G_1^3 \rightarrow \{0,1\}^{n+\ell}$.

Keygen: Same as the original scheme.

Signcrypt: To signcrypt a message $m \in \{0,1\}^n$ for receiver R, sender S conducts the following steps:
1. Pick a random $r \leftarrow \mathbb{Z}_q$ and compute $U = rP$.
2. Compute $V = x_S H_1(m, U, Y_R)$.
3. Compute $W = V \oplus H_2(U, Y_R, rY_R)$ and $Z = (m \| Y_S) \oplus H_3(U, Y_R, rY_R)$.

The ciphertext is $\sigma = \langle U, W, Z \rangle$.

De-signcrypt: When a ciphertext $\sigma = \langle U, W, Z \rangle$ is received, receiver R performs the following steps:
1. Compute $V = W \oplus H_2(U, Y_R, x_R U)$
2. Compute $(m \| Y_S) = Z \oplus H_3(U, Y_R, x_R U)$.
3. If $Y_S \notin G_1$, output `reject`. Otherwise, compute $H = H_1(m, U, Y_R)$ and check if $e(Y_S, H) = e(P, V)$.
4. If the equation holds, output $\langle m, (U, Y_R, V), Y_S \rangle$; otherwise, output `reject`.

Verify: For a message-signature pair $(m, (U, Y_R, V))$ and a signing key Y_S, the algorithm checks if $e(Y_S, H_1(m, U, Y_R)) = e(P, V)$. If the condition holds, it outputs `true`. Otherwise, it outputs `false`.

4.2 Security Analysis of the Improved Scheme

The improved scheme can effectively thwart the attack described in Sec. 3.3. It is also obvious that the scheme satisfies the completeness definition (Definition 1). The following theorems state that the improved scheme is secure in the models defined in Sec. 2.

Theorem 1. *The improved signcryption scheme is SC-IND-CCA secure in the random oracle model under the assumption that Gap Diffie-Hellman Problem is hard.*

Proof. For contradiction, we assume that there exists an adversary $\mathcal{A}$ who wins the game given in Definition 2 with non-negligible advantage. In the following, we construct an algorithm $\mathcal{B}$ to solve the CDH problem in G_1.

Suppose $\mathcal{B}$ is given a random instance of the CDH problem (aP, bP), $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine to find the solution abP. $\mathcal{B}$ sets up a simulation environment for $\mathcal{A}$ as follows:

$\mathcal{B}$ gives bP to $\mathcal{A}$ as the challenging public key Y_u.

$\mathcal{B}$ maintains three lists L1, L2 and L3 to simulate the hash oracles H_1, H_2 and H_3, respectively. In each entry of the lists, it keeps the query and the corresponding return of the oracle.

When a hash query $H_1(m, P_1, P_2)$ is received, where $m \in \{0,1\}^n$ and $P_1, P_2 \in \{0,1\}^\ell$, $\mathcal{B}$ first checks if the query tuple (m, P_1, P_2) is already in L1. If it exists, the existing result in L1 is returned. If it does not exist, $\mathcal{B}$ randomly chooses $t \leftarrow \mathbb{Z}_q$ and returns tP to $\mathcal{A}$ provided that tP is not in L1. Otherwise, $\mathcal{B}$ should keep trying other random values for t until there is no collision found. The query tuple and return value are then saved in L1. For enabling the retrieval of t possibly in some later time of the simulation, the value is also saved in L1.

Hash queries to H_2 or H_3 are handled similarly in the way that randomly chosen values returned cannot be equal to any other value previously returned. Of course, returned values are chosen from the corresponding ranges of these hash functions. There is also one additional step: Let a query tuple be $(P_1, P_2, P_3) \in G_1^3$. If $e(P_1, P_2) = e(P, P_3)$ and $(P_1, P_2, \top)$ is in the corresponding hash list, where '$\top$' is a special symbol, $\mathcal{B}$ replaces '$\top$' in the entry with P_3 and uses the return value of the entry as the value to be returned. The reason will be given shortly.

For a signcryption query on a message m with a receiver's public key Y_R both chosen by $\mathcal{A}$, $\mathcal{B}$ first checks if $Y_R \in G_1$. If it is incorrect or $Y_R = Y_u$, $\mathcal{B}$ returns the symbol '$\perp$' for rejection. Otherwise, $\mathcal{B}$ picks a random $r \leftarrow \mathbb{Z}_q$, computes $U = rP$ and simulates the $H_1(m, U, Y_R)$ hash query described as above. After obtaining t' such that $t'P := H_1(m, U, Y_R)$, $\mathcal{B}$ computes $V = t'(Y_u)$ which is equal to $bH_1(m, U, Y_R)$. $\mathcal{B}$ then simulates H_2 and H_3 as above for obtaining $H_2(U, Y_R, rY_R)$ and $H_3(U, Y_R, rY_R)$, and computes the result ciphertext $\sigma = (U, W, Z)$ according to the description of the improved signcryption scheme.

When $\mathcal{A}$ performs a De-signcrypt(σ, sk_u) query, where $\sigma = (U, W, Z)$, $\mathcal{B}$ looks for tuples of the form (U, Y_u, λ) in L2 and L3 such that $e(P, \lambda) = e(U, Y_u)$. For each of L2 and L3, if the tuple (U, Y_u, λ) does not exist in the list, $\mathcal{B}$ adds a new entry into that particular list by saving $(U, Y_u, \top)$ as the query tuple and a value randomly drawn from the range as the oracle return value, provided that the value is not in the list yet (for preventing collision). The special symbol '$\top$' is used as a marker for denoting that the real value should be the solution of the CDH problem instance (U, Y_u). This step ensures that the values of $H_2(U, Y_u, \lambda)$ and $H_3(U, Y_u, \lambda)$ are *fixed* before σ is de-signcrypted. After that, $\mathcal{B}$ computes $V = W \oplus H_2(U, Y_u, \lambda)$ and $m \| Y_S = Z \oplus H_3(U, Y_u, \lambda)$. Then $\mathcal{B}$ checks if $e(P, V) = e(H_1(m, U, Y_u), Y_S)$ holds where $H_1(m, U, Y_u)$ is simulated as above. If this condition holds, $(m, (U, Y_u, V), Y_S)$ are returned as the message-signature pair and the sender's public key. Otherwise, the symbol '$\perp$' is returned for rejection.

After completing the first stage of the game, $\mathcal{A}$ chooses two n-bit plaintexts m_0 and m_1 together with a sender's private key x_S, and requests $\mathcal{B}$ for a challenge ciphertext built under the receiver's challenging public key Y_u.

$\mathcal{B}$ updates L1 with $H_1(m_0, aP, Y_u)$ and $H_1(m_1, aP, Y_u)$ by executing the simulator for H_1 on these inputs and then sets the challenge ciphertext to $\sigma = (aP, W, Z)$ where W and Z are randomly drawn from distributions. $\mathcal{B}$ answers $\mathcal{A}$'s queries as in the first stage. If $\mathcal{A}$ queries H_2 or H_3 with (aP, Y_u, λ) such that $e(aP, Y_u) = e(\lambda, P)$, then $\mathcal{B}$ outputs λ and halts. If $\mathcal{A}$ halts without making this query, $\mathcal{B}$ outputs a random point in G_1 and halts.

Analysis. Obviously, the running time of $\mathcal{B}$ is in polynomial of $\mathcal{A}$'s running time. To see that the simulated game is computationally indistinguishable from a real game, we note that the simulated game above could never have a collision happen while a real game may have collisions. Other than that, the two games are identical to each other. Suppose the number of hash queries made in one run of the game is at most q_H. It is a polynomial in the security parameter k. Note that ℓ must be no smaller than k as the order of G_1 is k. The probability of having at least one collision is no more than $\frac{q_H(q_H-1)}{2\times 2^k}$ which is negligible. In the following, we analyze $\mathcal{B}$'s success rate.

Let $\mathbf{E}$ be the event that (aP, Y_u, aY_u) is queried on H_2 or H_3. $\bar{\mathbf{E}}$ denotes the event that (aP, Y_u, aY_u) is not queried on H_2 or H_3. Note that $\mathcal{B}$ solves the CDH problem instance in event $\mathbf{E}$.

We claim that for event $\bar{\mathbf{E}}$, $\mathcal{A}$ does not have any advantage in winning the game over random guessing: Let $V_b = x_S H_1(m_b, aP, Y_u)$ for $b = 0, 1$. Then $\sigma = (aP, W, Z)$ is the signcryption of m_0 if the values of $H_2(aP, Y_u, aY_u)$ and $H_3(aP, Y_u, aY_u)$ are $W \oplus V_0$ and $(m_0||Y_s) \oplus Z$, respectively. While σ is the signcryption of m_1 if the values of the two hashes are $W \oplus V_1$ and $(m_1||Y_s) \oplus Z$. Since H_2 and H_3 are not queried with (aP, Y_u, aY_u), due to the random oracle assumption, $\mathcal{A}$ does not have any advantage in determining the oracle returns of H_2 and H_3 on this query tuple. This is because $\mathcal{B}$ has not decided on the oracle returns yet. Hence,

$$\Pr[\mathcal{A} \text{ wins the game } |\bar{\mathbf{E}}] = \frac{1}{2}.$$

From the assumption,

$$\begin{aligned}\Pr[\mathcal{A} \text{ wins the game}] &= \frac{1}{2} + \rho(k) \\ &\leq \Pr[\mathbf{E}] + \frac{1}{2}(1 - \Pr[\mathbf{E}])\end{aligned}$$

where ρ is $\mathcal{A}$'s non-negligible advantage in winning the game defined in Definition 2 and k is the system-wide security parameter. Therefore,

$$\Pr[\mathbf{E}] \geq 2\rho(k)$$

which is non-negligible. □

Theorem 2. *The improved signcryption scheme is SC-EUF-CMA secure in the random oracle model under the assumption that Gap Diffie-Hellman Problem is hard.*

Proof. We prove it also by contradiction, namely if $\mathcal{F}$ can successfully produce a forgery, there exists an algorithm $\mathcal{B}$ that can solve the CDH problem in G_1. After $\mathcal{B}$ is given a random instance of the CDH problem (aP, bP), $\mathcal{B}$ runs $\mathcal{F}$ as a subroutine to find the solution.

$\mathcal{B}$ gives $\mathcal{F}$ bP as the challenge public key Y_u.

$\mathcal{B}$ maintains three lists L1, L2 and L3 to simulate the hash oracles H_1, H_2 and H_3, respectively. In each entry of the lists, it keeps the query and the corresponding return of the oracle. Hash oracles H_2 and H_3 are simulated as in the proof of Theorem 1.

When a hash query $H_1(m, P_1, P_2)$ is asked by $\mathcal{F}$, $\mathcal{B}$ first checks if the query tuple (m, P_1, P_2) is already in L1. If it exists, the existing result in L1 is returned. If it does not exist, $\mathcal{B}$ randomly chooses $t \leftarrow \mathbb{Z}_q$ and returns $t(aP)$ to $\mathcal{F}$ provided that $t(aP)$ is not in L1. Otherwise, $\mathcal{B}$ should keep trying other random values for t until there is no collision found. The query tuple and return value are then saved in L1. For enabling the retrieval of t possibly in some later time of the simulation, the value is also saved in L1.

For a signcryption query on a message m with a receiver's public key Y_R both chosen by $\mathcal{F}$, $\mathcal{B}$ first checks if $Y_R \in G_1$. If it is incorrect or $Y_R = Y_u$, $\mathcal{B}$ returns the symbol '$\perp$' for rejection. Otherwise, $\mathcal{B}$ picks a random $r \leftarrow \mathbb{Z}_q$ and computes $U = rP$. If the tuple (m, U, Y_R) is already defined in L1, $\mathcal{B}$ picks a new random r and recompute U until the tuple (m, U, Y_R) is not in L1 yet. Then $\mathcal{B}$ selects a random $t' \leftarrow \mathbb{Z}_q$ and returns $t'P$ as the value of $H_1(m, U, Y_R)$ provided that $t'P$ is not in L1. Otherwise, $\mathcal{B}$ should keep trying other random values for t' until there is no collision found. The query tuple, oracle return and the value of t' are then saved in L1. After obtaining t' such that $t'P := H_1(m, U, Y_R)$, $\mathcal{B}$ computes $V = t'(Y_u)$ which is equal to $bH_1(m, U, Y_R)$. $\mathcal{B}$ then simulates H_2 and H_3 as in the proof of Theorem 1 for obtaining $H_2(U, Y_R, rY_R)$ and $H_3(U, Y_R, rY_R)$, and computes the result ciphertext $\sigma = (U, W, Z)$ according to the description of the improved signcryption scheme.

When $\mathcal{F}$ performs a De-signcrypt(σ, sk_u) query, where $\sigma = (U, W, Z)$, $\mathcal{B}$ looks for tuples of the form (U, Y_u, λ) in L2 and L3 such that $e(P, \lambda) = e(U, Y_u)$. For each of L2 and L3, if the tuple (U, Y_u, λ) does not exist in the list, $\mathcal{B}$ adds a new entry into that particular list by saving $(U, Y_u, \top)$ as the query tuple and a value randomly drawn from the range as the oracle return value, provided that the value is not in the list yet (for preventing collision). The special symbol '$\top$' is used as a marker for denoting that the real value should be the solution of the CDH problem instance (U, Y_u). This step ensures that the values of $H_2(U, Y_u, \lambda)$ and $H_3(U, Y_u, \lambda)$ are *fixed* before σ is de-signcrypted. After that, $\mathcal{B}$ computes $V = W \oplus H_2(U, Y_u, \lambda)$ and $m \| Y_S = Z \oplus H_3(U, Y_u, \lambda)$. $\mathcal{B}$ then checks if the tuple (m, U, Y_u) is already in L1. If it exists, the existing result in L1 is obtained. If it does not exist, $\mathcal{B}$ simulates the $H_1(m, U, Y_u)$ hash query described as above, which sets the hash value to $t(aP)$, where t is a distinct random element in $\mathbb{Z}_q$. Then $\mathcal{B}$ checks if $e(P, V) = e(H_1(m, U, Y_u), Y_S)$ holds. If this condition holds, $(m, (U, Y_u, V), Y_S)$ are returned as the message-signature pair and the sender's public key. Otherwise, the symbol '$\perp$' is returned for rejection.

When $\mathcal{F}$ produces a ciphertext $\sigma = (U, W, Z)$ and a receiver's key pair (x_R, Y_R), $\mathcal{B}$ de-signcrypts the ciphertext in the same way as the simulation of the de-signcrypt query above. If the forgery is valid, which means (m, V, Y_u) are returned as the message-signature pair and the sender's public key, and $e(P, V) = e(Y_u, H_1(m, U, Y_R))$.

From the simulation of the de-signcrypt query above, we can see that there must be an entry in L1 for $H_1(m, U, Y_R)$. We also claim that the corresponding oracle return in the entry must be in the form $t(aP)$ for some $t \in \mathbb{Z}_q$, which can be retrieved from L1. Notice that if $H_1(m, U, Y_R)$ is equal to tP, which is generated in a signcryption query, the values of W and Z would also have been determined in that signcryption query, which contradicts the restriction of the game defined in Definition 3.

Since $e(Y_u, H_1(m, U, Y_R)) = e(bP, taP) = e(P, V)$, $\mathcal{B}$ can get $V = tabP$ and compute $abP = t^{-1}V$ with the probability equal to the advantage of winning the game by $\mathcal{F}$, which is non-negligible. The running time of $\mathcal{B}$ is also in polynomial of $\mathcal{F}$'s running time. As in the proof of Theorem 1, the simulated game is also computationally indistinguishable from a real game. □

Theorem 3. *The improved signcryption scheme is SC-ANON-CCA secure in the random oracle model under the assumption that Gap Diffie-Hellman Problem is hard.*

Proof. The proof follows that of Theorem 1. Suppose $\mathcal{B}$ is given (aP, cP) as a random instance of the CDH problem, $\mathcal{B}$ runs $\mathcal{D}$ to find the solution.

$\mathcal{B}$ picks two random elements $x, y \in \mathbb{Z}_q$ and sets the two challenge public keys as $pk_{R,0} = x(cP)$ and $pk_{R,1} = y(cP)$. $\mathcal{B}$ then simulates all the hash queries, signcryption queries and de-signcryption queries as in the proof of Theorem 1.

After the completion of the first stage, $\mathcal{D}$ chooses two private keys $sk_{S,0}, sk_{S,1}$ and a plaintext $m \in \{0,1\}^n$ and requests a challenge ciphertext built under $sk_{S,b}$ and $pk_{R,b'}$ where $b, b' \overset{R}{\leftarrow} \{0,1\}$.

$\mathcal{B}$ then updates L1 with $H_1(m, aP, pk_{R,0})$ and $H_1(m, aP, pk_{R,1})$, and returns $\sigma = (aP, W, Z)$ as the challenge ciphertext where W, Z are randomly drawn from the distributions. $\mathcal{B}$ answers $\mathcal{D}$'s queries as in the first stage. If $\mathcal{D}$ queries H_2 or H_3 with $(aP, pk_{R,0}, \lambda)$ such that $e(aP, pk_{R,0}) = e(P, \lambda)$, $\mathcal{B}$ halts and outputs $x^{-1}\lambda$; If $\mathcal{D}$ queries H_2 or H_3 with $(aP, pk_{R,1}, \lambda)$ such that $e(aP, pk_{R,1}) = e(P, \lambda)$, $\mathcal{B}$ halts and outputs $y^{-1}\lambda$. $\mathcal{B}$ halts when $\mathcal{D}$ halts.

Analysis. Obviously, the running time of $\mathcal{B}$ is in polynomial of $\mathcal{D}$'s running time, and the simulated game is computationally indistinguishable from a real game. In the following, we analyze $\mathcal{B}$'s success rate.

Let $\mathbf{E}$ be the event that $(aP, pk_{R,0}, a(pk_{R,0}))$ or $(aP, pk_{R,1}, a(pk_{R,1}))$ has been queried on H_2 or H_3. $\bar{\mathbf{E}}$ denotes event $\mathbf{E}$ does not happen. Note that $\mathcal{B}$ solves the CDH problem instance in event $\mathbf{E}$.

We claim that for event $\bar{\mathbf{E}}$, $\mathcal{D}$ does not have any advantage in winning the game over random guessing: Let $V_{(b,b')} = sk_{S,b}H_1(m, aP, pk_{R,b'})$ for $b, b' \overset{R}{\leftarrow} \{0,1\}$. Then $\sigma = (aP, W, Z)$ is the signcryption of m under $sk_{S,b}$ and $pk_{R,b'}$ if the values of $H_2(aP, pk_{R,b'}, a(pk_{R,b'}))$ and $H_3(aP, pk_{R,b'}, a(pk_{R,b'}))$ are $W \oplus V_{(b,b')}$ and $(m||pk_{S,b}) \oplus Z$, respectively. Since H_2 and H_3 are not queried with $(aP, pk_{R,0}, a(pk_{R,0}))$ or $(aP, pk_{R,1}, a(pk_{R,1}))$, due to the random oracle assumption, $\mathcal{D}$ does not have any advantage in determining the oracle returns of H_2

and H_3 on these query tuples. This is because $\mathcal{B}$ has not decided on the oracle returns yet. Hence,

$$\Pr[\mathcal{D} \text{ wins the game } |\bar{\mathbf{E}}] = \frac{1}{4}.$$

From the assumption,

$$\begin{aligned}\Pr[\mathcal{D} \text{ wins the game}] &= \frac{1}{4} + \rho(k) \\ &\leq \Pr[\mathbf{E}] + \frac{1}{4}(1 - \Pr[\mathbf{E}])\end{aligned}$$

where ρ is $\mathcal{D}$'s non-negligible advantage in winning the game defined in Definition 4 and k is the system-wide security parameter. Therefore,

$$\Pr[\mathbf{E}] \geq \frac{4}{3}\rho(k)$$

which is non-negligible. □

4.3 Performance

As explained at the end of Sec. 3.2, the original Libert-Quisquater signcryption scheme is sequential. Whereas our improved scheme supports parallel computing. In the improved scheme, Z can be computed in parallel with the computations of V and W. Also in a de-signcryption process, 'unwrapping' the signature and revealing the message from a signcryption (Step 1 and 2 of **De-signcrypt** in Sec. 4) can be carried out in parallel. Thus, an implementation may make use of this property to reduce the computation time of signcryption and de-signcryption operations.

5 Conclusion

In this paper, we show that the Libert-Quisquater signcryption scheme cannot achieved the claimed security with respect to SC-IND-CCA (confidentiality) and SC-ANON-CCA (ciphertext anonymity). The scheme is shown to be insecure even in a weaker model, namely, the security against chosen plaintext insider attacks.

Improvement for the scheme is given and security proofs are provided to show that the improved scheme is secure under the strong security models defined (in Sec. 2). We also observe that the improved scheme supports parallel processing for both signcryption and de-signcryption. This feature could be used to reduce the computation time when compared with the original scheme.

Acknowledgement

The authors are grateful to anonymous reviewers for their comments.

References

[1] J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Proc. EUROCRYPT 2002*, pages 83–107. Springer-Verlag, 2002. LNCS 2332.

[2] F. Bao and R. H. Deng. A signcryption scheme with signature directly verifiable by public key. In *PKC'98*, pages 55–59. Springer-Verlag, 1998. LNCS 1431.

[3] J. Beak, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. In *PKC'02*, pages 80–98. Springer-Verlag, 2002. LNCS 2274.

[4] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *Proc. ASIACRYPT 2001*, pages 566–582. Springer-Verlag, 2001. LNCS 2248.

[5] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, 1993. ACM.

[6] D. Boneh. The decision Diffie-Hellman problem. In *Proc. of the Third Algorithmic Number Theory Symposium*, pages 48–63. Springer-Verlag, 1998. LNCS 1423.

[7] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In *Proc. CRYPTO 2001*, pages 213–229. Springer-Verlag, 2001. LNCS 2139.

[8] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Proc. ASIACRYPT 2001*, pages 514–532. Springer-Verlag, 2001. LNCS 2248.

[9] X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *Proc. CRYPTO 2003*, pages 383–399. Springer-Verlag, 2003. LNCS 2729.

[10] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attack. *SIAM J. Computing*, 17(2):281–308, April 1988.

[11] B. Libert and J.-J. Quisquater. Efficient signcryption with key privacy from gap Diffie-Hellman groups. In *PKC'04*, pages 187–200. Springer-Verlag, 2004. LNCS 2947.

[12] J. Malone-Lee and W. Mao. Two birds one stone: Signcryption using RSA. In *Topics in Cryptology - proceedings of CT-RSA 2003*, pages 211–225. Springer-Verlag, 2003. LNCS 2612.

[13] Y. Mu and V. Varadharajan. Distributed signcryption. In *INDOCRYPT 2000*, pages 155–164. Springer-Verlag, 2000. LNCS 1977.

[14] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Proc. CRYPTO 91*, pages 433–444. Springer, 1992. LNCS 576.

[15] R. Steinfeld and Y. Zheng. A signcryption scheme based on integer factorization. In *ISW'00*, pages 308–322. Springer-Verlag, 2000. LNCS 1975.

[16] D. H. Yum and P. J. Lee. New signcryption schemes based on KCDSA. In *Information Security and Cryptology - ICISC 2001*, pages 305–317. Springer-Verlag, 2002. LNCS 2288.

[17] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $<<$ cost(signature) + cost(encryption). In *Proc. CRYPTO 97*, pages 165–179. Springer-Verlag, 1997. LNCS 1294.