Efficient Identity-Based Signcryption from Pairings

Tsz Hon Yuen and Victor K. Wei

Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
thyuen1,kwwei@ie.cuhk.edu.hk

Abstract. We present a new identity based signcryption (IBSC) scheme which involves the least computation of pairing. Our scheme is the fastest, while preserving the maximum security, when we compare it with the existing IBSC schemes. We provides a more strict definition for IBSC security model. We provides choices for using our scheme with linkability of ciphertext or not. Besides, we present the world's first blind signcryption scheme by extending our IBSC scheme and provide security proof for it. Furthermore, our scheme provides practical features like forward secrecy and trusted authority (TA) compatibility.

1 Introduction

Identity based cryptography is a kind of asymmetric key cryptography using the recipient's identity as the public key. In 1984, Shamir [16] firstly proposed the idea of identity based cryptography. Since then, there are many suggestions for the implementation of identity based encryption ([11], [18], [15], [9]). However they are not fully satisfactory. In 2001, Boneh and Franklin [4] proposed the first practical identity based encryption scheme using bilinear pairings on elliptic curves.

The basic idea of identity based cryptography is to use the recipient's identity as the public key. The identity can be name, email address or combining any other strings that can help to identify a person uniquely. Usually a trusted authority (TA) is needed to generate private keys according to the public keys. For example, Alice would like to encrypt a message and send to Bob. Alice can use Bob's identity $\langle Bob, bob@abc.com \rangle$ as Bob's public key to encrypt the message. After Bob receive the ciphertext, Bob uses his private key from TA to decrypt the message. The advantage of identity based cryptography over traditional public key cryptography is that distribution of public key in advance is not needed. Besides, revocation of public key can be achieved by using short-lived time-dependent identities.

Since the first practical identity based encryption scheme was proposed in 2001 [4], there are many new development in identity based cryptography, like identity based signatures [6], authenticated key agreement [17], [8].

Privacy and authenticity are the basic aims of public-key cryptography. We have encryption and signature to achieve these aims. There are many researches for encryption or signature separately. Yet, there are some applications that requires the use of both encryption and signature, like signing an e-mail and then encrypted before sent. Zheng [22] proposed that encryption and signature can be combined as "signcryption" which can be more efficient in computation than running encryption and signature separately. The security of signcryption is discussed by An et al. [1]

1.1 Related Results

Shamir [16] suggested an identity based signature scheme. Boneh and Franklin [4] proposed an identity based encryption scheme. There are some papers [14], [5], [12], [10], [13] concerning the combination of signature and encryption to form a new identity based signcryption scheme (IBSC). The advantage of identity-based signcryption scheme is that it involves less computation and usually has a shorter ciphertext than using encryption and signature scheme separately.

Let us consider the efficiency and proven security of known identity-based signcryption schemes from pairings. The most expensive single operation is the pairing computations. The scheme of Malone-Lee [14], Boyen [5] and Libert and Quisquater second scheme [13] use 5 pairings, while Libert and Quisquater first scheme [12] uses 6, Nalla and Reddy [10] uses 4. The scheme of Boyen is proven secure in a stronger model than Malone-Lee and Libert and Quisquater. The scheme of Nalla and Reddy [10] has no security proof in his paper.

The detailed comparison of our scheme and other schemes will be discussed in chapter 6.

The concept of blind signatures was introduced by Chaum [7], which provides anonymity of users in applications such as e-cash. It allows users to get a signature of a message in a way that the signer learns neither the message nor the resulting signature. Some ID-based blind signature schemes is developed recently [19], [20], [21].

1.2 Contributions

We present an efficient identity based signcryption scheme from pairings. It is the world's most efficient signcryption scheme in terms of the number of the most expensive computations, pairings, that is proven secure.

Our scheme uses the least computation of bilinear pairings, and achieves the maximum security. We provides a new and more strict definition for IBSC security model. We proves that our scheme satisfly this security model, while the existing schemes fail in different parts of our strict security model. The detailed comparison of our scheme with previous work can be found in chapter 6.

In the existing IBSC schemes, there are some ([14], [12]) schemes providing linkability of ciphertext, while some ([5]) provides unlinkability. In this paper, we provide the flexibility for the user to choose whether using linkability or not. Therefore we provide two IBSC schemes in chapter 5.

We have proposed the first world's first blind signcryption scheme by modifing our IBSC scheme. This blind IBSC schemes is secure against one-more forgery attacks for ciphertext.

Our scheme is also the first secure scheme which can provide TA compatibility for sender and recipient. Even if the sender and the recipient using different TAs, they can still uses our scheme to perform signcryption.

Furthermore, our scheme provides forward secrecy such that even if the private key of sender is compromised, the past communications will not be compromised.

1.3 Organization

In chapter 2, we will define an abstract IBSC specification. In chapter 3, we will provide a formal security model for IBSC scheme and blind IBSC scheme. In chapter 4, we will provide definitions on ID-based cryptography and blind IBSC. In chapter 5, we will introduce our IBSC scheme. In chapter 6, we will provide the security analysis for our IBSC scheme. In chapter 7, we will compare our scheme with existing IBSC schemes. In chapter 8, we will introduce the additional functionalities of our scheme.

2 IBSC Specification

An identity based signcryption (IBSC) scheme consists of four algorithms: Setup, Extract, Signcrypt and Unsigncrypt. The functions of the algorithms are specified as follows.

Setup: On input a security parameter k, the TA generates $\langle \zeta, \pi \rangle$ where ζ is the randomly generated master key, and π is the corresponding common public parameter.

Extract: On input ID, the TA computes its corresponding private key S_{ID} (corresponding to $\langle \zeta, \pi \rangle$) and sends back to its owner in a secure channel.

Signcrypt: On input the private key of sender A, S_A, recipient identity ID_B and a message m, outputs a ciphertext σ corresponding to π.

Unsigncrypt: On input the private key of recipient B, S_B, and a ciphertext σ, decrypt to get the sender identity ID_A, the message m and the signature s corresponding to π. Verify s and verify if encryptor = signer and output $\top$ for "true" or $\perp$ for "false".

We make the consistency constraint that if $\sigma \leftarrow Signcrypt(S_A, ID_B, m)$, then $m \leftarrow Unsigncrypt(S_B, \sigma)$.

3 IBSC Security Model

We define a new security model for identity based signcryption. It is a more strict definition than the one by Malone-Lee [14] and Boyen [5]. It includes indistinguishability for message, sender and recipient against adaptive chosen

ciphertext attack and existential unforgeability for ciphertext against adaptive chosen message attack. After that, we will propose a blind version of IBSC and then define the security against one-more forgery for ciphertext attack for blind IBSC.

Indistinguishability for message allows the communicating parties to preserve secrecy for their communication. Indistinguishability for sender and recipient allows the communications appear anonymous against outsiders. Existential unforgeability of ciphertext means that the signature in the ciphertext speaks in the name of sender and the sender cannot deny signcrypting the message. Secure against one-more forgery for ciphertext means that any adversary cannot produce $L+1$ ciphertext from L valid ciphertext.

3.1 Indistinguishability

Indistinguishability for IBSC against adaptive chosen ciphertext attack (IND-IBSC-CCA) is defined as in the following game. It is similar to the indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) for traditional public key encryption scheme.

1. Simulator selects the public parameter and sends to the Adversary.
2. Adversary generates m_1, ID_{A1}, ID_{B1}, and sends to Simulator. Adversary knows SK_{A1}.
3. Simulator generates m_0, ID_{A0}, ID_{B0}, randomly chooses $b \in_R \{0,1\}$. Simulator delivers $\sigma \leftarrow Signcrypt(S_{Ab}, ID_{Bb}, m_b)$, without specifying sender and receiver to Adversary.
4. Adversary tries to compute b, in the following three sub-games
   (a) Simulator ensures $B0 = B1$, $m_0 = m_1$, Adversary computes b.
   (b) Simulator ensures $A0 = A1$, $m_0 = m_1$, Adversary computes b.
   (c) Simulator ensures $A0 = A1$, $B0 = B1$, Adversary computes b.

The Adversary wins the game if he can guess b correctly.

In this game, the Adversary is allowed to query the key extraction oracle, signcryption oracle and unsigncryption oracle adaptively before or after the challenge of the Simulator.

Key extraction oracle $\mathcal{K}.\mathcal{E}.\mathcal{O}.$: Upon input an identity, the key extraction oracle outputs the private key coresponding to this identity.

Signcryption oracle $\mathcal{S}.\mathcal{O}.$: Upon input m, ID_A, ID_B, produce valid signcryption σ for the triple of input.

Unsigncryption oracle $\mathcal{U}.\mathcal{O}.$ Upon input ciphertext σ and receiver ID, the unsigncryption oracle outputs all of the following:

1. decryption result.

2. verification outcome of signature.
3. verification outcome of encryptor=signer.

Oracle query to $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ to extract private key of ID_B is not allowed. Oracle query to $\mathcal{S}.\mathcal{O}.$ for m_1, ID_{A1}, ID_{B1} is not allowed. Oracle query to $\mathcal{S}.\mathcal{O}.$ for the challenge ciphertext from Simulator is not allowed.

Definition 1. *(Indistinguishability) The* advantage *of the adversary is the probability, over half, that he can compute b accurately. The signcryption is secure against* IND-IBSC-CCA *if no PPT adversary has non-negligible advantage in any of the three sub-games.*

In this game, the Adversary is allowed to know the private key of sender S_A of the challenge ciphertext. This gives us a strong *insider-security* for indistinguishability in [1].

Notice that the original definiton for indistinguishability for IBSC in Malone-Lee's [14] paper is similar to the IND sub-game-c (IND-C) here. We combine the security model of "anonymity" for IBSC in Boyen's [5] paper as the IND sub-game-a (IND-A) and IND sub-game-b (IND-B). This new IND definition provides a more comprehensive view of indistinguishability.

3.2 Existential unforgeability

Existential unforgeability against adaptive chosen message attack for identity based signcryption (EU-IBSC-CMA) is defined as in the following game. It is similar to the existential unforgeability against adaptive chosen message attack (EU-CMA) for traditional signature scheme.

1. Simulator selects the public parameter and sends to the Adversary.
2. Adversary delivers valid (σ, ID_B) where σ is not produced by any signcryption oracle query, and Adversary never extracted the secret key of ID_A.

The Adversary wins the game if he can produce a valid tuple (σ, ID_B) that can decrypts, under the private key of ID_B, to a message m, sender identity ID_A and a signature s. It is required that s pass the verification test for ID_A, and σ passes the verification that tests if encryptor = signer.

In this game, the Adversary is allowed to query the key extraction oracle, signcryption oracle and unsigncryption oracle adaptively. The definiton for $\mathcal{K}.\mathcal{E}.\mathcal{O}.$, $\mathcal{S}.\mathcal{O}.$ and $\mathcal{U}.\mathcal{O}.$ are same as above section.

Oracle query to $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ to extract private key of ID_A is not allowed. The Adversary's answer (σ, ID_B) should not be computed by the $\mathcal{S}.\mathcal{O}.$ before.

Definition 2. *(Existential Unforgeability) The* advantage *of the adversary is the probability that he can produce* (σ, ID_B) *to win the above game. The signcryption is secure against* EU-IBSC-CMA *if no PPT adversary has non-negligible advantage in this game.*

The Adversary is allowed to ask the private key of ID_B which is the recipient identity in the Adversary's answer. This condition is necessary to prevent a dishonest recipient to send a ciphertext to himself on Alice's behalf and to try to convince a third party that Alice was the sender. This gives us a strong *insider-security* for existential unforgeability in [1]. It is stronger than Boyen's [5] existential unforgeability in the sense that our model provides non-repudiation for the ciphertext while Boyen's provides non-repudiation for the decrypted signature only.

3.3 Blind Identity-Based Signcryption (BIBSC)

A BIBSC is a five-tuple (Setup, Extract, TransferSigncrypt or TS, Warden, Unsigncrypt) where the Setup, Extract and Unsigncrypt primitives are identical as chapter 2. The other two are defined as follows.

(TS, Warden) is a 4-move interactive protocol with the following inputs and output:

1. Common inputs: sender's identity ID_A, recipient's identity ID_B.
2. Additional input to TS: sender's private key S_A.
3. Additional input to Warden: message m.
4. Output: Warden outputs a ciphertext σ on message m from ID_A to ID_B.

The interactive protocol is as follows:

1. Move-1: TS sends a commit X to Warden.
2. Move-2: Warden challenges TS with h.
3. Move-3: TS sends back the response W and V to Warden.
4. Move-4: Warden outputs a ciphertext σ.

One-more forgery for BIBSC: One-more forgery under chosen message attack for blind identity based signcryption (OMF-BIBSC-CMA) is defined as in the following game. It is similar to the one-more forgery under chosen message attack for traditional blind signature scheme [2], [3], [21]. First of all, we have to define the oracles. The unsigncryption oracle $\mathcal{U}.\mathcal{O}.$ and key extraction oracle $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ are same as before. We have the new interactive blind signcryption oracle $\mathcal{B}.\mathcal{S}.\mathcal{O}.$:

$\mathcal{B}.\mathcal{S}.\mathcal{O}.$: Upon input ID_A, ID_B, it returns a number X. Then inputs a number h. It produces an output (W, V) based on ID_A, ID_B and h.

Then, we define the game as follows:

1. Phase 1: Simulator selects the public parameter and sends to the Adversary.
2. Phase 2: Adversary makes polynomially number of query to the oracles. It makes exactly q_B queries to $\mathcal{B}.\mathcal{S}.\mathcal{O}.$.
3. Phase 3: Adversary delivers $q_B + 1$ triples (ID_{Bi}, σ_i), $1 \leq i \leq q_B + 1$.

The Adversary wins the game if he can produce a all $q_B + 1$ triples (ID_{Bi}, σ_i), $1 \leq i \leq q_B + 1$, that can decrypts, under the private key of ID_{Bi}, to a message m_i, sender identity ID_{Ai} and a signature s_i. It is required that s_i pass the verification test for ID_{Ai}, and s_i passes the verification that tests if encryptor = signer. It is also required that the private key of ID_{Ai} is never extracted by $\mathcal{K}.\mathcal{E}.\mathcal{O}.$.

Definition 3. *(One-more forgery) The* advantage *of the adversary is the probability that he can produce $q_B + 1$ distinct pairs of (ID_{Bi}, σ_i) to win the above game. The signcryption is secure against* OMF-BIBSC-CMA *if no PPT adversary has non-negligible advantage in this game.*

4 Definitions on ID-based cryptography and blind IBSC

The security analysis for indistinguishability and existential unforgeability for the ciphertext are based on the assumption of the hardness of co-GBDH and co-CDH problem respectively. The definition of GBDH is firstly in [13]. Here we extend it to co-GBDH.

Definition 4. *(co-GBDH problem)The co-Gap Bilinear Diffie-Hellman problem is, given P, P^α, P^β in G_1, Q in G_2, for unknown $\alpha, \beta \in Z_q$, to compute $e(P,Q)^{\alpha\beta}$, with the help of an oracle that is able to decide within a unit time whether a tuple $\langle P, P^{\alpha'}, P^{\beta'}, Q, h\rangle \in {G_1}^3 \times G_2 \times G_T$ is such that $h' = e(P,Q)^{\alpha'\beta'}$ or not.*

The co-GBDH assumption is that there is no PPT algorithm solving the co-GBDH problem with non-negligible probability.

Definition 5. *(co-GDH problem) The co-Gap Diffie-Hellman problem is, given $P, P^\alpha \in G_1$, $Q \in G_2$ for unknown $\alpha \in Z_q$, to compute Q^α, with the help of an oracle that is able to decide within a unit time whether a tuple $\langle P, P^{\alpha'}, Q, Q^{\beta'}\rangle \in {G_1}^2 \times {G_2}^2$ is such that $\alpha = \beta$ or not.*

The co-GDH assumption is that there is no PPT algorithm solving the co-GDH problem with non-negligible probability.

The security analysis for one-more forgery is based on the hardness of the chosen target transfer proof of knowledge(TPoK) problem.

Definition 6. *(chosen target TPoK problem) Let the system parameter in* Setup *is known to the Adversary. When a transfer signer gives a vector $\boldsymbol{t}$ of size q_B, the Adversary has to computes vectors $\hat{\boldsymbol{t}}, \hat{\boldsymbol{c}}$ of size $q_B + 1$. The Adversary computes a vector $\boldsymbol{c}$ of size q_B and returns to transfer signer. The transfer signer returns a vector $\boldsymbol{s}$ of size q_B. The Adversary can compute a vector $\hat{\boldsymbol{s}}$ of size $q_B + 1$. In order to achieve this, the Adversary has to find a PPT transform $\boldsymbol{A}$ such that:*

$$\boldsymbol{tA} = \hat{\boldsymbol{t}}$$
$$\boldsymbol{cA} = \hat{\boldsymbol{c}}$$
$$\boldsymbol{sA} = \hat{\boldsymbol{s}}$$

The chosen target TPoK assumption is that there is no PPT algorithm solving the chosen target TPoK problem with non-negligible probability.

5 An Efficient IBSC scheme

In this chapter, we present a new identity based signcryption scheme from the bilinear pairings. This new IBSC scheme is more efficient than the existing IBSC scheme by using less computation. We will first introduce the new IBSC scheme with ciphertext linkability property, and then propose a varient of it having the property of ciphertext unlinkability.

5.1 New IBSC scheme

This IBSC scheme follows the definition in chapter 2. Let G_1, G_2 be two (multiplicative) cyclic groups of prime order p. The bilinear mapping is given as $e : G_1 \times G_2 \to G_T$. Then for all $P \in G_1$, $Q \in G_2$ and for all $a, b \in Z$ we have $e(P^a, Q^b) = e(P, Q)^{ab} = e(P^b, Q^a)$. Now we define our scheme as follows.

Setup: The setup of the TA is similar to the setup in [4]. On inputting a security parameter $n \in N$, the BDH parameter generator $G[1^n]$ will generates G_1, G_2, p and e. The TA chooses an arbitrary generator $P \in G_1^*$ and pick a random $s \in Z_p^*$. Then the TA sets $P_{TA} = P^s$. After that the TA chooses cryptographic hash functions $H_0 : \{0,1\}^* \to G_2^*, H_1 : \{0,1\}^* \times G_2^* \to F_p^*, H_2 : G_T \to \{0,1\}^*, H_3 : G_T \times \{0,1\}^* \to \{0,1\}^*$. The system parameters are:

$$PARAMS = \langle p, G_1, G_2, G_T, e, P, P_{TA}, H_0, H_1, H_2, H_3 \rangle$$

The master-key of TA is s.

Extract: Given a user with identity string $ID \in \{0,1\}^*$. His public key is $Q_{ID} = H_0(ID) \in G_2^*$. His private key S_{ID} is calculated by the TA where $S_{ID} = (Q_{ID})^s$. The private key is sent to the user by a secure channel.

Signcrypt: Suppose that Alice wants to signcrypt a message m to Bob. Divide the signcryption scheme into 2 parts: Sign and Encrypt. Alice firstly signs the message and then encrypts it and sends to Bob.

- Sign: Assume Alice's identity is ID_A. The public key and private key of Alice are Q_A and S_A respectively from Extract. Alice chooses a random $r \in F_p^*$ and then computes:

$$\begin{aligned} X &= P^r \\ h &= H_1(m, X) \oplus ID_B \\ W &= {S_A}^h {Q_A}^r \end{aligned}$$

  Alice outputs the signature $\langle X, W \rangle$ and forwards the parameters $\langle m, r \rangle$ for using in Encrypt.
- Encrypt: Assume Bob's identity is ID_B. Alice computes:

$$\begin{aligned} Q_B &= H_0(ID_B) \\ V &= e({P_{TA}}^r, Q_B) \\ Y &= H_3(V, ID_A) \oplus W \\ Z &= H_2(V) \oplus \langle ID_A, m \rangle \end{aligned}$$

Alice outputs the ciphertext $\sigma = \langle X, Y, Z \rangle$ after encryption and sends to Bob.

Unsigncrypt: Divide the unsigncryption scheme into 2 parts: Decrypt and Verify. Bob receive the ciphertext and decrypt it. After that Bob verify if the signature is indeed come from Alice.

- Decrypt: Assume the private key of Bob is S_B from Extract. Let $\sigma = \langle X, Y, Z \rangle$ be the ciphertext received. Bob decrypts by computing:

$$V' = e(X, S_B)$$
$$\langle ID_A, m \rangle = H_2(V') \oplus Z$$

  Output $\langle ID_A, m \rangle$ together with $\langle X, Y, V' \rangle$ to Verify.

- Verify: Alice verifies the signature by computing:

$$W' = H_3(V', ID_A) \oplus Y$$

  Accept the message if:

$$e(P, W') = e(X{P_{TA}}^h, Q_A) \text{ where } h = H_1(m, X) \oplus ID_B$$

  Output $\top$ if the above verification is true, or output $\perp$ if false.

Note that in chapter 2, the Unsigncrypt requires decryption of the ciphertext, verification of the signature, and verification for checking encryptor = signer. The first two parts are done in the previous steps. The last one is implicitly done in Decrypt and Verify as both of them use the same X in σ to decrypt and verify.

Finally, we show the consistency constraint is satisfied in Decrypt and Verify. In Decrypt, V can be recovered as:

$$\begin{aligned} e(X, S_B) &= e(P^r, {Q_B}^s) \\ &= e(P^{rs}, Q_B) \\ &= e({P_{TA}}^r, Q_B) \end{aligned}$$

In Verify, if the signature is valid, both sides in the verification should be equivalent because:

$$\begin{aligned} e(P, W) &= e(P, {S_A}^h {Q_A}^r) \\ &= e(P, {Q_A}^{(sh+r)}) \\ &= e(P^{(r+sh)}, Q_A) \\ &= e(X{P_{TA}}^h, Q_A) \end{aligned}$$

5.2 Ciphertext unlinkable version

One of the main difference bewteen our scheme in previous section and Boyen's scheme [5] is that our scheme has linkability while Boyen's scheme has unlinkability. As unlinkability may also be important in some applications, we provide the unlinkable version of our scheme.

The only change to our scheme is to change h in Sign into $h = H_1(m, X)$. All other steps remains the same. Therefore this unlinkable version is as efficient as the original one.

Notice that by changing to unlinkable version, unforgeability for ciphertext reduces to unforgeability for signature only. Other security levels remains the same as the linkable version.

5.3 BIBSC version

In this BIBSC, the Setup, Extract and Unsigncrypt are same as chapter 4.1. Now, we describe the interactive protocol for TS and Warden, and also Unsigncrypt.

First of all, TS sends $X = P^r$ to the warden. The warden randomly picks α, β and computes $\hat{X} = X^\alpha P^\beta$, $\hat{h} = H(m, \hat{X})$ and $h = \alpha^{-1}\hat{h}$. The warden sends h to the signer. The signer computes $W = {S_A}^h {Q_A}^r$, $V = e({P_{TA}}^r, Q_B)$ and returns to the warden. The warden computes $\hat{W} = W^\alpha {Q_A}^\beta$ and $\hat{V} = V^\alpha e({P_{TA}}^\beta, Q_B)$. Finally, the warden computes $\hat{Y} = H_3(\hat{V}, ID_A) \oplus \hat{W}$, $\hat{Z} = H_2(\hat{V}) \oplus \langle ID_A, m\rangle$. The warden output the ciphertext $\sigma = \langle \hat{X}, \hat{Y}, \hat{Z}\rangle$.

TS		Warden
randomly choose r		
send $X = P^r$	$\longrightarrow$	
		randomly choose α, β
		computes $\hat{X} = X^\alpha P^\beta$, $\hat{h} = H(m, \hat{X})$
	$\longleftarrow$	send $h = \alpha^{-1}\hat{h}$
send $W = {S_A}^h {Q_A}^r$, $V = e({P_{TA}}^r, Q_B)$	$\longrightarrow$	
		computes $\hat{W} = W^\alpha {Q_A}^\beta$
		computes $\hat{V} = V^\alpha e({P_{TA}}^\beta, Q_B)$
		computes $\hat{Y} = H_3(\hat{V}, ID_A) \oplus \hat{W}$
		computes $\hat{Z} = H_2(\hat{V}) \oplus \langle ID_A, m\rangle$
		outputs $\sigma = \langle \hat{X}, \hat{Y}, \hat{Z}\rangle$

Consistency is verified as:

$$\begin{aligned} e(P, \hat{W}) &= e(P, W^\alpha {Q_A}^\beta) \\ &= e(P, Q_A)^{\alpha(sh+r)+\beta} \\ &= e(P, Q_A)^{s\hat{h}+\alpha r+\beta} \\ &= e({P_{TA}}^{\hat{h}} X^\alpha P^\beta, Q_A) \\ &= e(\hat{X} {P_{TA}}^{\hat{h}}, Q_A) \end{aligned}$$

and

$$\begin{aligned}\hat{V} &= V^{\alpha} e({P_{TA}}^{\beta}, Q_B) \\ &= e({P_{TA}}^{r}, Q_B)^{\alpha} e({P_{TA}}^{\beta}, Q_B) \\ &= e(P^{(r\alpha+\beta)}, {Q_B}^{s}) \\ &= e(X^{\alpha} P^{\beta}, S_B) \\ &= e(\hat{X}, S_B)\end{aligned}$$

Lemma 1. *The BIBSC scheme is blind.*

Proof. Prove in Appendix A.

6 Security analysis

We find that our IBSC scheme in chapter 4.1 satisfy our security models in chapter 3: indistinguishability and existential unforgeability. Let the number of query to random oracle H_i is μ_i for $i = 0, 1, 2, 3$.

The security analysis results are given as follows:

Theorem 1. *Let A be a polynomial time IND-IBSC-CCA attacker that has an advantage $\geq \epsilon$. Then there exist a polynomial time algorithm S that solves the BDH problem with advantage $\geq \epsilon/\mu_0\mu_2$.*

Theorem 2. *Let A be a polynomial time EU-IBSC-CMA attacker that has an advantage $\geq \epsilon$. Then there exist a polynomial time algorithm S that solves the BDH problem with advantage $\geq 2\epsilon/\mu_0{\mu_1}^2\mu_2$.*

The security proof for the linkable version in chapter 4.2 is similar to the above, except the EU-IBSC-CMA is changed to the unforgeability for signature as in [5].

For the BIBSC, we also have the result:

Theorem 3. *Let A be a polynomial time OMF-BIBSC-CMA attacker that has an advantage $\geq \epsilon$. Then there exist a polynomial time algorithm S that solves the chsoen target TPoK with advantage $\geq \epsilon$.*

The security proof for the above theorems will be given in Appendix A.

7 Comparing Performance

In this chapter, we will compare our IBSC scheme will the existing scheme from Malone-Lee(M) [14], Libert and Quisquater scheme 1(LQ1) [12], Nalla and Reddy(NR) [10], Boyen(B) [5] and Libert and Quisquater scheme 2(LQ2) [13]. We also include the Sign-then-Encrypt(StE) and Encrypt-then-Sign(EtS) using ID-based encryption from Boneh and Franklin [4] and ID-based signature from Cha and Cheon [6]. We will compare in terms of security, size of ciphertext and computation time.

For security analysis, we divide into four parts as in chapter 3. The indistinguishability for sender implies anonymity of sender(IND-A). The indistinguishability for recipient implies anonymity of recipient(IND-B). The indistinguishability for message implies message confidentiality(IND-C). The existential unforgeability implies signature non-repudiation(EU). The computation time of IBSC scheme includes the number of bilinear pairings and the number of exponential as they are the most expensive computation in IBSC scheme. The comparisons are summarized in the following table.

Scheme	Security				Ciphertext Size	Signcrypt Time		Unsigncrypt Time	
	IND			EU					
	A	B	C			#pair	#exp	#pair	#exp
EtS	×	√	√	√	$(2k+1)G_1 + 2\|\|m\|\|(+ID)$	1	4 (1)	3	1 (1)
StE	√	√	√	×	$(2k+1)G_1 + 2\|\|m\|\| + ID$	1	4 (1)	3	1 (1)
M [14]	×	√	×	√	$(k+1)G_1 + \|\|m\|\|(+ID)$	1	3 (1)	4	1 (1)
LQ1 [12]	×	×	*	√	$k(G_1 + F_p) + \|\|m\|\|(+ID)$	2	2 (1)	4	1 (1)
NR [10]	×	×	*	×	$(k+1)G_1 + \|\|m\|\|(+ID)$	1	3 (2)	3	1 (1)
B [5]	√	√	√	*	$(k+1)G_1 + \|\|m\|\| + ID$	1	4 (3)	4	2 (2)
LQ2 [13]	√	√	√	*	$(k+1)G_1 + \|\|m+\delta\|\| + ID$	1	4 (3)	4	1 (1)
This scheme	√	√	√	√	$(k+1)G_1 + \|\|m\|\| + ID$	1	4 (1)	3	1 (1)

7.1 Computation Time

The computation of bilinear pairings is the most expensive computation in IBSC scheme. From the above table, we can see that our scheme is the fastest among existing schemes, with similar running time as Nalla and Reddy's [10]. We have the same speed as EtS and StE.

If we look further to the number of exponential computation involved, our scheme is in the middle place in exponential calculation. However, there are some components in our scheme that can be pre-computed. For any random number r, X can be pre-computed in Sign part. Also, U and ${P_{TA}}^r$ can be pre-computed in Encrypt part. All of them can be computed before knowing the recipient identity and message. Therefore the actual number of exponential calculation in our scheme which cannot be pre-computed is two. The actual numbers of exponential calculation of other schemes are shown in the bracket after the original number of exponential calculation in the above table. We can see that our scheme is again the fastest in terms of exponential computation.

Therefore, our scheme is faster than any existing identity based signcryption scheme in terms of the computation of pairings and exponentials.

7.2 Ciphertext Size

For fair comparison on ciphertext size, we assume that a message m of length $||m||$ have to cut into k pieces for signcryption. Also, sender's identity must be known in advance to unsigncryption for the schemes which do not pass IND-A

test. Therefore sender's identity is also included in those schemes. Parameters for signcryption of same m is reused whenever possible.

In LQ2 [13], δ is 160 bits for ciphertext unlinkability, and is 0 bit for ciphertext linkability.

As in the comparison table, we can see that our scheme has the shortest ciphertext size.

7.3 Security

The security analysis follows our definition of security models in chapter 3: IND-A, IND-B, IND-C, UF. We will analysis whether the existing scheme satisfly our security model.

- IND-A: The schemes of Malone-Lee, Libert and Quisquater 1, and Nalla and Reddy are not IND-A secure. It is because the unsigncryption of ciphertext requires the knowledge of sender's identity in advance. Boyen's scheme and our scheme can acheive anonymity of sender.
- IND-B: The schemes of Libert and Quisquater 1, and Nalla and Reddy are not IND-B secure. Any adversary which knows the sender's identity, private key and the message signcrypted can distinguish the identity of the recipient. Cryptanalysis based on IND-B security is given in Appendix B.
- IND-C: Malone-Lee's scheme is not indistinguishability against adaptive CCA secure as shown in [12]. The schemes of Libert and Quisquater 1, and Nalla and Reddy are IND secure according to the security model of Libert and Quisquater. However, they are not secure in Boyen's and our IND security models, where the adversary is allowed to know the private key of sender. Cryptanalysis based on IND-C security is given in Appendix C.
- EU: Nalla and Reddy's scheme is not EU-IBSC-CMA secure in our security model and also security models in all other 3 papers. Any adversary can forge a signcryption from any sender to a recipient ID_B, where the private key of ID_B is known to the adversary.
  Boyen's scheme has unforgeability for the signature only. It does not satisfly the unforgeability for the ciphertext as required in our security model and also the security model of standard signcryption in [1]. It is related to the property of "unlinkability" in Boyen's scheme. Libert and Quisquater 2 scheme is similar to Boyen's in this aspect. Our IBSC scheme avoids this controversial property of unlinkability and achieves unforgeability for ciphertext. Cryptanalysis based on EU security is given in Appendix D.

To conclude, our IBSC scheme achieves the maximum security with the fastest computation time and shortest ciphertext length.

8 Important Functionality of Our Scheme

From our new efficient IBSC scheme, we can achieve further functionalities which are useful in reality. They are the TA compatibility and forward secrecy.

8.1 TA Compatibility

In the reality, it is quite often that the sender and the recipient use different TAs. If this situation happens, our scheme can still be used without major changes.

Assume all TAs use universally agreed bilinear map e, hash functions and $P \in G_1$. Now let Alice uses $TA1$ with master key s_1. Hence $P_{TA1} = P_{s1}$ and $S_A = {Q_A}^{s_1}$. Similarly Bob uses $TA2$ with master key s_2. Hence $P_{TA2} = P_{s2}$ and $S_B = {Q_A}^{s_2}$.

In our scheme, only the parts involving P_{TA} need to be changed to P_{TA1} or P_{TA2}. Therefore the Sign part remains unchanged. In Encrypt, $V = e({Q_B}^r, P_{TA2})$ and others remain unchanged. The Decrypt part remains unchanged. In Verify, $e(P, Y) = e({P_{TA1}}^h X, Q_A)$ and others remain unchanged.

Consistency is verified as:

$$\begin{aligned} V &= e(P_{TA2}, {Q_B}^r) \\ &= e(P^{s2}, {Q_B}^r) \\ &= e(X, S_B) \end{aligned}$$

and:

$$\begin{aligned} e(P, W) &= e(P, {S_A}^h {Q_A}^r) \\ &= e(P, {Q_A}^{(r+hs1)}) \\ &= e({P_{TA1}}^h P^r, Q_A) \\ &= e({P_{TA1}}^h X, Q_A) \end{aligned}$$

The security and efficiency of our scheme remains unaffected. Comparing with the existing schemes, only the scheme of Malone-Lee can be modified to achieve TA compatibility. However this scheme is not adaptive CCA secure. Therefore, our scheme is the only secure scheme which can have the TA compatibility function.

8.2 Forward secrecy

Our scheme can also achieve forward secrecy. It means that even if the private key of the sender is compromised in the future, the past communications will not be compromised. It can be achieved as in our scheme, we make use of the random number r in the computation of pairing:

$$V = e(P_{TA}, {Q_B}^r)$$

which cannot be known even if the private key of the sender is compromised in the future. Therefore the adversary cannot compute V and hence cannot recover m from Z.

If the sender and recipient use different TAs as in chapter 8.1, then our scheme can even achieve partial TA forward secrecy. If the master key of $TA1$ is compromised, then the past communications with users using different TAs will not be compromised. It is because the computation of pairing requires the knowledge of r or s_2:

$$V = e(P_{TA2}, {Q_B}^r) = e({P^{s_2}}, {Q_B}^r)$$

Therefore even $s1$ is compromised in the future, the adversary still cannot compute V and hence cannot recover m from Z.

9 Conclusion

In this paper, we have proposed a new identity based signcryption scheme. Compare with existing scheme, our scheme is the fastest, have maximum security and have a short ciphertext. It is proven secure in a stronger security model than the models in existing schemes. We provide the flexibility for choosing linkability of ciphertext or not.

We proposed the first blind signcryption scheme from the new identity based signcryption scheme. It is secure against one-more forgery attack.

Moreover, our scheme provides practical features of TA compatibility and forward secrecy.

References

1. J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Proc. CRYPTO 2002*, pages 83–107. Springer-Verlag, 2002. Lecture Notes in Computer Science No. 2332.
2. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problem and the security of Chaum's blind signature scheme. *J. of Cryptology*, pages 185–215, 2003.
3. A. Boldyreva. Efficient threshold signature, multisignature, and blind signature schemes based on the Gap-Diffie-Hellman-group signature scheme. In *PKC'03*, pages 31–46. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 567.
4. D. Boneh and M. Franklin. Identity-based encryption from the weil paring. In *Proc. CRYPTO 2001*, pages 213–229. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2139.
5. X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *Proc. CRYPTO 2003*, pages 382–398. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2729.
6. J.C. Cha and J.H. Cheon. An identity-based signature from gap diffie-hellman groups. In *Practice and Theory in Public Key Cryptography – PKC'2003*, pages 18–30. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2567.
7. D. Chaum. Blind signatures for untraceable payments. In *Proc. CRYPTO 82*, pages 199–203. NY, 1983. Plenum.
8. L. Chen and C. Kudla. Identity based authenticated key agreement from pairings. Cryptology ePrint Archive, Report 2002/184, 2002. http://eprint.iacr.org/.
9. C. Cocks. Non-interactive public-key cryptography. In *Cryptography and Coding*, pages 360–363. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2260.
10. K.C. Reddy D. Nalla. Signcryption scheme for identity-based cryptosystems. Cryptology ePrint Archive, Report 2003/066, 2003. http://eprint.iacr.org/.

11. Y. Desmedt and J. Quisquater. Public-key systems based on the difficulty of tampering. In *Proc. CRYPTO 86*, pages 111–117. Springer-Verlag, 1986. Lecture Notes in Computer Science No. 263.
12. B. Libert and J.-J. Quisquater. New identity based signcryption schemes from pairings. IEEE Information Theory Workshop, Paris (France), 2003.
13. B. Libert and J.-J. Quisquater. The exact security of an identity based signature and its applications. Cryptology ePrint Archive, Report 2004/102, 2004. http://eprint.iacr.org/.
14. J. Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. http://eprint.iacr.org/.
15. U. Maurer and Y. Yacobi. Non-interactive public-key cryptography. In *Proc. CRYPTO 91*, pages 498–507. Springer-Verlag, 1991. Lecture Notes in Computer Science No. 547.
16. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. CRYPTO 84*, pages 47–53. Springer-Verlag, 1984. Lecture Notes in Computer Science No. 196.
17. N.P. Smart. An identity based authenticated key agreement protocol based on the weil pairing. Electronic Letters 38, pp.630-632, 2002.
18. S. Tsuji and T. Itoh. An ID-based cryptosystem based on the discrete logarithm problem. *IEEE Journal on Selected Areas in Communication*, 7(4):467–473, 1989.
19. F. Zhang and K. Kim. ID-Based blind signature and ring signature from pairings. In *Proc. ASIACRYPT 2002*, pages 533–547. Springer-Verlag, 2002. Lecture Notes in Computer Science No. 2501.
20. F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In *Proc. ACISP'03*, pages 312–323. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2727.
21. F. Zhang, R. Safavi-Naini, and W. Susilo. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In *Proc. INDOCRYPT03*, pages 191–204. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2904.
22. Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost (encryption). In *Proc. CRYPTO 97*, pages 165–179. Springer-Verlag, 1997. Lecture Notes in Computer Science No. 1294.

A Proof of Security

A.1 IBSC scheme

When we define our IBSC scheme, we subdivide Signcrypt into Sign and Encrypt. We also subdivide Unsigncrypt into Decrypt and Verify. It is used to make the scheme easy to understand and easy to fit in the security model. The definitions are specified as follows.

Sign: On input $\langle S_A, m, ID_B \rangle$, outputs a signature s under π and some ephemeral state data r.

Encrypt: On input $\langle ID_B, m, s, r \rangle$, outputs a ciphertext σ corresponding to π.

Decrypt: On input $\langle S_B, \sigma \rangle$, outputs the sender identity ID_A, the message m and the signature s corresponding to π.

Verify: On input $\langle ID_A, m, s, ID_B \rangle$, outputs $\top$ "true" or $\perp$ "false" indicating if s is a valid signature for message m from sender ID_A.

A.2 Definiton of the Random Oracle Model

Define a security model with entities: Dealer D, simulator S, forger F. They proceed as follows:

1. D generates t instantiations of a hard problem, gives to S to solve.
2. S constructs an anonymous identity based signcryption problem, gives to F to solve.
   Remark: S can obtain key pairs from D which does not help solve the t hard problems instantiations, e.g. these key pairs have zero knowledge about the t instantiations.
3. F proceeds. Allowed to query signcryption oracle $\mathcal{S}.\mathcal{O}.$ or unsigncryption oracle $\mathcal{U}.\mathcal{O}.$ which is simulated/computed by S. Returns an answer.
   Remarks: F is also allowed to query the private key from key extraction oracle $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ with restriction as described belows.
4. S uses F's answer to solve the t hard problem instantiations.

A.3 Proof of Theorem 1

Now D gives $(P, P^{\alpha}, P^{\beta}, Q)$ to S and wants S to compute $e(P,Q)^{\alpha\beta}$. It is assumed that the other parameters p, G_1, G_2 and e are well-known by all entities. S constructs the IND-IBSC-CCA sub-game (a) as in chapter 3, treats F as the adversary A. S sends the system parameter to F with $P_{TA} = P^{\beta}$ as in Setup. S needs to maintain tapes L_0, L_1, L_2 and L_3 that are initially empty and the tapes are used to keep track of queries to random oracles H_0, H_1, H_2 and H_3. The number of queries to these random oracles are restricted to μ_0, μ_1, μ_2 and μ_3 respectively. We assume that any Signcrypt or Unsigncrypt request on a pair of identities happens after F asked the hashing H_0 of these identities. Any key extraction query on an identity is also preceded by a hash query on the same identity. At the end of the game, F returns an answer. S makes use of the tape of the random oracle to solve the co-GBDH problem with non-negligible probability if F can win the IND-IBSC-CCA game with non-negligible probability.

B picks a random number η_Q from $1, 2, ..., \mu_0$. All queries to oracles are subject to the following constraints:
As regards queries to the random oracles:

- Queries on H_0 for a given identity ID are handled as follows:
  - The η_Q-th distinct query to H_0 is back patched to the value Q. The corresponding identity is denoted as ID_Q. Adds the entry $\langle ID_Q, Q\rangle$ to L_0, and returns the public key Q.
  - Otherwise, picks a random $\lambda \in F_p^*$, adds the entry $\langle ID, \lambda\rangle$ to the list L_0, and return the public key $Q_{ID} = P^{\lambda}$.
- Queries on H_1, H_2 and H_3 are handled by producing a randomly sampled element from the codomain, and adding both query and answer to L_1, L_2 and L_3.

As regards to oracle queries for:

- Key extraction oracle $\mathcal{K}.\mathcal{E}.\mathcal{O}.$: suppose S is queried for identity ID_A.
  - If a key extraction query is made on an identity ID_Q, then D terminates its interaction with F, having failed to guess the targeted recipient among those in L_0.
  - Otherwise, S retrieves $\langle ID_A, \lambda_A \rangle$ from L_0 and returns $S_A = (P^\beta)^{\lambda_A}$.
- Signcryption oracle $\mathcal{S}.\mathcal{O}.$: suppose S is given a message m, a sender ID_A, and a recipient ID_B.
  - If ID_A is the identity ID_Q, then S does the following. First, S randomly chooses $r, h \in F_p^*$, and lets $X = P^r(P^\beta)^{-h}$, $W = (Q)^r$. Then, S adds the tuple $\langle m, X, h \oplus ID_B \rangle$ to L_1 in order to force the random oracle $H_1[m, X] = h \oplus ID_B$. Finally, S uses the value of $\langle X, W, m, r, ID_B \rangle$ to run $\mathsf{Signcrypt}$ to produce the desired ciphertext σ.
  - Otherwise, S retrieves $\langle ID_A, \lambda_A \rangle$ from L_0 and computes $S_A = (P^\beta)^{\lambda_A}$. Then S will run $\mathsf{Signcrypt}$ as in the IBSC scheme using S_A. The desired ciphertext σ is returned after $\mathsf{Signcrypt}$.
- Unsigncryption oracle $\mathcal{U}.\mathcal{O}.$: suppose S is given a recipient identity ID_B and a ciphertext $\sigma = \langle X, Y, Z \rangle$.
  - If ID_B is the identity ID_Q, then S does the following. First, search all combinations $\langle ID_A, m, X, W \rangle$ such that $\langle m, X, h_1 \rangle \in L_1$, $\langle V, h_2 \rangle \in L_2$, $\langle V, ID_A, h_3 \rangle \in L_3$, for some h_1, h_2, h_3, V, under the constraints that $h_3 \oplus Y = W$, $h_2 \oplus Z = \langle ID_A, m \rangle$ and $\mathsf{Verify}[ID_A, m, X, W, ID_B] = \top$. Pick a $\langle ID_A, m \rangle$ in one of the combinations above to return as the unsigncrypted plaintext that passed the verification. If no such triple is found, the oracle signals that the ciphertext is invalid.
  - Otherwise, S retrieves $\langle ID_B, \lambda_B \rangle$ from L_0 and computes $S_B = (P^\beta)^{\lambda_B}$. Then S will run $\mathsf{Unsigncrypt}$ as in the IBSC scheme using S_B. $\langle ID_A, m \rangle$ is returned after $\mathsf{Unsigncrypt}$ or $\perp$ is returned to show that the ciphertext is invalid.

Witness Extraction

As in the IND-IBSC-CCA game, at some point F chooses a plaintext m_1, a sender ID_{A1}, and a recipient ID_{B1} on which he wishes to be challenged. The identities ID_{B1} cannot be run in $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ in the previous step. S responds with the challenge ciphertext $\langle X, Y, Z \rangle$, where:

$$X = P^\alpha$$

Y and Z are random strings of appropriate size. All further queries by F are processed adaptively as in the oracles above.

Finally, F returns its final guess, as in the IND-IBSC-CCA game. S ignores the answer from F, picks an entry $\langle V, h_2 \rangle$ uniformly at random in L_2, and returns V as its guess for the solution to the BDH problem.

If the recipient identity ID_{A1} selected by F is the same as ID_Q selected by S, the simulation provided by S is indistinguishable from a genuine attack scenario, except for the challenge ciphertext eventually presented to F.

Since the challenge ciphertext presented to F is randomly distributed in the space of ciphertexts, F cannot gain any advantage in this simulation. Thus, any adversary that has advantage $Adv[F]$ in the real IND-IBSC-CCA game must necessarily recognize with probability at least $Adv[F]$ that the challenge ciphertext provided by S is incorrect.

To recognize that the challenge ciphertext of the form $\langle X, Y, Z\rangle$ with $X = P^{\alpha}$ is incorrect, F needs to query a random oracle query $H_2(V)$ with

$$V = e(X, S_Q) = e(P^{\alpha}, Q^{\beta}) = e(P, Q)^{\alpha\beta}$$

It will leave an entry $\langle V, h_2\rangle$ on L_2, from which B can then extract $V = e(P, Q)^{\alpha\beta}$ with probability $1/\mu_2$.

Taking into account the marginal probability $1/\mu_0$ of the conditioning event that F makes the correct choice for the guessed identity ID_Q, the probability of F correctly solving the co-GBDH problem becomes:

$$Adv[F] = \frac{1}{\mu_0\mu_2}\epsilon$$

where $\epsilon = Adv[A]$ (the advantage of A in the IND-IBSC-CCA game)

Notice that if F does not get the parameters P^{α} and Q together, then the chance of F correctly solving the co-GBDH problem is negligible by the lunchtime attack argument.

The complexity of the above simulation can be controlled by limiting the number of queries in IND-IBSC-CCA game. We can set the number of queries to H_0, H_1, H_2 and H_3 limited to some finite number μ_0, μ_1, μ_2 and μ_3.

A.4 Proof of Theorem 2

Now D gives (P, P^{α}, Q) to S and wants S to compute Q^{α}. It is assumed that the other parameters p, G_1, G_2 and e are well-known by all entities. S constructs the EU-IBSC-CMA game above, and treats F as the Adversary. S sends the system parameter to F with $P_{TA} = P^{\alpha}$ as in Setup. S needs to maintain tapes L_0, L_1, L_2 and L_3 that are initially empty and the tapes are used to keep track of queries to random oracles H_0, H_1, H_2 and H_3. We assume that any Signcrypt or Unsigncrypt request on a pair of identities happens after F asked the hashing H_0 of these identities. Any key extraction query on an identity is also preceded by a hash query on the same identity. At the end of the game, F returns an answer. S makes use of the tape of the random oracle to solve the co-GDH problem with non-negligible probability if F can win the EU-IBSC-CMA game with non-negligible probability.

B picks random numbers η_Q from $1, 2, ..., \mu_0$. All queries to oracles are subject to the following constraints:
As regards queries to the random oracles:

- Queries on H_0 for a given identity ID are handled as follows:

  - The η_Q-th distinct query to H_0 is back patched to the value Q. The corresponding identity is denoted as ID_Q. Adds the entry $\langle ID_Q, Q\rangle$ to L_0, and returns the public key Q.
  - Otherwise, picks a random $\lambda \in F_p^*$, adds the entry $\langle ID, P^\lambda\rangle$ to the list L_0, and return the public key $Q_{ID} = P^\lambda$.
- Queries on H_1, H_2 and H_3 are handled by producing a randomly sampled element from the codomain, and adding both query and answer to L_1, L_2 and L_3.

As regards to oracle queries for:

- $\mathcal{K}.\mathcal{E}.\mathcal{O}.$: suppose S is given an indentity ID_A.
  - If a key extraction query is made on an identity ID_Q, then D terminates its interaction with F, having failed to guess the targeted recipient among those in L_0.
  - Otherwise, S retrieves $\langle ID_A, \lambda_A\rangle$ from L_0 and computes $S_A = (P^\alpha)^{\lambda_A}$.
- $\mathcal{S}.\mathcal{O}.$: suppose S is given a message m, a sender ID_A, and a recipient ID_B.
  - If ID_A is the identity ID_Q, then S does the following. First, S randomly chooses $r, h \in F_p^*$, and lets $X = P^r(P^\gamma)^{-h}$, $W = (P^\alpha)^r$. Then, S adds the tuple $\langle m, X, h \oplus ID_B\rangle$ to L_1 in order to force the random oracle $H_1[m, X] = h \oplus ID_R$. Finally, S uses the value of $\langle X, W, m, r, ID_B\rangle$ to run $\mathsf{Signcrypt}$ to produce the desired ciphertext σ.
  - Otherwise, S retrieves $\langle ID_A, \lambda_A\rangle$ from L_0 and computes $S_A = (P^\gamma)^{\lambda_A}$. Then S will run $\mathsf{Signcrypt}$ as in the IBSC scheme using S_A. The desired ciphertext σ is returned after $\mathsf{Signcrypt}$.
- $\mathcal{U}.\mathcal{O}.$: suppose S is given a recipient identity ID_B and a ciphertext $\sigma = \langle X, Y, Z\rangle$.
  - If ID_B is the identity ID_Q, then S does the following. First, search all combinations $\langle ID_A, m, X, W\rangle$ such that $\langle m, X, h_1\rangle \in L_1$, $\langle V, h_2\rangle \in L_2$, $\langle V, ID_A, h_3\rangle \in L_3$, for some h_1, h_2, h_3, V, under the constraints that $h_3 \oplus Y = W$, $h_2 \oplus Z = \langle ID_A, m\rangle$ and $\mathsf{Verify}[ID_A, m, X, W, ID_B] = \top$. Pick a $\langle ID_A, m\rangle$ in one of the combinations above to return as the unsigncrypted plaintext that passed the verification. If no such triple is found, the oracle signals that the ciphertext is invalid.
  - Otherwise, S retrieves $\langle ID_B, \lambda_B\rangle$ from L_0 and computes $S_B = (P^\alpha)^{\lambda_B}$. Then S will run $\mathsf{Unsigncrypt}$ as in the IBSC scheme using S_B. $\langle ID_A, m\rangle$ is returned after $\mathsf{Unsigncrypt}$ or $\perp$ is returned to show that the ciphertext is invalid.

Witness Extraction

As in the EU-IBSC-CMA game, at some point F returns its final answer (σ, ID_B). It is required that σ must not be computed by the $\mathcal{S}.\mathcal{O}.$ before. S runs a rewind simulation by rewinding back to the time where H_1 is called. S derives the value $T = (W'W^{-1})^{(h'-h)^{-1}}$. S returns the value T to D.

Note that in a true attack scenario, there are two ways to forge a ciphertext:

1. Forge a signature and then run $\mathsf{Encrypt}$ to get a ciphertext.

2. Extract a signature for previous query to $\mathcal{U}.\mathcal{O}.$, then forge an encryption using the valid signature to get a ciphertext.

Now we show that either F produces a successful forgery for the signature or F produces a successful forgery for the encryption, S is in a position to solve the co-GDH problem with non-negligible probability. These include the following cases:

1. F produces a successful forgery for the signature and then gets a ciphertext σ. Then we get W from σ and h from L_1 by Unsigncrypt where:

$$W = {S_A}^h {Q_A}^r$$

After that, we rewind to the time where H_1 is called in this signcryption and runs the signcryption again to get σ'. And after rewind, we get W' from σ' and h' from L_1 where:

$$W' = {S_A}^{h'} {Q_A}^r$$

Note that if the sender identity ID_A selected by F is the same as ID_Q selected by S, then S successfully computes $Q^\alpha = (W'W^{-1})^{(h'-h)^{-1}}$. The identity matches ($ID_A = ID_Q$) with probability $1/\mu_0$. The probability of finding the correct h and h' from H_1 is $1/{\mu_1}^2$.

2. F produces a successful forgery for the encryption. F gets the valid signature from previous query to $\mathcal{S}.\mathcal{O}.$ followed by $\mathcal{U}.\mathcal{O}.$ for the same message. However, the signature contains the information of the recipient. Therefore the valid signature from $\mathcal{S}.\mathcal{O}.$ cannot be used to produce the answer. Hence, the case reduce to produce a successful forgery for the signature.

The probability of F correctly solving the BDH problem becomes:

$$Adv[F] = \frac{\epsilon}{\mu_0 {\mu_1}^2}$$

where $\epsilon = Adv[A]$ (the advantage of A in the EU-IBSC-CCA game)

Notice that if F does not get the parameters Q, then the chance of F correctly solving the co-GDH problem is negligible by the lunchtime attack argument.

The complexity of the above simulation can be controlled by limiting the number of queries in EU-IBSC-CCA game. We can set the number of queries to H_0, H_1, H_2 and H_3 limited to some finite number μ_0, μ_1, μ_2 and μ_3.

A.5 Proof of Lemma 1

To prove the blindness of BIBSC, we show that given a valid ciphertext $\langle \hat{X}, \hat{Y}, \hat{Z} \rangle$ and any view (X, h, W, V), there always exist a unique pair of blinding factors $\alpha, \beta \in Z_q^*$. Since the blinding factors are randomly chosen, the blindness of BIBSC is achieved.

Given a valid ciphertext $\langle \hat{X}, \hat{Y}, \hat{Z} \rangle$, then there exist a unique $(\hat{X}, \hat{W}, \hat{V}, m)$ for this ciphertext. Then for any view (X, h, W, V), the following equations must hold for $\alpha, \beta \in Z_q^*$:

$$\begin{aligned}\hat{X} &= X^{\alpha}P^{\beta}\\ h &= \alpha^{-1}H_1(m,\hat{X})\\ \hat{W} &= W^{\alpha}{Q_A}^{\beta}\\ \hat{V} &= V^{\alpha}e({P_{TA}}^{\beta},Q_B)\end{aligned}$$

From the second equation, we see that there exist a blinding factor α and it is equal to $H_1(m,\hat{X})/h$. For this α, there exist a blinding factor β from the first equation and it is equal to $log_P(\hat{X}X^{-\alpha})$. Therefore we have to show that these blinding factors α,β satisfly the last two equations.

Notice that there exist a S_B which is the private key for Q_B. Then:

$$\begin{aligned}\hat{V} &= e(\hat{X},S_B)\\ &= e(X^{\alpha}P^{\beta},S_B)\\ &= e(X,S_B)^{\alpha}e(P^{\beta},S_B)\\ &= V^{\alpha}e({P_{TA}}^{\beta},Q_B)\end{aligned}$$

Furthermore, $\langle\hat{X},\hat{W},m\rangle$ is a valid signature. Hence:

$$e(\hat{X},Q_A) = e(P,\hat{W})e(P_{TA},Q_A)^{-H_1(m,\hat{X})}.$$

Therefore we have:

$$\begin{aligned}e(P,\hat{W}) &= e(\hat{X},Q_A)e(P_{TA},Q_A)^{H_1(m,\hat{X})}\\ &= e(X^{\alpha}P^{\beta},Q_A)e(P_{TA},Q_A)^{\alpha h}\\ &= e(X{P_{TA}}^{h},Q_A)^{\alpha}e(P^{\beta},Q_A)\\ &= e(P,W)^{\alpha}e(P,{Q_A}^{\beta})\\ &= e(P,W^{\alpha}{Q_A}^{\beta})\end{aligned}$$

Hence, given a valid ciphertext $\langle\hat{X},\hat{Y},\hat{Z}\rangle$ and any view (X,h,W,V), there always exist a unique pair of blinding factors $\alpha,\beta\in Z_q^*$. The blindness of BIBSC is proved.

A.6 Proof of Theorem 3

It is assumed that the parameters p, G_1, G_2 and e are well-known by all entities. S constructs the OMF-BIBSC-CMA game above, and treats F as the Adversary. S sends the system parameter to F as in Setup. S needs to maintain tapes L_0, L_1, L_2 and L_3 that are initially empty and the tapes are used to keep track of queries to random oracles H_0, H_1, H_2 and H_3. S also have a tape L_4 which is used by $\mathcal{B}.\mathcal{S}.\mathcal{O}.$ to record its input/output parameters (X,h,W,V). At the end of the game, F returns an answer. S makes use of the tape L_4 to solve the transfer TPoK problem with non-negligible probability if F can win the OMF-BIBSC-CMA game with non-negligible probability.

Now F has access to the random oracles,$\mathcal{K}.\mathcal{E}.\mathcal{O}.$, $\mathcal{U}.\mathcal{O}.$ and $\mathcal{B}.\mathcal{S}.\mathcal{O}.$. The definition of $\mathcal{U}.\mathcal{O}.$, $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ and the random oracles are similar to previous proof. As regards queries to the random oracles:

- Queries on H_0 for a given identity ID, picks a random $\lambda \in F_p^*$, adds the entry $\langle ID, P^\lambda \rangle$ to the list L_0, and return the public key $Q_{ID} = P^\lambda$.
- Queries on H_1, H_2 and H_3 are handled by producing a randomly sampled element from the codomain, and adding both query and answer to L_1, L_2 and L_3.

As regards to oracle queries for:

- $\mathcal{K}.\mathcal{E}.\mathcal{O}.$: suppose S is given an indentity ID_A, S retrieves $\langle ID_A, \lambda_A \rangle$ from L_0 and computes $S_A = (P^\alpha)^{\lambda_A}$.
- $\mathcal{U}.\mathcal{O}.$: suppose S is given a recipient identity ID_B and a ciphertext $\sigma = \langle X, Y, Z \rangle$. S retrieves $\langle ID_B, \lambda_B \rangle$ from L_0 and computes $S_B = (P^\alpha)^{\lambda_B}$. Then S will run Unsigncrypt as in the BIBSC scheme using S_B. $\langle ID_A, m \rangle$ is returned after Unsigncrypt or $\perp$ is returned to show that the ciphertext is invalid.
- $\mathcal{B}.\mathcal{S}.\mathcal{O}.$: suppose S is given a sender identity ID_A and a recipient identity ID_B. S randomly picks a number $r \in Z_q^*$. S returns $X = P^r$. After that, S retrieves the challenge h. Then S retrieves $\langle ID_A, \lambda_A \rangle$ from L_0 and computes $S_A = (P^\alpha)^{\lambda_A}$. S computes $W = {S_A}^h {Q_A}^r$ and $V = e({P_{TA}}^r, Q_B)$. S puts $\langle X, h, W, V \rangle$ into L_4. S returns (W, V) as the answer to the query.

The number of query to $\mathcal{B}.\mathcal{S}.\mathcal{O}.$ is equal to $L-1$. Eventually F halts and outputs a list of pairs $(\langle \sigma_1, ID_{B_1} \rangle, ..., \langle \sigma_L, ID_{B_L} \rangle)$. For each $1 \leq i \leq L$, F decrypts σ_i to get $\langle \hat{X}_i, \hat{W}_i, \hat{V}_i, m_i, ID_{A_i} \rangle$. F finds $\hat{h_i}$ such that $\langle m_i, \hat{X}_i, \hat{h_i} \rangle \in L_1$. ID_{A_i} should never be the input to the $\mathcal{K}.\mathcal{E}.\mathcal{O}.$. S forms a vector $\hat{\boldsymbol{X}} = [\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_L]$. Similarly S forms vectors $\hat{\boldsymbol{h}}$, $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{V}}$. S retrieves all $\langle X, h, W, V \rangle$ from L_4. S forms them into vectors $\boldsymbol{X}$, $\boldsymbol{h}$, $\boldsymbol{W}$ and $\boldsymbol{V}$ of length $L-1$ each. Then, S can find a PPT transform $\boldsymbol{A}$ such that:

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{X}^{-1}\hat{\boldsymbol{X}} \\ &= \boldsymbol{h}^{-1}\hat{\boldsymbol{h}} \\ &= \boldsymbol{W}^{-1}\hat{\boldsymbol{W}} \\ &= \boldsymbol{V}^{-1}\hat{\boldsymbol{V}} \end{aligned}$$

S returns $\boldsymbol{A}$ as the solution to the chosen target TPoK problem.

Notice that if F can find a new private key and public key pair from the key pairs returned by $\mathcal{K}.\mathcal{E}.\mathcal{O}.$, the F can solve the chosen target CDH problem as in [3]. Therefore, we exclude the possibility of finding extra key pairs in the previous proof.

Then it is easy to see that Theorem 3 is true.

B Cryptanalysis for IND-B

In the following cryptanalysis, please refer to the original paper for original schemes and the definition of the symbols used. In the IND sub-game (b), the Adversary chooses message m, sender identity ID_A and recipient identity ID_{B1}. The Adversary knows the private key of ID_A. Simulator chooses a recipient

identity ID_{B0}, and randomly picks $b \in \{0,1\}$. Simulator signcrypt the message m from sender ID_A to recipient ID_{Bb} and returns the ciphertext to the Adversary. The Adversary has to guess b.

B.1 Libert and Quisquater's scheme 1 [12]

The Adversary has the ciphertext $\langle c, r, S\rangle$ and d_A, the private key of ID_A. The Adversary computes:

$$k_2 = H_2(e(S, Q_{B1})e(d_A, Q_{B1})^r)$$
$$m' = D_{k_2}(c)$$

The Adversary outputs $b = 1$ if $m' = m$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

B.2 Nalla and Reddy's scheme [10]

The Adversary has the ciphertext $\langle R, S, C\rangle$ and S_A, the private key of ID_A. The Adversary computes:

$$R' = (R||H_1(e(Q_{B1}, S_A))||m)$$
$$k_A = H''(e(Q_{B1}, R)^{H'(R')})$$
$$C' = k_A \oplus m$$

The Adversary outputs $b = 1$ if $C' = C$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

C Cryptanalysis for IND-C

In the following cryptanalysis, please refer to the original paper for original schemes and the definition of the symbols used. In the IND sub-game (c), the Adversary chooses message m_1, sender identity ID_A and recipient identity ID_B. The Adversary knows the private key of ID_A. Simulator chooses a recipient identity m_0, and randomly picks $b \in \{0,1\}$. Simulator signcrypt the message m_b from sender ID_A to recipient ID_B and returns the ciphertext to the Adversary. The Adversary has to guess b.

C.1 Malone-Lee's scheme [14]

The Adversary has the ciphertext $\langle c, U, V\rangle$ and S_A. The Adversary computes:

$$r' = H_2(U||m_1)$$

The Adversary outputs $b = 1$ if $e(V, P) = e(Q_A, Q_{TA})^r \cdot e(U, Q_{TA})$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

Notice that this break does not require the private key of sender at all.

C.2 Libert and Quisquater's scheme 1 [12]

The Adversary has the ciphertext $\langle c, r, S\rangle$ and d_A, the private key of ID_A. The Adversary computes:

$$k_2 = H_2(e(S, Q_B)e(d_A, Q_B)^r)$$
$$m' = D_{k_2}(c)$$

The Adversary outputs $b = 1$ if $m' = m_1$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

C.3 Nalla and Reddy's scheme [10]

The Adversary has the ciphertext $\langle R, S, C\rangle$ and S_A, the private key of ID_A. The Adversary computes:

$$R' = (R||H_1(e(Q_B, S_A))||m_1)$$
$$k_A = H''(e(Q_B, R)^{H'(R')})$$
$$C' = k_A \oplus m_1$$

The Adversary outputs $b = 1$ if $C' = C$. Otherwise, the Adversary outputs $b = 0$. Then the Adversary wins the IND game with probability 1.

D Cryptanalysis for EU

In the following cryptanalysis, please refer to the original paper for original schemes and the definition of the symbols used. In the EU game, the Adversary chooses message m, sender identity ID_A and recipient identity ID_B. The Adversary knows the private key of ID_B. The Adversary returns a ciphertext σ and recipient identity ID_B to the Simulator.

D.1 Nalla and Reddy's scheme [10]

The Adversary has S_B, the private key of ID_B. The Adversary randomly chooses $a \in R$ and computes:

$$R = {S_B}^a$$
$$R' = (R||H_1(e(S_B, Q_A))||m)$$
$$S = {Q_B}^{aH'(R')}$$
$$k_A = H''(e(Q_B, S_B)^{aH'(R')})$$
$$C = k_A \oplus m$$

The Adversary outputs the ciphertext $\sigma = \langle R, S, C\rangle$, sender identity ID_A and recipient identity ID_B to the Simulator.

The Simulator decrypts by computing:

$$k_B = H''(e(S, S_B))$$
$$m = k_B \oplus C$$

The decryption succeeds. Then in verification, the Simulator computes $R' = (R||H_1(e(S_B, Q_A))||m)$ and checks if:

$$e(S_B, S) = e(Q_B, R)^{H'(R')}$$

By the above construction, the ciphertext must pass the verification. Then the Adversary wins the EU game with probability 1.

D.2 Boyen's scheme [5]

The Adversary asks the $\mathcal{S}.\mathcal{O}.$ to signcrypt a message m from sender ID_A to recipient ID_C and obtains a ciphertext σ. (Notice that in Boyen's original security model, this oracle query is not allowed. Our security model allows this query in order to provide better security.) The Adversary asks the $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ for the private key of ID_C. The Adversary unsigncrypt σ using the private key of ID_C. The Adversary extracts the signature $\langle j, v\rangle$ from the unsigncryption procedure.

Then the Adversary can forge a signcryption for m from ID_A to some recipient ID_B, where d_B the private key of ID_B is known to the Adversary. The Adversary computes:

$$\begin{aligned} u &= e(i_A, d_B) \\ k &= H_3(u) \\ x &= j^k \\ w &= e(x, d_B) \\ z &= H_4(v) \oplus \langle ID_A, m\rangle \end{aligned}$$

The Adversary outputs the ciphertext $\sigma = \langle x, y, z\rangle$, and recipient identity ID_B to the Simulator. Notice that the above calculation is the same as "EncryptToSelf" algorithm in Boyen's paper. Boyen calls this property as "unlinkability". The ciphertext can be decrypted successfully and pass the verification. Then the Adversary wins the EU game with probability 1. Therefore Boyen's scheme is not secure under our more strict definition of existential unforgeability for ciphertext.

D.3 Libert and Quisquater's scheme 2 [13]

The Adversary asks the $\mathcal{S}.\mathcal{O}.$ to signcrypt a message m from sender ID_A to recipient ID_C and obtains a ciphertext σ. The Adversary asks the $\mathcal{K}.\mathcal{E}.\mathcal{O}.$ for the private key of ID_C. The Adversary unsigncrypt σ using the private key of ID_C. The Adversary extracts the signature $\langle U, V\rangle$ from the unsigncryption procedure.

Then the Adversary can forge a signcryption for m from ID_A to some recipient ID_B, where d_B the private key of ID_B is known to the Adversary. The Adversary randomly picks $\tau \in \{0,1\}^\delta$ and computes:

$$\begin{aligned} x &= H_5(ID_A, ID_B, \tau) \\ X &= U^x \\ W &= V \oplus H_3(X, Q_B, e(X, d_B)) \\ \kappa &= H_4(V) \\ Z &= \mathcal{E}_\kappa(M||ID_A||\tau) \end{aligned}$$

The Adversary outputs the ciphertext $\sigma = \langle x, y, z \rangle$, and recipient identity ID_B to the Simulator. The ciphertext can be decrypted successfully and pass the verification. Then the Adversary wins the EU game with probability 1. Therefore Libert and Quisquater's scheme 2 is not secure under our more strict definition of existential unforgeability for ciphertext.