Fast and Proven Secure Blind Identity-Based Signcryption from Pairings

Abstract. We present the first blind identity-based signcryption (BIBSC). We formulate its security model and define the security notions of blindness and parallel one-more unforgeability (p1m-uf). We present an efficient construction from pairings, then prove a security theorem that reduces its p1m-uf to Schnorr's ROS Problem in the random oracle model plus the generic group and pairing model. The latter model is an extension of the generic group model to add support for pairings, which we introduce in this paper. In the process, we also introduce a new security model for (non-blind) identity-based signcryption (IBSC) which is a strengthening of Boyen's. We construct the first IBSC scheme proven secure in the strenghened model which is also the fastest (resp. shortest) IBSC in this model or Boyen's model. The shortcomings of several existing IBSC schemes in the strenghened model are shown.

1 Introduction

Identity based cryptography is a kind of asymmetric key cryptography using recipient's identity as the public key. In 1984, Shamir [20] firstly proposed the idea of identity based cryptography. Since then, there are many suggestions for the implementation of identity based encryption ([12, 23, 16, 10]). However they are not fully satisfactory. In 2001, Boneh and Franklin [4] proposed the first practical identity based encryption scheme using 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. The advantage is that distribution of public key in advance is not needed.

There are also developments in identity based signatures [6], resp. authenticated key agreement [22, 8], ..., etc. Identity-based encryptions prior to [4], either requires high complexity to compute the key pair or is insecure against colluders.

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 signer learns neither the message nor the resulting signature.

Privacy and authenticity are also the basic aims of public-key cryptography. We have encryption and signature to achieve these aims. Zheng [27] 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]

Contributions This paper makes the following contributions to the literature:

1. We present the first blind identity-based signcryption (BIBSC). Roughly speaking, BIBSC works as follows: Upon request from Warden, a blind signcryption oracle makes a commitment, then blindly signs and computes the randomness term in the encryption for Warden. Warden deblinds the signature and uses the randomness in its encryption to produce a signcryption.
2. We formulate the first BIBSC security models to define security notions including blindness and parallel one-more unforgeability (`p1m-uf`).
3. We construct the first BIBSC scheme from pairings, and prove its security. The blindness of our BIBSC from pairings is statistical ZK, and the `p1m-uf` is reduced to Schnorr's ROS Problem in the random oracle model plus the generic group and pairing model (GGPM).
4. We introduce the generic group and pairing model (GPPM) which is an extension of the generic group model [17, 21, 19] by including support for pairings. We use this model to prove `p1m-uf` of our BIBSC.
5. We also introduce a strengthening of Boyen's [5] security model for (non-blind) identify-based signcryption (IBSC) to add support of authenticated encryption.
6. We construct the first proven secure IBSC in the strengthened model. It is also the fastest (resp. shortest) IBSC in our model as well as in Boyen's [5].
7. Shortcomings of several existing IBSC in the strengthened model are shown.

Organization: In Section 2, we define the preliminaries. In Section 3, we define the IBSC and BIBSC security model. In Section 4, we introduce our schemes. In Section 5, we introduce the generic group and pairing model. In Section 6, we compare our IBSC scheme with existing schemes.

2 Preliminaries

2.1 Related Results

Shamir [20] suggested an identity based signature scheme. Boneh and Franklin [4] proposed an identity based encryption scheme. There are some papers [15, 5, 13, 11, 9, 14] concerning the combination of signature and encryption to form IBSC scheme. The most expensive single operation is pairing computations. The scheme of [15, 5, 14] use 5 pairings, while [13, 9] use 6, [11] uses 4. [5] is proven secure in a stronger model than [15, 13]. [11] has no security proof.

Blind signatures was introduced by Chaum [7]. Some ID-based blind signature schemes is developed recently [24–26].

2.2 Pairings

Our BIBSC and IBSC schemes uses bilinear pairings on elliptic curves. We now give a brief revision on the property of pairings and some candidate hard problems from pairings that will be used later.

Let G_1, G_2, G_3 be cyclic groups of prime order q, writing the group action multiplicatively. Let g_1 (resp. g_2) be a generator of G_1 (resp. G_2). There exist ψ which is isomorphism from G_2 to G_1, with $\psi(g_2) = g_1$.

Definition 1. *A map $e : G_1 \times G_2 \rightarrow G_3$ is called a bilinear pairing if, for all $x \in G_1, y \in G_2$ and $a, b \in Z$, we have $e(x^a, y^b) = e(x, y)^{ab}$, and $e(g_1, g_2) \neq 1$.*

Definition 2. *(co-BDH problem)The co-Bilinear Diffie-Hellman problem is, given $P, P^\alpha, P^\beta \in G_1$, $Q \in G_2$, for unknown $\alpha, \beta \in Z_q$, to compute $e(P, Q)^{\alpha\beta}$.*

Definition 3. *(co-CDH problem) The co-Computational 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.*

2.3 Blind signatures and Schnorr's ROS Problem

Blind signature is described as follows: Upon request from Warden, a signing oracle makes a commitment, then blindly signs a message for Warden. Warden deblinds the signature such that the signing oracle knows neither the message nor the output signature.

Parallel one-more forgery against blind signature is that an attacker interacts for l times with a signer and produces from these interactions $l + 1$ signatures. Schnorr [19] reduced the parallel one-more unforgeability (p1m-uf) of the blind Schnorr signature to the ROS Problem in the random oracle plus generic group model (ROM+GGM). The following are from Schnorr[19]:

Definition 4. *(ROS problem) Find an overdetermined, solvable system of linear equations modulo q with random inhomogeneities. Specifically, given an oracle random function $F : Z_q^l \leftarrow Z_q$, find coefficients $a_{k,i} \in Z_q$ and a solvable system of $l + 1$ distinct equations of Eq. (1) in the unknowns $c_1, \ldots, c_l$ over Z_q:*

$$a_{k,1}c_1 + \ldots + a_{k,l}c_l = F(a_{k,1}, \ldots, a_{k,l}) \text{ for } k = 1, \ldots, t. \quad (1)$$

Theorem 1. *[19] Given generator g, public key h and an oracle for H, let a generic adversary $\mathcal{A}$ performs t generic steps and interacts for l times with a signer. If $\mathcal{A}$ succeeds in a parallel attack to produce $l + 1$ signatures with a probability of success better than $\binom{t}{2}/q$, then $\mathcal{A}$ must solve the ROS-problem in ROM+GGM.*

3 BIBSC and Enhanced IBSC Security Model

We define the first security models for BIBSC and also an enhancement of Boyen's security model for IBSC. For logistics, we present the latter first.

3.1 Enhanced IBSC Security Model

We present an enhancement of Boyen's security model for IBSC. The main addition is to add support for *authenticated encryption* where the signer and encryptor of signcryption are assured to be the same. The signer cannot deny signcrypting the message to the recipient. Boyen's IBSC model is restricted to *ciphertext unlinkability* where this assurance is not required. Our model below is capable of supporting authenticated encryption, resp. ciphertext unlinkability.

3.1.1 Primitives An IBSC scheme consists of four algorithms: (Setup, Extract, Signcrypt, Unsigncrypt). 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 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 private key of recipient B, S_B, and ciphertext σ, decrypt to get sender identity ID_A, message m and signature s corresponding to π. Verify s and verify if encryptor = signer. Output $\top$ for "true" or $\perp$ for "false".

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

3.1.2 Indistinguishability Indistinguishability for IBSC against adaptive chosen ciphertext attack (IND-IBSC-CCA2) is defined as in the following game. The Adversary is allowed to query the random oracles, key extraction oracle, signcryption oracle and unsigncryption oracle. The game is defined as follows:

1. Simulator selects the public parameter and sends to the Adversary.
2. Adversary performs polynomial number of oracle queries adaptively.
3. Adversary generates m_1, ID_{A1}, ID_{B1}, and sends to Simulator. Adversary knows S_{A1}. Simulator generates m_0, ID_{A0}, ID_{B0}, randomly chooses $b \in_R \{0, 1\}$. Simulator delivers $\sigma \leftarrow Signcrypt(S_{Ab}, ID_{Bb}, m_b)$ to Adversary.
4. Adversary performs polynomial number of oracle queries adaptively.
5. 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. The *advantage* of the adversary is the probability, over half, that he can compute b accurately.

The oracles are defined as follows:
Key extraction oracle $\mathcal{KEO}$: Upon input an identity, the key extraction oracle outputs the private key corresponding to this identity.
Signcryption oracle $\mathcal{SO}$: Upon input m, ID_A, ID_B, produce valid signcryption σ for the triple of input.
Unsigncryption oracle $\mathcal{UO}$: Upon input ciphertext σ and receiver ID, the unsigncryption oracle outputs the decryption result, verification outcome of signature and verification outcome of encryptor=signer.

Oracle query to $\mathcal{KEO}$ to extract private key of ID_{B0}, ID_{B1} is not allowed. Oracle query to $\mathcal{SO}$ for m_1, ID_{A1}, ID_{B1} is not allowed. Oracle query to $\mathcal{UO}$ for the challenge ciphertext from Simulator is not allowed.

Definition 5. *(Indistinguishability) The IBSC is* IND-IBSC-CCA2 *secure if no PPT adversary has non-negligible advantage in any of the three sub-games above.*

Our security notion above is a strong one. It incorporates previous security notions including *insider-security* in [1], *indistinguishability* in [15], and *anonymity* in [5].

3.1.3 Existential unforgeability Existential unforgeability against adaptive chosen message attack for IBSC (EU-IBSC-CMA) is defined as in the following game. The Adversary is allowed to query the random oracles, $\mathcal{KEO}$, $\mathcal{SO}$ and $\mathcal{UO}$ adaptively. The definition for oracles are same as above section. The game is defined as follows:

1. Simulator selects the public parameter and sends to the Adversary.
2. Adversary performs polynomially number of oracle queries adaptively.
3. 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 (σ, ID_B) that can be decrypted, under the private key of ID_B, to a message m, sender identity ID_A and a signature s which passes all verification test.

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

Definition 6. *(Existential Unforgeability) A IBSC is secure against* EU-IBSC-CMA *if no PPT adversary has a non-negligible probability in the successful completion of the game above.*

The Adversary is allowed to ask private key of the recipient in Adversary's answer. This gives us a *insider-security* in [1]. It is stronger than Boyen's [5] existential unforgeability in the sense that our model provides non-repudiation for ciphertext while Boyen's provides non-repudiation for decrypted signature only. For *ciphertext unlinkability*, we have to add one more restriction for our model. Oracle query to $\mathcal{SO}$ for ID_A, m in the output using any recipient identity is not allowed. Then the model changes to non-repudiation for signature.

3.2 Introducing BIBSC security model

We will propose the primitives of blind version of IBSC and then define the security notions for blindness and parallel one-more unforgeability.

3.2.1 Primitives A BIBSC is a five-tuple (Setup, Extract, BlindSigncrypt, Warden, Unsigncrypt) where Setup, Extract and Unsigncrypt primitives are identical as primitives in IBSC. (BlindSigncrypt, Warden) is a 3-move interactive protocol. Input to BlindSigncrypt is sender's identity ID_A and private key S_A, and recipient's identity ID_B. Input to Warden is ID_A, ID_B and a message m. The 3-move interactive protocol is as follows:

1. BlindSigncrypt sends a commit X to Warden.
2. Warden challenges BlindSigncrypt with h.
3. BlindSigncrypt sends back the response W and V to Warden.

Finally Warden outputs a ciphertext σ.

3.2.2 Blindness Here we define the blindness of BIBSC scheme. The Adversary is allowed to makes q_B query to blind signcryption oracle $\mathcal{BSO}$, q_H query to random oracles, q_S query to $\mathcal{SO}$, and q_U query to $\mathcal{UO}$. The Adversary keeps the transcripts $\mathcal{T}$ recording the interaction between BlindSigncrypt and Warden.

Definition 7. *(Blindness) A BIBSC is blind if given a ciphertext σ by Warden, $Prob\{\sigma \text{ by } Warden\} = Prob\{\sigma \text{ by } Warden|\mathcal{T}\}$*

3.2.3 Parallel One-more Unforgeability Parallel one-more unforgeability for BIBSC (p1m-uf) is defined as in the following game. It is similar to the one-more forgery for traditional blind signature scheme [2, 3, 26].

1. Sender identity ID_A is given to Adversary.
2. Adversary makes a total of q_B queries to blind signcryption oracles $\mathcal{BSO}_{ID_k}$, $1 \leq k \leq K$, and q_H (resp. q_S) queries to random (resp. Signcryption) Oracle.
3. Adversary delivers $q_B + 1$ tuples (ID_i, m_i, σ_i) to Simulator, $1 \leq i \leq q_B + 1$.

The Adversary wins the game if he can produce $q_B + 1$ valid tuples (ID_i, m_i, σ_i) that can decrypts, under the private key of ID_i, to message m_i and sender identity ID_A. The $\mathcal{SO}, \mathcal{UO}$ and $\mathcal{KEO}$ are same as the one in IBSC. We have the new interactive $\mathcal{BSO}$:
$\mathcal{BSO}_{ID_A}$: Upon input ID_B, it returns a number X. Then inputs a number h. It produces an output (W, V) based on sender ID_A, recipient ID_B, X and h.
It is required that the private key of ID_A is never extracted by $\mathcal{KEO}$. 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.

Definition 8. *(Parallel One-more Unforgeability) The BIBSC is p1m-uf secure if no PPT adversary has non-negligible advantage in this game.*

4 Efficient and Secure BIBSC (resp. IBSC) Schemes

We present our constructions of efficient and secure BIBSC and IBSC schemes from pairings. For logistics of presentation, we present the IBSC first.

4.1 A new efficient and secure IBSC scheme

This IBSC scheme follows the primitives in Section 2. Let G_1, G_2, G_3 be (multiplicative) cyclic groups of order q. The pairings is given as $e : G_1 \times G_2 \rightarrow G_3$. Now we define our scheme as follows.

Setup: The setup of TA is similar to the setup in [4]. On inputting a security parameter $n \in N$, a generator $G[1^n]$ will generates G_1, G_2, G_3, q and e. The TA chooses a generator $P \in G_1$ and pick a random $s \in Z_q$ as master key. Then TA sets $P_{TA} = P^s \in G_1$. After that TA chooses cryptographic hash functions $H_0 : \{0,1\}^* \rightarrow G_2, H_1 : \{0,1\}^* \times G_2 \times \{0,1\}^* \rightarrow Z_q, H_2 : G_3 \rightarrow \{0,1\}^*, H_3 : G_3 \times$

$\{0,1\}^* \rightarrow G_2$. The system parameters are $\langle q, G_1, G_2, G_3, e, P, P_{TA}, H_0, H_1, H_2, H_3\rangle$.

Extract: Given a user identity string $ID \in \{0,1\}^*$, his public key is $Q_{ID} = H_0(ID) \in G_2$. His private key $S_{ID} = (Q_{ID})^s \in G_2$ is calculated by TA.

Signcrypt: Suppose Alice wants to signcrypt a message m to Bob. Alice firstly signs the message and then encrypts it and sends to Bob.

- Sign: Assume Alice's identity is ID_A and Bob's identity is ID_B. The public key and private key of Alice are Q_A and S_A respectively. Alice chooses a random $r \in Z_q$ and computes:

$$\begin{aligned} X &= P^r \in G_1 \\ h &= H_1(m, X, ID_B) \in Z_q \\ W &= {S_A}^h {Q_A}^r \in G_2 \end{aligned}$$

- Encrypt: Alice computes $Q_B = H_0(ID_B) \in G_2$ and:

$$\begin{aligned} V &= e({P_{TA}}^r, Q_B) \in G_3 \\ Y &= H_3(V, ID_A) \oplus W \in G_2 \\ Z &= H_2(V) \oplus \langle ID_A, m\rangle \in \{0,1\}^* \end{aligned}$$

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

Unsigncrypt: Bob receives the ciphertext $\sigma = \langle X, Y, Z\rangle$ and decrypts it. After that Bob verifies if the signature is indeed come from Alice.

- Decrypt: Assume the private key of Bob is S_B. Bob decrypts σ by computing:

$$\begin{aligned} V' &= e(X, S_B) \\ \langle ID_A, m\rangle &= H_2(V') \oplus Z \end{aligned}$$

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

- Verify: Bob computes $W' = H_3(V', ID_A) \oplus Y$. Compare if:

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

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

In Section 3.1, the Unsigncrypt requires decryption of ciphertext, verification of 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: $e(X, S_B) = e(P^r, {Q_B}^s) = e({P_{TA}}^r, Q_B)$. In Verify, if the signature is valid, both sides should be equivalent because:
$e(P, W) = e(P, {S_A}^h {Q_A}^r) = e(P, {Q_A}^{(sh+r)}) = e(P^{(r+sh)}, Q_A) = e(X{P_{TA}}^h, Q_A)$.

Theorem 2. *Our IBSC scheme is* IND-IBSC-CCA2 *secure provided the co-BDH Problem is hard in the random oracle model.*

Theorem 3. *Our IBSC scheme is* EU-IBSC-CMA *secure provided the co-CDH Problem is hard, in the random oracle model.*

Proof sketches of the above two theorems are in Appendix A.

Dual support of ciphertext unlinkability (CU) and authenticated encryption (AE): One of the main difference between our IBSC scheme and Boyen's scheme [5] is that our scheme has linkability (AE) while Boyen's scheme has unlinkability (CU). As unlinkability may also be important in some applications, we provide the CU version of our scheme.

The only change to our scheme is that in Sign change $h = H_1(m, X)$. Other steps remains the same. Therefore this CU version is as efficient as the original AE version. Notice that by changing to CU, unforgeability for ciphertext reduces to unforgeability for signature only, as in [5].

4.2 The first BIBSC scheme

In BIBSC, Setup, Extract and Unsigncrypt are same as Section 4.1. Now, we describe the interactive protocol for BlindSigncrypt and Warden in following table.

BlindSigncrypt		Warden
randomly choose r		randomly choose α, β
send $X = P^r \in G_1$	$\longrightarrow$	
		computes $\hat{X} = X^\alpha P^\beta \in G_1$, $\hat{h} = H(m, \hat{X}, ID_B) \in Z_q$
	$\longleftarrow$	send $h = \alpha^{-1}\hat{h} \in Z_q$
send $W = {S_A}^h {Q_A}^r \in G_2$		
and $V = e({P_{TA}}^r, Q_B) \in G_2$	$\longrightarrow$	
		computes $\hat{W} = W^\alpha {Q_A}^\beta \in G_2$
		computes $\hat{V} = V^\alpha e({P_{TA}}^\beta, Q_B) \in G_3$
		computes $\hat{Y} = H_3(\hat{V}, ID_A) \oplus \hat{W} \in G_2$
		computes $\hat{Z} = H_2(\hat{V}) \oplus \langle ID_A, m\rangle \in \{0,1\}^*$
		outputs $\sigma = \langle \hat{X}, \hat{Y}, \hat{Z}\rangle$

Consistency is verified as:

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

Remark: In our proofs, we use an alternative representation for $\hat{Y}$ and $\hat{Z}$. Let θ_4 (resp. θ_5) be a bijective mapping from G_2 to G_4 (resp. from $\{0,1\}^*$ to G_5) where G_4 (resp. G_5) is a cyclic group. Change $H_2 : G_3 \to G_5, H_3 : G_3 \times \{0,1\}^* \to G_4$. Then $\hat{Y} = H_3(\hat{V}, ID_A) \oplus \theta_4(\hat{W}) \in G_4$ and $\hat{Z} = H_2(\hat{V}) \oplus \theta_5(\langle ID_A, m\rangle) \in G_5$. In Unsigncrypt, we can use θ_4^{-1} and θ_5^{-1} to recover the message. The efficiency and security of BIBSC will not be affected.

Theorem 4. *Our BIBSC scheme has blindness.*

Theorem 5. *Our BIBSC scheme is* p1m-uf *secure provided Schnorr's ROS Problem is hard in the ROM+GGPM.*

5 Generic Group and Pairing Model (GGPM)

We (briefly) introduce the generic group and pairing model (GGPM) by extending the generic group model (GGM) of [17, 21, 18], to include support for the pairing oracle. There are two types of data, namely, group elements and non-group data. Group elements fall into three kinds: elements of G_1, G_2, or G_3. The group cardinalities are prime numbers q_1, q_2, q_3, respectively, with $q_1 = q_2 = q_3 = q$. Non-group data are integers in Z (or in Z_q, Z_q, Z_q respectively, depending on convention). The group elements of G_3 can be randomly generated, obtained from the blind signcryption oracle, or computed as the pairing of one element from G_1 and an element from G_2. The GGPM, in its basic form, consists of:

1. three GGM's, one for each of G_1, G_2, and G_3. Denote their *encodings* by $\theta_i : G_i \to S_i$, $i = 1, 2, 3$. With usual assumptions such that generic algorithm is banned for some non-group operations. (In ECDL with parameters p, G, q, the group is $(< G >, *)$, and arithmetics in Z_p are explicitly banned in GGM. In DL with parameters p, q, g, the group is $(< g >, *)$, and additions (resp. subtractions) in Z_p are banned in GGM.)
2. a pairing oracle, $\hat{e} : S_1 \times S_2 \to S_3$, satisfying bilinear properties.
3. Other oracles in the security model such as Blind Signcryption Oracle $\mathcal{BSO}$, Key Extraction Oracle $\mathcal{KEO}$ and Random Oracle.

Here three GGM's definitions are similar to Schnorr's GGM [18]. So we only focus on the new pairing model. Only a restricted set of operations for group elements are allowed. Each *generic step* is a computation of one of the following:

mex-1: $Z_q^{d_1} \times G_1^{d_1} \to G_1, (a_1^{(1)}, \cdots, a_{d_1}^{(1)}, g_1^{(1)}, \cdots, g_{d_1}^{(1)}) \mapsto \prod_i (g_i^{(1)})^{a_i^{(1)}}$

mex-2: $Z_q^{d_2} \times G_2^{d_2} \to G_2, (a_1^{(2)}, \cdots, a_{d_2}^{(2)}, g_1^{(2)}, \cdots, g_{d_2}^{(2)}) \mapsto \prod_{i'} (g_{i'}^{(2)})^{a_{i'}^{(2)}}$

mex-3: $Z_q^{d_3+d_1d_2} \times G_3^{d_3} \times G_1^{d_1} G_2^{d_2} \to G_3$,

$(a_1^{(3)}, \cdots, a_{d_3+d_1d_2}^{(3)}, g_1^{(3)}, \cdots, g_{d_3}^{(3)}, (g_1^{(1)}, g_1^{(2)}), \cdots, (g_{d_1}^{(1)}, g_{d_2}^{(2)}))$

$\mapsto \prod_{i=1}^{d_3} (g_i^{(3)})^{a_i^{(3)}} \prod_{j=1}^{d_1} \prod_{k=1}^{d_2} e(g_j^{(1)}, g_k^{(2)})^{a^{(3)}_{d_3+d_2(j-1)+k}}$

mex-p : $Z_q^{d_1+d_2} \times G_1^{d_1} \times G_2^{d_2} \to G_3$,

$(a_1^{(4)}, \cdots, a_{d_1}^{(4)}, a_1^{(5)}, \cdots, a_{d_2}^{(5)}, g_1^{(1)}, \cdots, g_{d_1}^{(1)}, g_1^{(2)}, \cdots, g_{d_2}^{(2)}) \mapsto \prod_j \prod_k e(g_j^{(1)}, g_k^{(2)})^{a_j^{(4)} a_k^{(5)}}$

The elements $g_i^{(1)}$'s consist of P, P_{TA}, $\mathcal{BSO}$ commitments X_i's, and randomly generate G_1 elements. The elements $g_i^{(2)}$'s consist of Q_{ID}'s, S_{ID}'s, $\mathcal{BSO}$ responses W_i's, and randomly generate G_2 elements. The elements $g_i^{(3)}$'s consist of $\mathcal{BSO}$ responses V_i's and randomly generate G_3 elements. Similar to Schnorr [19], we omit randomly generated group elements below w.l.o.g.

A (non-interactive) **generic algorithm** is a sequence of t_{total} generic steps

1. Inputs are: $f_1^{(u)}, \cdots, f_{t'_u}^{(u)} \in G_u$ for $u = 1, 2, 3$, $1 \leq t'_u < t_{total}$, where $t' = \sum_u t'_u < t_{total}$ and non-group data like Z_q in given ciphertext or signature.
2. Computation steps are: $f_i^{(u)} = \prod_{j=1}^{i-1} (f_j^{(u)})^{a_{i,j}^{(u)}}$, for $i = t'_u + 1, \cdots, t_u$, $u = 1, 2$, and $f_i^{(3)} = \prod_{j=1}^{i-1} (f_j^{(3)})^{a_{i,j}^{(3)}} \cdot \prod_{1 \leq k, \ell < t} e(f_k^{(1)}, f_\ell^{(2)})^{b_{i,k,\ell}}$ for $i = t'_3 + 1, \cdots, t_3$, where $t_{total} = t_1 + t_2 + t_3 + t_4$ and exponents $a_{i,j}^{(u)}$ depends arbitrarily on i, j, and non-group inputs.
3. Ouputs are: non-group data and group elements $f_{\sigma_1}^{(u)}, \cdots, f_{\sigma_d}^{(u)}$where the integers $\sigma_1, \cdots, \sigma_d \in \{1, \cdots, t_u\}$ that depend arbitrarily on the non-group input.

The generic adversary can also perform equality test, if-then-else, looping, and other logical operations. We omit discussions about them here.

In the generic algorithm, each computation step $f_\sigma^{(u)}$ must be represented as the product of powers of group elements $g_i^{(1)}$'s, $g_{i'}^{(2)}$'s, $g_{i''}^{(3)}$'s, and $e(g_k^{(1)}, g_\ell^{(2)})$'s. There are only polynomially many group elements involved in any PPT algorithm. Each step can be represented as a sequence of exponents, and that representation should be unique. A *collision* is when a step can have multiple representations w.r.t. the bases consisting of the prescribed set of group elements. The following lemma shows the *collision* probability for $f_i^{(1)}, f_j^{(2)}, f_k^{(3)}$ are negligible except when involving oracle queries. The proof is similar to Schnorr's Lemma 1 and omitted.

Lemma 1. *The probability of a PPT generic algorithm being able to compute a collision is negligible, except those collisions obtain via oracle queries.*

The only non-negligible collisions are obtained from the blind signcryption oracle which are of the type $e(A, B) = e(C, D)$ in G_3.

Next we elaborate on **interactive generic algorithms**. We count the following generic steps:

- group operations mex-1, mex-2, mex-3, mex-p
- queries to hash oracle H
- queries to key extraction oracle $\mathcal{KEO}$
- interactions with a blind signcryption oracle $\mathcal{BSO}$.

A **generic adversary** is an interactive algorithm that interacts with $\mathcal{BSO}$. The construction is similar to Schnorr's, unless specified below. The *input* consists of generators $g^{(1)}, g^{(2)}, g^{(3)}$, public keys $Q_1, \cdots, Q_K \in G_2$, master public key $P_{TA} \in G_1$, group order q, pairing $e(\cdot, \cdot)$ and collection of messages, ciphertexts and so on, which can be broken into group elements and non-group data.

$\mathcal{A}$'s *transmission* to $\mathcal{KEO}$ depends arbitrarily on given group elements and non-group data. Notice that key extraction for sender's private key is not allowed.

The *restriction* is that $\mathcal{A}$ can use group elements only for generic group operations, equality tests and for queries to hash oracle and $\mathcal{KEO}$, whereas non-group data can be arbitrarily used without charge. The computed group elements are given as explicit multiplicative combinations of given group elements. Let $X_\ell = {g^{(1)}}^{r_\ell} \in G_1, W_\ell = {Q_A}^{r_\ell + sh_\ell} \in G_2, V_\ell = e(X_\ell, S_{B_\ell})$ for $\ell = 1, \cdots, l$ be the group elements that $\mathcal{A}$ gets from $\mathcal{BSO}$ using the sender ID_A and recipient ID_{B_ℓ}. A computed $f_j^{(1)} \in G_1$ is of the form $f_j^{(1)} = P^{a_{j,-1}^{(1)}} P_{TA}^{a_{j,0}^{(1)}} \prod_{\ell=1}^{l} X_\ell^{a_{j,\ell}^{(1)}}$, where the exponents $a_{j,-1}^{(1)}, \cdots, a_{j,l}^{(1)} \in Z_q$ depend arbitrarily on given non-group data. A computed $f_j^{(2)} \in G_2$ is of the form $f_j^{(2)} = Q_A^{a_{j,0}^{(2)}} \prod_{\ell=1}^{l} W_\ell^{a_{j,\ell}^{(2)}}$, where the exponents depend arbitrarily on given non-group data. A computed $f_j^{(3)} \in G_3$ is of the form $f_j^{(3)} = e(P, Q_A)^{a_{j,-1}^{(3)}} e(P_{TA}, Q_A)^{a_{j,0}^{(3)}} \prod_{\ell=1}^{l} V_\ell^{a_{j,\ell}^{(3)}}$.

Powers and limitations of GGM and GGPM Lemma 1 implies that co-CDH is hard. The perspective is that co-CDH constitutes collisions in GGPM. The real-world interpretation of this model-based result is roughly as follows: GGM (resp. GGPM) *bans* certain operations, in the sense that it can be assumed w.l.o.g. that the generic algorithm does not use these operations. The justification is that these operations are thought to be of no help. In GGM for discrete logarithm with parameters p, q, g, the additions (resp. subtractions) in Z_p are banned. In GGM for ECDL with parameters p, q, base point G whose order is q, arithmetics in Z_p are banned. In GGPM where we have in mind the G_1, G_2, and G_3 are all groups of elliptic curve points, the GGPM model allows point operations, arithmetics in Z_q, but bans arithmetics in Z_p on the argument that they do not help.

Based on such model assumptions, GGM has be used to prove results that often cannot be proved in other models. The GGM has been used to prove the hardness of the discrete logarithm [17, 21]. It has also been used to reduce p1m-uf of Schnorr or Okamoto-Schnorr blind signature to the ROS Problem [19], or the one-more discrete logarithm problem. Note that the one-more discrete logarithm problem is proven hard in the GGM by simple applications of the methods used in [18]. Based on similar model assumptions, we use GGPM to reduce p1m-uf of blind signcryption to the ROS Problem or the one-more co-CDH Problem in this paper. Note that one-more co-CDH is proven hard in GGPM.

Algorithms already exist that exploits operations *banned* from GGM. The index calculus method to compute the discrete logarithm utilizes size information in Z_p to achieve efficiency. It is outside the boundary of GGM. In ECDL, it is suspected but not yet explicitly demonstrated that arithmetics in Z_p and properties of the curve can be exploited. Therefore, GGM and GGPM are used with these elliptic curves applications in mind. If and when exploitations of Z_p arithmetics or curve properties, or other unforeseen techniques outside the model, can be exhibited, both GGM and GGPM will need to be reexamined.

Lemma 1 also implies the hardness of the one-more co-CDH Problem in the GGPM. The *one-more co-CDH Problem* is (roughly speaking): Given q_B queries to the co-CDH Oracle, compute $q_B + 1$ co-CDH Problems.

6 Comparing Performance

In this Section, we will compare our IBSC scheme with existing schemes. We will compare in terms of security, size of ciphertext and computation time.

For security analysis, we divide into the followings: IND-A implies anonymity of sender. IND-B implies anonymity of recipient. IND-C implies message confidentiality. EU implies ciphertext non-repudiation. The computation time includes the number of pairings and exponential computation as they are the most expensive in IBSC scheme. The actual number of computation which cannot be pre-computed is shown in bracket. 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	$\times$	$\surd$	$\surd$	$\surd$	$(2k+1)G_1 + 2\|\|m\|\|(+ID)$	1	4 (1)	3	1 (1)
StE	$\surd$	$\surd$	$\surd$	$\times$	$(2k+1)G_1 + 2\|\|m\|\| + ID$	1	4 (1)	3	1 (1)
M [15]	$\times$	$\surd$	$\times$	$\surd$	$(k+1)G_1 + \|\|m\|\|(+ID)$	1	3 (1)	4	1 (1)
LQ1 [13]	$\times$	$\times$	*	$\surd$	$k(G_1 + F_p) + \|\|m\|\|(+ID)$	2	2 (1)	4	1 (1)
NR [11]	$\times$	$\times$	*	$\times$	$(k+1)G_1 + \|\|m\|\|(+ID)$	1	3 (2)	3	1 (1)
B [5]	$\surd$	$\surd$	$\surd$	*	$(k+1)G_1 + \|\|m\|\| + ID$	1	4 (3)	4	2 (2)
CYSC [9]	$\times$	$\surd$	$\surd$	$\surd$	$k(G_1 + F_p) + \|\|m\|\|(+ID)$	2	2 (1)	4	1 (1)
LQ2 [14]	$\surd$	$\surd$	$\surd$	*	$(k+1)G_1 + \|\|m + \delta\|\| + ID$	1	4 (3)	4	1 (1)
This scheme	$\surd$	$\surd$	$\surd$	$\surd$	$(k+1)G_1 + \|\|m\|\| + ID$	1	4 (1)	3	1 (1)

As we can see, our IBSC scheme is the fastest, with shortest ciphertext size and proven secure in the strongest model among the existing schemes. Detailed comparison will be given in Appendix B.

Additional functionalities 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.

TA Compatibility. In reality, sender and recipient may use different TAs. If it happens, our scheme can still be used with slight changes. Assume all TAs use same pairing e, hash functions and $P \in G_1$. Now let Alice uses TA_1 with master key s_1. and Bob uses TA_2 with master key s_2. In Encrypt, change $V = e({Q_B}^r, P_{TA_2})$. In Verify, $e(P, Y) = e({P_{TA_1}}^h X, Q_A)$. Others remain unchanged.

Forward secrecy. Our scheme can achieve forward secrecy. It is implied by IND-CCA2. If sender and recipient use different TAs, then it can even achieve partial TA forward secrecy. If master key of $TA1$ is compromised, then past communications with users using different TAs will not be compromised, since the adversary still cannot compute V.

7 Conclusion

In this paper, we have proposed a new BIBSC scheme and its security model. We introduce the generic group and pairing model (GGPM). We proof the BIBSC scheme is secure against p1m-uf in ROM+GGPM.

For the IBSC scheme, our scheme is the fastest, have a short ciphertext and proven secure in a stronger security model when comparing with existing scheme. We provide the flexibility for choosing linkability of ciphertext or not.

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. S. Chow, S.M. Yiu, L. Hui, and K.P. Chow. Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity. In *ICISC 2003*, pages 352–369. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2971.
10. C. Cocks. Non-interactive public-key cryptography. In *Cryptography and Coding*, pages 360–363. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2260.
11. K.C. Reddy D. Nalla. Signcryption scheme for identity-based cryptosystems. Cryptology ePrint Archive, Report 2003/066, 2003. http://eprint.iacr.org/.
12. 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.
13. B. Libert and J.-J. Quisquater. New identity based signcryption schemes from pairings. IEEE Information Theory Workshop, Paris (France), 2003.
14. 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/.
15. J. Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. http://eprint.iacr.org/.
16. 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.
17. V.I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes 55*, pages 165–172, 1994.
18. C. P. Schnorr. Practical security in public-key cryptography. In *Proc. ICISC*. Springer, 2001. Lecture Notes in Computer Science.

19. C. P. Schnorr. Security of blind discrete log signatures against interactive attacks. In *Proc. ICISC*, pages 1–12. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2229.
20. 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.
21. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Proc. EUROCRYPT 97*, pages 256–266. Springer-Verlag, 1997. Lecture Notes in Computer Science No. 1233.
22. N.P. Smart. An identity based authenticated key agreement protocol based on the weil pairing. Electronic Letters 38, pp.630-632, 2002.
23. 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.
24. 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.
25. 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.
26. 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.
27. 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 Sketch of Security

A.1 Proof Sketch of Theorem 2

Setting up: Dealer D gives $(P, P^{\alpha}, P^{\beta}, Q)$ to Simulator S and wants S to compute $e(P,Q)^{\alpha\beta}$. S sends the system parameter to F with $P_{TA} = P^{\beta}$ as in Setup. S picks a random number η_Q from $\{1, 2, ..., \mu_0\}$, where μ_0 is the number of query to H_0.

Simulating Oracles: As regards queries to the random oracles:

- Query on H_0 for identity ID is 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 tape 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 tape 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 random element from the codomain, and adding both query and answer to tape L_1, L_2 and L_3.

As regards to oracle queries for:

- $\mathcal{KEO}$: For input identity ID_A.

  - If $ID_A = ID_Q$, 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 $S_A = (P^\beta)^{\lambda_A}$.
- $\mathcal{SO}$: For input message m, sender ID_A, and recipient ID_B.
  - If $ID_A = ID_Q$, then 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 to force the random oracle $H_1(m, X) = h \oplus ID_B$. Finally, S uses $\langle X, W, m, r, ID_B \rangle$ to run 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 Signcrypt using S_A and get ciphertext σ.
- $\mathcal{UO}$: For input recipient ID_B and ciphertext $\sigma = \langle X, Y, Z \rangle$.
  - If $ID_B = ID_Q$, then S searches 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 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 answer. If no such tuple 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 Unsigncrypt using S_B to get $\langle ID_A, m \rangle$ or $\perp$.

Witness Extraction: As in the IND-IBSC-CCA2 game, at some point F chooses plaintext m_1, sender ID_{A1}, and recipient ID_{B1} on which he wishes to be challenged. S responds with 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. S ignores the answer from F, randomly picks an entry $\langle V, h_2 \rangle$ in L_2, and returns V as the solution to the co-BDH problem.

If the recipient identity $ID_{A1} = ID_Q$ selected by S, to recognize the challenge ciphertext $\langle X, Y, Z \rangle$ with $X = P^\alpha$ is incorrect, F needs to query random oracle $H_2(V)$ 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 $V = e(P, Q)^{\alpha\beta}$. □

A.2 Proof Sketch of Theorem 3

Setting up: Dealer D gives (P, P^β, Q) to Simulator S and wants S to compute Q^β. Others are same as in the proof sketch of Theorem 2.

Oracle Simulation: The signcryption oracle, the unsigncryption oracle, and the key extraction oracle are simulated in the same way as in the proof of Theorem 2.

Witness Extraction: Assume $\mathcal{F}$ is a PPT forger. Rewind $\mathcal{F}$ to the random oracle query whose output appears in the verification in unsigncryption. Then we obtain $W = S_A^h Q_A^r$ and $W' = S_A^{h'} Q_A^r$ in respective forks. Combining, we can compute the co-CDH Problem if $Q_A = Q$. Then $Q^\beta = S_A = (W'/W)^{(h'-h)^{-1}}$. □

A.3 Proof Sketch of Theorem 4

To prove the blindness of BIBSC, we show that given a valid ciphertext $\langle \hat{X}, \hat{Y}, \hat{Z} \rangle$ and any transcript of blind signcryption (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 transcript of blind signcryption (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 $\alpha = H_1(m, \hat{X})/h$. For this α, there exist a blinding factor β from the first equation and $\beta = log_P(\hat{X} X^{-\alpha})$. Therefore we have to show that these blinding factors α, β satisfy the last two equations.

Notice that there exists 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. 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 transcript of blind signcryption (X, h, W, V), there always exists a unique pair of blinding factors $\alpha, \beta \in Z_q^*$. Therefore, $Prob\{\sigma \ by \ Warden\} = Prob\{\sigma \ by \ Warden | \mathcal{T}\}$. The blindness of BIBSC is proved. □

A.4 Proof Sketch of Theorem 5

This section refers to a generic adversary $\mathcal{A}$ performing some t generic steps, including some q_B interactions $(X_1, h_1, W_1, V_1), \cdots, (X_{q_B}, h_{q_B}, W_{q_B}, V_{q_B})$ with

$\mathcal{BSO}$, producing some $t'^{(u)}$ group elements in G_u. We let $\boldsymbol{r} = (r_1, \cdots, r_{q_B})$ denote $\mathcal{BSO}$ random coins. Let $f_1 = P, f_2 = P_{TA}, f_3, \cdots, f_{t'^{(1)}} \in G_1$ denote the group elements of $\mathcal{A}$'s computation. The generic $\mathcal{A}$ computes $f_j = P^{a_{j,-1}} P_{TA}^{a_{j,0}} \prod_{\ell=1}^{q_B} X_\ell{}^{a_{j,\ell}}$ where X_ℓ are $\mathcal{BSO}$ commitments and the exponents depend arbitrarily on previously computed non-group data.

Schnorr's Lemma 2 implies DLP is hard (uncomputable by PPT generic adversary) in GGM. Similarly, it applies here. It is hard to get s from $Q_B{}^s$.

Let $\mathcal{A}$ outputs $(\hat{X}_i, \hat{W}_i, \hat{V}_i)$ be valid for message $\hat{m}_i$, sender ID_A and recipient ID_{B_i}, $1 \leq i \leq q_B + 1$. Then we have $\hat{h_i} = H_1(\hat{X}_i, \hat{m}_i, ID_{B_i})$ for some hash query satisfying $e(\hat{X}_i P_{TA}^{\hat{h_i}}, Q_A) = e(P, \hat{W}_i)$. Let $\hat{X}_i = f_{\sigma_i}^{(1)}$.

The equation $e(P, \hat{W}_i)e(P_{TA}^{-\hat{h_i}}, Q_A){=}e(f_{\sigma_i}, Q_A){=}e(P^{a_{\sigma_i,-1}} P_{TA}^{a_{\sigma_i,0}} \prod_{\ell=1}^{q_B} X_\ell{}^{a_{\sigma_i,\ell}}, Q_A)$ and $e(X_\ell, Q_A){=}e(P, W_\ell)e(P_{TA}^{-h_\ell}, Q_A)$ imply:

$$\hat{W}_i = Q_A{}^{a_{\sigma_i,-1}} \cdot \prod_{\ell=1}^{q_B} W_\ell^{a_{\sigma_i,\ell}} \cdot Q_A{}^{(a_{\sigma_i,0} - \sum_{\ell=1}^{q_B} a_{\sigma_i,\ell} h_\ell + \hat{h_i})s}$$

If $\hat{h_i} = -a_{\sigma_i,0} + \sum_{\ell=1}^{l} a_{\sigma_i,\ell} h_\ell$, then $\mathcal{A}$ can easily compute the correct $\hat{W}_i$. Then we have $\hat{W}_i = Q_A{}^{a_{\sigma_i,-1}} \prod_{\ell=1}^{l} W_\ell^{a_{\sigma_i,\ell}}$ where $W_1,\cdots,W_l$, $a_{\sigma_i,-1},\cdots,a_{\sigma_i,l}$ are known to $\mathcal{A}$.

Conversely, $\mathcal{A}$ must select $h_1, \cdots, h_l$ as to zero the coefficient involving the master secret key s. Otherwise we can recover $Q_A{}^s$ from $W_1,\cdots,W_l$, $a_{\sigma_i,-1},\cdots,a_{\sigma_i,l}$, $\hat{h_i}$, $\hat{W}_i$ which are known to $\mathcal{A}$. Then it can solve the 1m-co-CDH problem, as we get q_K private keys from $\mathcal{KEO}$. The probability of solving 1m-co-CDH in GGPM is negligible. Hence $\mathcal{A}$ must solve the ROS problem. □

B Detailed comparison on performance of IBSC

We compare our IBSC scheme with existing schemes from Malone-Lee(M) [15], Libert and Quisquater scheme 1(LQ1) [13] and 2(LQ2)[14], Nalla and Reddy(NR) [11], Boyen(B) [5], and Chow et al.(CYSC) [9]. We also include the Sign-then-Encrypt(StE) and Encrypt-then-Sign(EtS) using ID-based encryption from [4] and ID-based signature from [6].

B.1 Security

The security analysis follows our definition of security models in Section 2: IND-A, IND-B, IND-C, EU.

- IND-A: The schemes of M, LQ1, NR and CYSC are not IND-A secure. It is because the unsigncryption of ciphertext requires the knowledge of sender's identity in advance.
- IND-B: The schemes of LQ1 and NR are not IND-B secure. Any adversary who knows the sender's identity, private key and the message signcrypted can distinguish the identity of the recipient.

- IND-C: The scheme of M is not IND-CCA2 secure shown in [13]. Schemes of LQ1 and NR are IND secure according to security model in LQ1, but not secure in Boyen's and our security models, where private key of sender is known to Adversary.
- EU: NR's scheme is not EU-CMA secure. Any adversary can forge a signcryption from any sender to recipient ID_B, where private key of ID_B is known to adversary. Boyen's scheme has unforgeability for the signature only. It does not satisfy the unforgeability for 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. LQ2 scheme is similar to Boyen's in this aspect. Our IBSC scheme avoids this controversial property of unlinkability and achieves unforgeability for ciphertext.

Some comments based on the above definitions are given in Appendix C.

B.2 Computation Time

The computation of 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 NR [11], 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 before knowing the recipient identity and message. For any random number r, X, ${Q_A}^r$ and ${P_{TA}}^r$ can be pre-computed. Therefore the actual number of exponential in our scheme which cannot be pre-computed is two, which is shown in bracket in the table. We can see that our scheme is again the fastest in terms of exponential computation.

B.3 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 [14], δ is 160 bits for ciphertext unlinkability, and is 0 bit for ciphertext linkability. As shown in the table, we can see that our scheme has the shortest ciphertext size.

C Comments on various IBSC's w.r.t. our security model

C.1 Comment for IND-B

In the following, please refer to the original paper for original scheme and the definition of the symbols used. In the IND sub-game (b), the Adversary chooses message m, sender ID_A and recipient ID_{B1}. The Adversary knows the private key of ID_A. Simulator chooses a recipient ID_{B0}, and randomly picks $b \in \{0, 1\}$. Simulator signcrypts the message m from sender ID_A to recipient ID_{Bb} and returns the ciphertext to the Adversary. The Adversary has to guess b.

Libert and Quisquater's scheme 1 [13] The Adversary has the ciphertext $\langle c, r, S\rangle$ and d_A, the private key of ID_A. The Adversary computes:

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

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

Nalla and Reddy's scheme [11] The Adversary has the ciphertext $\langle R, S, C\rangle$ and S_A, the private key of ID_A. The Adversary computes:

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

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.2 Comment for IND-C

In the IND sub-game (c), the Adversary chooses message m_1, sender ID_A and recipient ID_B. The Adversary knows the private key of ID_A. Simulator chooses a message m_0, and randomly picks $b \in \{0, 1\}$. Simulator signcrypts 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.

Nalla and Reddy's scheme [11] The Adversary has the ciphertext $\langle R, S, C\rangle$ and S_A, the private key of ID_A. The Adversary computes:

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

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.3 Comment for EU

In the EU game, the Adversary chooses message m, sender ID_A and recipient 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.

Nalla and Reddy's scheme [11] The Adversary has S_B, the private key of ID_B. The Adversary randomly chooses $a \in R$ and computes:

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

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:

$$\begin{aligned}
k_B &= H''(e(S, S_B)) \\
m &= k_B \oplus C
\end{aligned}$$

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.