Cryptanalysis of ID-Based Signcryption Scheme for Multiple Receivers

No author given

No Institute Given

Abstract. In ATC 2007, an identity-based signcryption scheme for multiple receivers was proposed by Yu et al. They prove confidentiality of their scheme and also claim unforgeability without any proof. In this paper, we show that their signcryption scheme is insecure by demonstrating a universal forgeability attack — anyone can generate a valid signcrypted ciphertext on any message on behalf of any legal user for any set of legal receivers without knowing the secret keys of the legal users. Further, we propose a corrected version of their scheme and formally prove its security (confidentiality and unforgeability) under the existing security model for signcryption. We also analyze the efficiency of the corrected scheme by comparing it with existing signcryption schemes for multiple receivers.

Keywords. Signcryption, Cryptanalysis, Multiple Receivers, Bilinear Pairing.

1 Introduction

Encryption and signatures are basic cryptographic tools offered by public key cryptography for achieving privacy and authenticity. Both primitives are used in a number of high level protocols. There are scenarios where properties of both primitives are needed. The most common example is secure emailing, where the messages should be encrypted and signed to provide confidentiality and authentication. For achieving this, encryption schemes and signature schemes can be combined to meet the requirements. This was shown to be complex by An et al. in [17]. Signcryption, introduced by Zheng in 1997 [2], is a cryptographic primitive that offers confidentiality and unforgeability simultaneously similar to the sign-then-encrypt technique, but with lesser computational complexity and lower communication cost. This has made signcryption a suitable primitive for applications that require secure and authenticated message delivery, where devices have limited resources. After Zheng's work, a number of signcryption schemes were proposed ([4], [19], [20], [21], [11], [10], [22]). The security notion for signcryption was first formally defined in 2002 by Baek et al. in [18]. This was similar to the notion of semantic security against adaptive chosen ciphertext attack and existential unforgeability against adaptive chosen message attack.

The concept of an Identity based (ID-based) cryptosystem was introduced by Shamir [24] in 1984. The distinguishing characteristic of ID-based cryptography is the ability to use any string as a public key. In particular, this string maybe the email address, telephone number, or any publicly available parameter of an individual that is unique to that individual.The corresponding private key can only be derived by a trusted Private Key Generator (PKG) who keeps a master secret which is involved in the said derivation. An ID-based cryptosystem removes the need for senders to look up the receiver's public key before sending out an encrypted message. It provides a more convenient alternative to conventional public key infrastructure.

ID-based signcryption schemes achieve the functionality of signcryption with the added advantage that ID-based cryptography provides. In [8], Malone-Lee gave the first ID-based signcryption scheme. Later it was found that Malone-Lee's scheme was not semantically secure. Since then, quite a few ID-based signcryption schemes have been proposed ([9], [22], [10], [23], [12]). To date, some of the most efficient ID-based signcryption schemes are that of Chen et al. [12], and Barreto et al. [23]

In practice, broadcasting a message to multiple users in a secure and authenticated manner is an important facility for a group of people who are jointly working on the same project to communicate with one another. While this can be achieved by using the single-user signcryption primitive individually for each recipient, this method results in huge computation and communication costs. Instead, we opt for multi-receiver signcryption, whose objective is to efficiently broadcast a single confidential ciphertext to different receivers by performing a single signcryption operation, while achieving the security properties of authenticity and unforgeability. We point out that there are only two multi-receiver ID-based signcryption schemes till date, which we discuss briefly. Duan et al. [15] were the first to come up with an ID-based scheme for multi-receiver signcryption. Their scheme requires just one pairing operation to signcrypt a single message for multiple receivers. They prove, in the random oracle model, that their scheme achieves confidentiality against chosen ciphertext attacks and strong existential unforgeability against chosen message attacks. Following this, You et al. [1] came up with another scheme with improved efficiency in the designcryption phase (their scheme requires one less pairing operation than Dual et al.'s).

Our Contribution. We show that the signcryption scheme of You et al. [1] is insecure with respect to unforgeability by demonstrating an attack which shows that any legal user of the system can generate a signcrypted ciphertext on any message on behalf of any other legal user for any set of receivers without knowing the secret key of any other legal users. Further, we propose a corrected version of their scheme and prove its security (confidentiality and unforgeability) under the existing security model for signcryption. We also analyze the efficiency of the corrected scheme by comparing it with an existing identity-based signcryption scheme for multiple receivers.

The rest of this paper proceeds as follows. In Section 2, we review the preliminaries like bilinear pairings and related computational problems, the general framework of ID-based signcryption schemes for multiple receivers, and the security models for such schemes. Next, in Section 3, we review You et al.'s proposed multi-receiver ID-based signcryption scheme. We present the attack on this scheme in Section 4. In Section 5, we lay out the details of our fix to You et al.'s scheme. In Section 6, we present the proof of unforgeability of our scheme, while we move the proofs of correctness and confidentiality of our scheme to the appendix. Section 7 discusses the efficiency of our scheme in comparison with the existing scheme of Duan et al. and Section 8 concludes the discussion.

2 Preliminaries

2.1 Bilinear Pairing

Let $\mathbb{G}_1$ be an additive cyclic group generated by P, with prime order q, and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order q. A bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties.

- ***Bilinearity.*** For all $P, Q, R \in \mathbb{G}_1$,
  - $\hat{e}(P+Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$
  - $\hat{e}(P, Q+R) = \hat{e}(P, Q)\hat{e}(P, R)$
  - $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- ***Non-Degeneracy.*** There exist $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq I_{\mathbb{G}_2}$, where $I_{\mathbb{G}_2}$ is the identity element of $\mathbb{G}_2$.
- ***Computability.*** There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

2.2 Computational Assumptions

In this section, we review the computational assumptions related to bilinear maps that are relevant to the protocol we discuss.

Bilinear Diffie-Hellman Problem (BDHP) Given $(P, aP, bP, cP) \in \mathbb{G}_1^4$ for unknown $a, b, c \in \mathbb{Z}_q^*$, the BDH problem in $\mathbb{G}_1$ is to compute $\hat{e}(P, P)^{abc}$.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the BDH problem in $\mathbb{G}_1$ is defined as

$$Adv_{\mathcal{A}}^{BDH} = Pr\left[\mathcal{A}(P, aP, bP, cP) = \hat{e}(P, P)^{abc} \mid a, b, c \in \mathbb{Z}_q^*\right]$$

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

Decisional Bilinear Diffie-Hellman Problem (DBDHP) Given $(P, aP, bP, cP, \alpha) \in \mathbb{G}_1^4 \times \mathbb{G}_2$ for unknown $a, b, c \in \mathbb{Z}_q^*$, the DBDH problem in $\mathbb{G}_1$ is to decide if $\alpha = \hat{e}(P, P)^{abc}$.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the DBDH problem in $\mathbb{G}_1$ is defined as

$$Adv_{\mathcal{A}}^{DBDH} = |Pr\left[\mathcal{A}(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1\right] - Pr\left[\mathcal{A}(P, aP, bP, cP, \alpha) = 1\right]|$$

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

Computation Diffie-Hellman Problem (CDHP) Given $(P, aP, bP) \in \mathbb{G}_1^3$ for unknown $a, b \in \mathbb{Z}_q^*$, the CDH problem in $\mathbb{G}_1$ is to compute abP.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{G}_1$ is defined as

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

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

2.3 Identity-Based Signcryption for Multiple Receivers

A generic IBSC multi-receiver scheme for sending a single message to t users consists of the following probabilistic polynomial time algorithms,

- ***Setup***(k)**.** Given a security parameter k, the Private Key Generator (PKG) generates the public parameters *params* and master secret key *msk* of the system.
- ***Keygen***(ID_{Alice})**.** Given an identity ID_{Alice}, the PKG computes the corresponding private key D_{Alice} and transmits it to *Alice* in a secure way.
- ***Signcrypt***$(m, ID_{Alice}, \mathcal{L} = \{ID_1, ID_2, \ldots, ID_t\}, D_{Alice})$**.** To send a message m to $(ID_1, ID_2, \ldots, ID_t)$, *Alice* with identity ID_{Alice} and private key D_{Alice} runs this algorithm to obtain the signcrypted ciphertext σ.
- ***Designcrypt***$(\sigma, ID_{Alice}, ID_{Bob}, D_{Bob})$**.** When *Bob* with identity ID_{Bob} and private key D_{Bob} receives the signcrypted ciphertext σ from *Alice* with identity ID_{Alice}, he runs this algorithm to obtain either the plain text m or $\perp$ according as whether σ was a valid signcryption from identity ID_{Alice} to identity ID_{Bob} or not.

For consistency, we require that if $\sigma = \mathit{Signcrypt}\,(m, ID_{Alice}, (ID_1, ID_2, \ldots, ID_t), D_{Alice})$, then $m = \mathit{Designcrypt}\,(\sigma, ID_{Alice}, ID_i, D_i)$ for $1 \leq i \leq t$.

2.4 Security Model for Identity-Based Signcryption for Multiple Receivers (MIBSC)

The notion of semantic security of public key encryption was extended to identity-based signcryption scheme by Malone-Lee in [8]. This was later modified by Sherman et al. in [10] which incorporates indistinguishability against adaptive chosen ciphertext and identity attacks (IND-IBSC-CCIA) and existential unforgeability against adaptive chosen message and identity attacks (EUF-IBSC-ACMIA). We describe below the security models for *confidentiality* and *unforgeability* given in [11], this is the strongest security notion for this problem.

Confidentiality A signcryption scheme is semantically secure against chosen ciphertext attack (IND-MIBSC-CCA2) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the *Setup* algorithm and sends the system public parameters to the adversary $\mathcal{A}$.
2. In the first phase, $\mathcal{A}$ makes polynomially bounded number of queries to the following oracles.
   (a) **Keygen Oracle** — $\mathcal{A}$ produces an identity ID_i and queries for the secret key of user i. The *Keygen Oracle* returns D_i to $\mathcal{A}$.
   (b) **Signcrypt Oracle** — $\mathcal{A}$ produces a message m, sender identity ID_A and a list of receiver identities $ID_1, ID_2, \ldots, ID_t$. $\mathcal{C}$ computes the secret key D_A from $Keygen(ID_A)$ and returns to $\mathcal{A}$, the signcrypted ciphertext σ from $\mathit{Signcrypt}\,(m, ID_A, \{ID_1, ID_2, \ldots, ID_t\}, D_A)$.
   (c) **Designcrypt Oracle** — $\mathcal{A}$ produces a sender identity ID_A, receiver identity ID_B and a signcryption σ. The challenger $\mathcal{C}$ computes the secret key D_B from $Keygen(ID_B)$, returning the result of $\mathit{Designcrypt}\,(\sigma, ID_A, ID_B, D_B)$ to $\mathcal{A}$. The result returned is $\perp$ if σ is an invalid signcryption from ID_A to ID_B.

3. $\mathcal{A}$ produces two messages m_0 and m_1 of equal length from the message space $\mathcal{M}$ and an arbitrary sender identity ID_A. The challenger $\mathcal{C}$ flips a coin, sampling a bit $b \leftarrow \{0,1\}$ and computes $\sigma^* = \mathit{Signcrypt}\left(m_b, ID_A, \{ID_1, ID_2, \ldots, ID_t\}, D_A\right)$. σ^* is returned to $\mathcal{A}$ as challenge signcrypted ciphertext.
4. $\mathcal{A}$ is allowed to make polynomially bounded number of new queries as in Step 2 with the restrictions that it should not query the *Designcryption Oracle* for the designcryption of σ^*, the *Signcryption Oracle* for the signcryption of m_0 or m_1 under the sender identity ID_A and the *Keygen Oracle* for the secret keys of $ID_1, ID_2, \ldots, ID_t$.
5. At the end of this game, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins the game if $b' = b$.

Unforgeability A signcryption scheme is existentially unforgeable under chosen message attack (EUF-MIBSC) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the *Setup* algorithm to generate the master public and private keys *params* and *msk* respectively. $\mathcal{C}$ gives system public parameters *params* to $\mathcal{A}$ and keeps the master private key *msk* secret from $\mathcal{A}$.
2. The adversary $\mathcal{A}$ makes polynomially bounded number of queries to the oracles as described in Step 2 of the confidentiality game.
3. $\mathcal{A}$ produces a signcrypted ciphertext σ and wins the game if the private key of sender identity ID_A was not queried in the previous step and $\perp$ is not returned by $\mathit{Designcrypt}(\sigma, ID_A, ID_B, D_B)$ and σ is not the output of a previous query to the *Signcrypt Oracle* with ID_A as sender.

3 Review of You's ID-Based Multi-Receiver Signcryption Scheme (Y-MIBSC)

The Y-MIBSC scheme in [1] has the following algorithms.

3.1 Setup(k)

The security parameter of the scheme is k and $\mathbb{G}_1$, $\mathbb{G}_2$ are two groups of prime order q and P is a generator of $\mathbb{G}_1$ and $\hat{e}$ is a bilinear map defined as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Let n_0, n_1, n_2 and n_3 denote the number of bits required to represent an identity, an element of $\mathbb{G}_1$, an element of $\mathbb{G}_2$ and a message respectively. Three hash functions $H_1 : \{0,1\}^{n_0} \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^{n_1+n_3} \rightarrow \mathbb{Z}_q^*$, $H_3 : \{0,1\}^{n_2} \rightarrow \{0,1\}^{n_3}$ are used. The PKG chooses $s \in \mathbb{Z}_q^*$ and $R \in \mathbb{G}_1 \backslash \{\mathbf{0}_{\mathbb{G}_1}\}$ and computes $P_{pub} = sP$ and $\theta = \hat{e}(R, P_{pub})$, where $\mathbf{0}_{\mathbb{G}_1}$ denotes the zero element of $\mathbb{G}_1$. The public parameters are $\langle \mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, R, \theta, \hat{e}, H_1, H_2, H_3 \rangle$.

3.2 Keygen(ID_A)

The public key and private key of user A are computed from his identity ID_A as $Q_A = H_1\left(ID_A\right)$ and $D_A = sQ_A$ respectively.

3.3 Signcrypt$(m, ID_A, ID_1, ID_2, \ldots, ID_n, D_A)$

Suppose A wants to encrypt a message m to n receivers with identities $ID_1, ID_2, \ldots, ID_n$. User A does the following.

1. Choose $r \in_R \mathbb{Z}_q^*$
2. Compute the following.
   (a) $X = rQ_A$
   (b) $h_2 = H_2(X \| m)$
   (c) $Z = (r + h_2) D_A$
   (d) $U = rP$
   (e) $\omega = \hat{e}(Z, P)$
   (f) $y = m \oplus H_3(\omega)$
   (g) $W = \theta^r \omega$
   (h) $T_i = rH_1(ID_i) + rR$, for $1 \leq i \leq n$.
3. The signcrypted ciphertext is $\sigma = \langle y, U, X, W, T_1, T_2, \ldots, T_n, L\rangle$, where L is the list of receivers who can decrypt the message. Here, T_i is meant for the receiver ID_i.

3.4 Designcrypt$(\sigma, ID_A, ID_i, D_i)$

A receiver with identity ID_i uses his secret key D_i to designcrypt $\sigma = \langle y, U, X, W, T_i, L\rangle$ from ID_A as follows.

1. Compute the following.
   (a) $\omega' = W\hat{e}(U, D_i)\,\hat{e}(P_{pub}, T_i)^{-1}$
   (b) $m' = y \oplus H_3(\omega')$
   (c) $Q_A = H_1(ID_A)$
   (d) $h_2' = H_2(X \| m')$
2. If $\omega' = \hat{e}(P_{pub}, X + h_2' Q_A)$, return m'. Otherwise, return $\perp$.

4 Attack on Y-MIBSC

The scheme described above [1] is insecure from the point of view of unforgeability. Anybody can generate a valid signcryption for any message m^* as if it were generated by another legal user. We describe how the attack proceeds in this section.

Let *Alice* be a legal user of the system and *Eve* be any forger. If *Eve* wants to generate a signcryption on any message m^* as if it were generated by *Alice* for a list of legal users of the system with identities $ID_1, ID_2, \ldots, ID_t$, *Eve* just has to do the following.

1. Choose $r^* \in_R \mathbb{Z}_q^*$
2. Compute the following.
   (a) $X^* = r^* Q_{Alice}$
   (b) $h_2^* = H_2(X^* \| m^*)$
   (c) $Z^* = (r^* + h_2^*)\, Q_{Alice}$.
   (d) $U^* = r^* P$
   (e) $\omega^* = \hat{e}(Z^*, P_{pub})$
   (f) $y^* = m^* \oplus H_3(\omega^*)$

(g) $W^* = \theta^{r^*}\omega^*$

(h) $T_j^* = r^* H_1(ID_j) + r^* R$, for $1 \leq j \leq t$

3. $\sigma^* = \langle y^*, U^*, X^*, W^*, T_1^*, T_2^*, \ldots, T_t^*, L^* \rangle$ is the signature of *Alice* on message m^* generated by *Eve* for the list of users L^* with identities $\{ID_j\}_{1\leq j\leq t}$

We now prove that the σ^* generated by *Eve* is a valid signcryption from *Alice* to the receivers in L^* on the message m^*.

Designcrypt$(\sigma^* = \langle y^*, U^*, X^*, W^*, T_1^*, T_2^*, \ldots, T_t^*, L^* \rangle, ID_{Alice}, ID_j, D_j)$**.** A receiver with identity ID_j uses his secret key D_j to designcrypt σ^* obtained from *Eve* as follows.

1. Compute the following.

(a) $Q_{Alice} = H_1(ID_{Alice})$

(b) Next, it can be seen that

$$\begin{aligned}\omega' &= W^* \hat{e}(U^*, D_j)\, \hat{e}\left(P_{pub}, T_j^*\right) \\ &= \theta^{r^*}\omega^* \hat{e}(r^*P, sQ_j)\, \hat{e}(P_{pub}, r^*Q_j + r^*R)^{-1} \\ &= \hat{e}(P_{pub}, R)^{r^*}\, \omega^* \hat{e}(P, Q_j)^{r^*s}\, \hat{e}(P, Q_j)^{-r^*s}\, \hat{e}(P, R)^{-r^*s} \\ &= \omega^*\end{aligned}$$

(c) $m' = y^* \oplus H_3(\omega') = m^*$

(d) $h_2' = H_2(X^* \| m') = h_2^*$

2. Next, the check $\omega' \overset{?}{=} \hat{e}(P_{pub}, X^* + h_2' Q_{Alice})$ is performed. We show below that this test will succeed and hence message m^* will be returned.

$$\begin{aligned}\hat{e}(P_{pub}, X^* + h_2' Q_{Alice}) &= \hat{e}(sP, r^*Q_{Alice} + h_2^* Q_{Alice}) \quad (\text{since } h_2' = h_2^*) \\ &= \hat{e}(sP, (r^* + h_2^*)\, Q_{Alice}) \\ &= \hat{e}(P_{pub}, Z^*) \quad (\text{from Step 2(c) of } \textit{Eve}\text{'s forgery above}) \\ &= \hat{e}(Z^*, P_{pub}) \quad (\text{by symmetry of the bilinear map}) \\ &= \omega^* = \omega'\end{aligned}$$

From this it is clear that *Eve* can succeed in generating a signcryption of message m^* with Alice as sender and identities ID_j, $1 \leq j \leq t$ as receivers without knowing the secret key of *Alice*. Thus any legal user can forge any message on behalf of any other legal user to any set of receivers.

5 Improved Multi-Receiver Identity Based Signcryption Scheme (I-MIBSC)

In this section, we propose an improved version of Y-MIBSC, which we formally prove to be secure. The setup and key generation algorithms of I-MIBSC are similar to that of Y-MIBSC, but with slightly different hash functions. The details are given below.

5.1 Setup($\boldsymbol{k}$)

Let k be the security parameter of the system. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q and let P be the generator of $\mathbb{G}_1$ and $\hat{e}$ be a bilinear map defined as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. As before, let n_0, n_1, n_2 and n_3 denote the number of bits required to represent an identity, an element of $\mathbb{G}_1$, an element of $\mathbb{G}_2$ and a message respectively. Consider three hash functions $H_1 : \{0,1\}^{n_0} \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^{n_0+2n_1+n_3} \rightarrow \mathbb{Z}_q^*$, $H_3 : \{0,1\}^{n_2} \rightarrow \{0,1\}^{n_1+n_3}$. The PKG chooses its secret key $s \in \mathbb{Z}_q^*$ and sets the public key $P_{pub} = sP$. The PKI also chooses $R \in \mathbb{G}_1 \setminus \{\mathbf{0}_{\mathbb{G}_1}\}$ and computes $\theta = e(R, sP)$, where $\mathbf{0}_{\mathbb{G}_1}$ denotes the zero element of $\mathbb{G}_1$. The public parameters of the system are $\langle \mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, R, \theta, \hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2, H_1, H_2, H_3 \rangle$.

5.2 Keygen($\boldsymbol{ID_A}$)

The public key and private key of user A are computed from his identity ID_A as $Q_A = H_1(ID_A)$ and $D_A = sQ_A$ respectively.

5.3 Signcrypt($\boldsymbol{m, ID_A, ID_1, ID_2, \ldots, ID_n, D_A}$)

For signcryption of message m by user A with identity ID_A and secret key D_A to n receivers with identities $ID_1, ID_2, \ldots, ID_n$, do the following.

1. Choose $r_1, r_2 \in_R \mathbb{Z}_q^*$
2. Compute the following.
   (a) $U = r_1 P$
   (b) $X = r_2 Q_A$
   (c) $h_2 = H_2(ID_A \| U \| X \| m)$
   (d) $Z = (r_2 + h_2) D_A$
   (e) $\omega = \hat{e}(Z, P)$
   (f) $y = (m \| Z) \oplus H_3(\omega)$
   (g) $W = \theta^{r_1} \omega$
   (h) $T_i = r_1(Q_i + R)$, for $1 \leq i \leq n$
3. The signcrypted ciphertext is $\sigma = \langle y, U, X, W, T_1, T_2, \ldots, T_n, L \rangle$, where L is the list of receivers who can decrypt the message. Here, T_i is meant for the receiver ID_i.

5.4 Designcrypt($\boldsymbol{\sigma, ID_A, ID_i, D_i}$)

A receiver with identity ID_i uses his secret key D_i to designcrypt $\sigma = \langle y, U, X, W, T_i, L \rangle$ from ID_A as follows.

1. Compute the following.
   (a) $\omega' = W \hat{e}(U, D_i) \hat{e}(P_{pub}, T_i)^{-1}$
   (b) $m' \| Z' = y \oplus H_3(\omega')$
   (c) $h_2' = H_2(ID_A \| U \| X \| m')$
2. If $\omega' = \hat{e}(Z', P)$ and $\omega' = \hat{e}(X + h_2' Q_A, P_{pub})$, return m'. Otherwise, return $\perp$.

We prove the correctness of our scheme in Appendix A and confidentiality of our scheme in Appendix B. Since the attack we presented on You et al.'s scheme was based on the unforgeability aspect, we present the proof of unforgeability of our scheme formally in Section 6.

6 Proof of Unforgeability of I-MIBSC

Theorem. *Our multi-receiver identity based signcryption scheme I-MIBSC is secure against any EUF-MIBSC adversary $\mathcal{A}$ under the random oracle model if CDHP is hard in $\mathbb{G}_1$.*

The challenger $\mathcal{C}$ receives an instance (P, aP, bP) of the CDH problem. His goal is to determine abP. Suppose there exists an EUF-MIBSC adversary $\mathcal{A}$ for our proposed I-MIBSC scheme. We show that $\mathcal{C}$ can use $\mathcal{A}$ to solve the CDH problem. $\mathcal{C}$ will set the random oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{KeyExtract}$, $\mathcal{O}_{Signcrypt}$ and $\mathcal{O}_{Designcrypt}$. The answers to the oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, and $\mathcal{O}_{H_3}$ are randomly selected, therefore, to maintain consistency, $\mathcal{C}$ will maintain three lists $L_1 = \langle ID_i, Q_i, x_i \rangle$, $L_2 = \langle ID_i, U, X, m, h_2 \rangle$, $L_3 = \langle \omega, h_3 \rangle$. We assume that $\mathcal{A}$ will ask for $H_1(ID)$ before ID is used in any key extraction, signcryption and designcryption queries. First, the adversary $\mathcal{A}$ outputs the identity ID_A of the sender whose signcryption he claims to be able to forge. Then, the challenger $\mathcal{C}$ gives $\mathcal{A}$ the system parameters *params*, consisting of P, $P_{pub} = bP$, R, $\theta = \hat{e}(R, P_{pub} = \hat{e}(R, bP)$. The descriptions of the oracles follow.

Oracle $\mathcal{O}_{\mathbf{H_1}}(\mathbf{ID_i})$***.*** $\mathcal{C}$ checks if there exists a tuple (ID_i, Q_i, x_i) in L_1. If such a tuple exists, $\mathcal{C}$ answers with Q_i. Otherwise, $\mathcal{C}$ does the following.

1. If $ID_i \neq ID_A$, choose a new[1] $x_i \in_R \mathbb{Z}_q^*$ and set $Q_i = x_i P$.
2. If $ID_i = ID_A$, choose a new $x_i \in_R \mathbb{Z}_q^*$ and set $Q_i = (x_i - a)P$.
3. Add the tuple (ID_i, Q_i, x_i) to L_1 and return Q_i.

Oracle $\mathcal{O}_{\mathbf{H_2}}(\mathbf{ID_i} \| \mathbf{U} \| \mathbf{X} \| \mathbf{m})$***.*** $\mathcal{C}$ checks if there exists a tuple (ID_i, U, X, m, h_2) in L_2. If such a tuple exists, $\mathcal{C}$ returns h_2. Otherwise, $\mathcal{C}$ chooses a new $h_2 \in_R \mathbb{Z}_q^*$, adds the tuple (ID_i, U, X, m, h_2) to L_2 and returns h_2.

Oracle $\mathcal{O}_{\mathbf{H_3}}(\omega)$***.*** $\mathcal{C}$ checks if there exists a tuple (ω, h_3) in L_3. If such a tuple exists, $\mathcal{C}$ returns h_3. Otherwise, $\mathcal{C}$ chooses a new $h_3 \in_R \{0,1\}^{n_1+n_3}$, adds the tuple (ω, h_3) in L_3 and returns h_3.

Oracle $\mathcal{O}_{\mathbf{KeyExtract}}(\mathbf{ID_i})$***.*** $\mathcal{C}$ does the following.

1. If $ID_i = ID_A$, return $\perp$.
2. If $ID_i \neq ID_A$, recover the tuple (ID_i, Q_i, x_i) from L_1 and return $D_i = x_i P_{pub} = bQ_i$.

Oracle $\mathcal{O}_{\mathbf{Signcrypt}}(\mathbf{m}, \mathbf{ID_i}, \mathcal{L})$***.*** On receiving this query, where $\mathcal{L} = \{ID_1, ID_2, \ldots, ID_t\}$ is the list of intended receivers, $\mathcal{C}$ checks if $ID_i = ID_A$. If not, $\mathcal{C}$ computes D_i using $\mathcal{O}_{KeyExtract}(ID_i)$, generates the signcryption in a normal way and returns it. Otherwise, that is, if $ID_i = ID_A$, it chooses r, r' and a new $h_2 \in_R \mathbb{Z}_q^*$ and does the following.

1. Compute $U = r'P$
2. Compute $X = rP - h_2\mathcal{O}_{H_1}(ID_A)$ and add the tuple (ID_A, U, X, m, h_2) to L_2.
3. Compute the following.
   (a) $Z = rP_{pub}$
   (b) $\omega = \hat{e}(Z, P)$
   (c) $y = \mathcal{O}_{H_3}(\omega) \oplus (m \| Z)$

[1] By new, we mean that the random value chosen must not have been already chosen during an earlier execution.

(d) For all $ID_j \in \mathcal{L}, T_j = r'(\mathcal{O}_{H_1}(ID_j) + R)$.
(e) $W = \theta^{r'}\omega$

4. Return the signcrypted ciphertext $\sigma = \langle y, U, X, W, T_1, T_2, \ldots, T_t, \mathcal{L}\rangle$.

Oracle $\mathcal{O}_{\textbf{Designcrypt}}(\sigma, \mathbf{ID_i}, \mathbf{ID_j})$**.** On receiving this query, where the signcryption $\sigma = \langle y, U, X, W, T_1, T_2, \ldots, T_t, \mathcal{L}\rangle$, $\mathcal{C}$ checks if $ID_j = ID_A$. If not, then $\mathcal{C}$ computes D_j using $\mathcal{O}_{KeyExtract}(ID_j)$, designcrypts σ in the normal way and returns what the designcryption algorithm returns. Otherwise, that is, if $ID_j = ID_A$, then $\mathcal{C}$ tries to locate entries $(ID_i, U, X, m, h_2) \in L_2$ and $(\omega, h_3) \in L_3$ for some h_2, h_3, and ω under the constraints that $\omega = \hat{e}(P_{pub}, X + h_2\mathcal{O}_{H_1}(ID_i))$, $(m\|Z) = h_3 \oplus y$, and $\omega = \hat{e}(Z, P)$. If no such entries are found, the oracle returns $\perp$. Otherwise, m is returned.

Eventually $\mathcal{A}$ outputs a forged signcryption $\sigma' = \langle y', U', X', W', T'_1, T'_2, \ldots, T'_t, \mathcal{L}'\rangle$ on some message m' from the sender ID_A to users in the set $\mathcal{L}' = \{ID_1, ID_2, \ldots, ID_t\}$, with $ID_A \notin \mathcal{L}'$. Challenger $\mathcal{C}$ designcrypts the ciphertext σ' with any of the identities $ID_j \in \mathcal{L}'$ to get the 'signature' Z' of ID_A on m', if σ' is a valid signcrypted ciphertext from ID_A to ID_j on message m'. Now, $\mathcal{C}$ applies the oracle replay technique to produce two valid signcrypted ciphertexts $\sigma_1 = \langle y_1, U_1, X_1, W_1, T'^1_1, T'^1_2, \ldots, T'^1_t, \mathcal{L}'\rangle$ and $\sigma_2 = \langle y_2, U_2, X_2, W_2, T'^2_1, T'^2_2, \ldots, T'^2_t, \mathcal{L}'\rangle$ on some message m' from the sender ID_A to users in the set $\mathcal{L}' = \{ID_1, ID_2, \ldots, ID_t\}$, with $ID_A \notin \mathcal{L}'$. $\mathcal{C}$ designcrypts σ_1 and σ_2 to obtain signatures $Z_1 = (r_2 + h'_2)D_A$ and $Z_2 = (r_2 + h''_2)D_A$. Now we can apply standard arguments for the outputs of the forking lemma since both Z_1 and Z_2 are valid signatures for the same message m' and same random tape of the adversary. Finally, $\mathcal{C}$ obtains the solution to the CDH instance as $x_A P_{pub} - (h'_2 - h''_2)^{-1}(Z_1 - Z_2)$. In fact,

$$\begin{aligned} x_A P_{pub} - (h'_2 - h''_2)^{-1}(Z_1 - Z_2) &= x_A P_{pub} - (h'_2 - h''_2)^{-1}(h'_2 - h''_2)D_A \\ &= x_A P_{pub} - D_A = x_A bP - D_A \\ &= x_A bp - (x_A - a)bP = abP \end{aligned}$$

So, we can see that the challenger $\mathcal{C}$ has the same advantage in solving the CDH problem as the adversary $\mathcal{A}$ has in forging a valid signcrypted ciphertext. So, if there exists an adversary who can forge a valid signcrypted ciphertext with non-negligible advantage, that means there exists an algorithm to solve the CDH problem with non-negligible advantage. Since this is not possible, no adversary can forge a valid signcrypted ciphertext with non-negligible advantage. Hence, I-MIBSC is secure against any EUF-MIBSC attack. □

7 Efficiency Analysis

In this section, we compare the efficiency of our scheme with Duan et al.'s scheme [15]. For this, we consider the costly operations which include map-to-point hash operation ($\mathbb{G}_1$ Map), point scalar multiplication on $\mathbb{G}_1$ ($\mathbb{G}_1$ Mul), exponentiation on $\mathbb{G}_2$ ($\mathbb{G}_2$ Exp) and pairing operation (Pairing).

Scheme	Signcrypt			
	$\mathbb{G}_1$ Map	$\mathbb{G}_1$ Mul	Pairing	$\mathbb{G}_2$ Exp
Our scheme	-	n+3	1	1
Duan et al.	1	n+4	1	-

Table 1. Comparison of Efficiency of I-MIBSC Scheme with Duan et al.'s Scheme (Signcryption)

Scheme	Designcrypt			
	$\mathbb{G}_1$ Map	$\mathbb{G}_1$ Mul	Pairing	$\mathbb{G}_2$ Exp
Our scheme	-	1	4	1
Duan et al.	1	2	4	1

Table 2. Comparison of Efficiency of I-MIBSC Scheme with Duan et al.'s Scheme (Designcryption)

8 Conclusion

In this paper, we have studied an existing multi-receiver ID-based signcryption scheme by You et al. [1]. They have proved the confidentiality of their scheme, but do not give any formal proof for unforgeability. We have shown a universal forgeability attack on their scheme whereby anybody can generate a valid signcryption of any message to any subset of legitimate users as if a legitimate user had generated it. We have also proposed an improved scheme and we have proved its security formally in the existing security model for multi-receiver ID-based signcryption schemes. We leave it as an open problem to investigate for more efficient schemes for multi-receiver ID-based signcryption.

References

1. Yong Yu, Bo Yang, Xinyi Huang, and Mingwu Zhang: *Efficient identity-based signcryption scheme for multiple receivers.* In:ATC 2007,LNCS 4610, pp. 1321, Springer-Verlag Berlin Heidelberg 2007.
2. Zheng Y.: *Digital signcryption or How to achieve cost(signature & Encrytpion) $\ll$ cost(signature) + cost(encryption).* In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165179. Springer, Heidelberg 1997.
3. Petersen H., Michels M.: *Cryptanalysis and improvement of signcryption schemes.* In: IEE proceedings-Computers and Digital Techniques 1998.
4. Bao F., Deng R.H.: *A signcryption scheme with signature directly verifiable by public key.* In: Imai, H., Zheng, Y. (eds.) PKC 1998. LNCS, vol. 1431, pp. 5559. Springer, Heidelberg 1998.
5. Zheng Y., Imai H.: *How to construct efficient signcryption schemes on elliptic curves.* In: Information Processing Letters 68(5), pp. 227233, 1998.
6. Malone-Lee J., Mao M: *Two birds one stone: signcryption using RSA.* In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 211226. Springer, Heidelberg 2003.
7. Jung H.Y., Lee D.H., Lim J.I., Chang K.S.: *New DSA-verifiable signcryption schemes.* In: Information Security Application-WISA 2001, pp. 463475, 2001.
8. Malone-Lee J.: *Identity based signcryption.* In: Cryptology ePrint Archive. Report 2002/098, 2002.
9. Libert B., Quisquator J.J.: *A new identity based signcryption scheme from pairings.* In: 2003 IEEE information theory workshop. Paris, France, pp. 155158, 2003.
10. Chow S.S.M., Yiu S.M., Hui L.C.K., Chow K.P.: *Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity.* In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 352369. Springer, Heidelberg 2004.

11. Boyen X.: Multipurpose identity based signcryption: a swiss army knife for identity based cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 383399. Springer, Heidelberg 2003.
12. Chen L., Malone-Lee J.: *Improved identity-based signcryption.* In: Vaudenay,S.(ed.) PKC 2005. LNCS, vol. 3386, pp. 362379. Springer, Heidelberg 2005.
13. Barreto P.S.L.M., Libert B., McCullagh N., Quisquater J.J.: *Efficient and provably-secure identity based signatures and signcryption from bilinear maps.* In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 515532. Springer, Heidelberg 2005.
14. Zheng Y.: *Signcryption and its applications in efficient public key solutions.* In: Okamoto, E. (ed.) ISW 1997. LNCS, vol. 1396, pp. 291312. Springer, Heidelberg 1998.
15. Duan S., Cao Z.: *Efficient and provably secure multi-receiver identity-based signcryption.* In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 195206. Springer, Heidelberg 2006.
16. Yuen T.H., Wei V.K.: *Fast and proven secure blind identity based signcryption from pairings.* In: Menezes, A.J. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 305322.Springer, Heidelberg 2005.
17. An J.H., Dodis Y., Rabin T.: *On the seurity of joint signature and encryption.* In:Cryptology ePrint Archive. Report 2002/046, 2002.
18. Baek J., Steinfeld R., Zheng Y.: *Formal proofs for the security of signcryption..* In:Public Key Cryptography - PKC 2002, volume 2274 of Lecture Notes in Computer Science, pages 80-98. Springer-Verlag, 2002.
19. Mu Y., Varadharajan V.: *Distributed signcryption.* In Roy, B., Okamoto, E. (eds.) INDOCRYPT 2000. LNCS, vol. 1977, pp. 155-164. Springer, Heidelberg (2000)
20. Steinfeld R., Zheng Y.: *A signcryption scheme based on integer factorization.* In:Okamoto, E., Pieprzyk, J.P., Seberry, J. (eds.) ISW 2000. LNCS, vol. 1975, pp. 308-322. Springer, Heidelberg (2000)
21. Yang G., Wong D.S., Deng X.: *Analysis and improvement of a signcryption scheme with key privacy.* In:Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 218-232. Springer, Heidelberg (2005).
22. Libert B., Quisquater J.-J.: *Efficient signcryption with key privacy from gap Diffie-Hellman groups.* In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 187-200. Springer, Heidelberg (2004).
23. Paulo S.L.M. Barreto, Benoit Libert, Noel McCullagh, Jean-Jacques Quisquater: *Efficient and Provably Secure Identity-Based Signatures and Signcryption from Bilinear Maps*In: B.Roy(ed.) ASIACRYPT 2005, LNCS, vol. 3788, pp. 515-532, 2005.
24. Adi Shamir: *Identity-Based Cryptosystems and Signature Schemes.* In: CRYPTO 1984, Lecture Notes in Computer Science, pp. 47-53, 1984.

A Proof of Correctness of I-MIBSC

In this section, we show that our improved scheme is consistent. If $\sigma = \langle y, U, X, W, T_i \rangle$ is a valid signcryption for a user with identity ID_i, then ***Designcrypt***$(\sigma, ID_A, ID_i, D_i)$ does the following.

1. Compute $Q_A = H_1(ID_A)$
2. Next, we observe that

$$\begin{aligned}\omega' &= W\hat{e}(U, D_i)\,\hat{e}(P_{pub}, T_i)^{-1}\\ &= \theta^{r_1}\omega\hat{e}(r_1P, sQ_i)\,\hat{e}(sP, r_1Q_i + r_1R)^{-1}\\ &= \hat{e}(P, R)^{r_1 s}\,\omega\hat{e}(P, Q_i)^{r_1 s}\,\hat{e}(P, Q_i)^{-r_1 s}\,\hat{e}(P, R)^{-r_1 s}\\ &= \omega\end{aligned}$$

3. Compute $m' \| Z' = c \oplus H_3(\omega') = m \| Z$
4. Compute $h_2' = H_2(ID_A \| U \| X \| m') = h_2$
5. Next, the checks $\omega' \overset{?}{=} \hat{e}(Z', P)$ and $\omega' \overset{?}{=} \hat{e}(X + h_2' Q_A, P_{pub})$ are performed. We show below that these tests will succeed and hence message m' will be returned.

– ***Check 1***

$$\omega' = \omega = \hat{e}(Z, P) = \hat{e}(Z', P)$$

– ***Check 2***

$$\begin{aligned}\hat{e}\left(X + h_2' Q_A, P_{pub}\right) &= \hat{e}\left(X + h_2 Q_A, P_{pub}\right)\\ &= \hat{e}\left(r_2 Q_A + h_2 Q_A, sP\right)\\ &= \hat{e}\left(\left(r_2 + h_2\right) Q_A, sP\right)\\ &= \hat{e}\left(\left(r_2 + h_2\right) D_A, P\right)\\ &= \omega = \omega'\end{aligned}$$

□

B Proof of Confidentiality of I-MIBSC

Theorem. *Our multi-receiver identity based signcryption scheme I-MIBSC is secure against any IND-MIBSC-CCA2 adversary $\mathcal{A}$ under the random oracle model if DBDHP is hard in $\mathbb{G}_1$.*

The challenger $\mathcal{C}$ receives an instance (P, aP, bP, cP, α) of the DBDH problem. His goal is to decide whether $\alpha = \hat{e}\left(P, P\right)^{abc}$ or not. Suppose there exists an IND-MIBSC-CCA2 adversary $\mathcal{A}$ for the proposed I-MIBSC scheme. We show that $\mathcal{C}$ can use $\mathcal{A}$ to solve the DBDH problem. $\mathcal{C}$ will set the random oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{KeyExtract}$, $\mathcal{O}_{Signcrypt}$ and $\mathcal{O}_{Designcrypt}$. The answers to the oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, and $\mathcal{O}_{H_3}$ are randomly selected, therefore, to maintain consistency, $\mathcal{C}$ will maintain three lists $L_1 = \langle ID_i, Q_i, x_i \rangle$, $L_2 = \langle ID_i, U, X, m, h_2 \rangle$, $L_3 = \langle \omega, h_3 \rangle$. We assume that $\mathcal{A}$ will ask for $H_1(ID)$ before ID is used in any key extraction, signcryption and designcryption queries. First, the adversary $\mathcal{A}$ outputs the list of identities $\mathcal{L} = \{ID_0^*, ID_1^*, \ldots, ID_t^*\}$ which is the set of target users. Then, the challenger $\mathcal{C}$ gives $\mathcal{A}$ the system parameters *params* consisting of P, $P_{pub} = cP$, $R = bP$, and $\theta = \hat{e}(R, P_{pub})\hat{e}(R, cP)$. The descriptions of the oracles follow.

Oracle $\mathcal{O}_{\mathbf{H_1}}(\mathbf{ID_i})$***.*** $\mathcal{C}$ checks if there exists a tuple (ID_i, Q_i, x_i) in L_1. If such a tuple exists, $\mathcal{C}$ answers with Q_i. Otherwise, $\mathcal{C}$ does the following.

1. If $ID_i \notin \mathcal{L}$, choose a new[2] $x_i \in_R \mathbb{Z}_q^*$ and set $Q_i = x_i P$.
2. If $ID_i \in \mathcal{L}$, choose a new $x_i \in_R \mathbb{Z}_q^*$ and set $Q_i = x_i P - R$.
3. Add the tuple (ID_i, Q_i, x_i) to L_1 and return Q_i.

Oracle $\mathcal{O}_{\mathbf{H_2}}(\mathbf{ID_i} \| \mathbf{U} \| \mathbf{X} \| \mathbf{m})$***.*** $\mathcal{C}$ checks if there exists a tuple (ID_i, U, X, m, h_2) in L_2. If such a tuple exists, $\mathcal{C}$ returns h_2. Otherwise, $\mathcal{C}$ chooses a new $h_2 \in_R \mathbb{Z}_q^*$, adds the tuple (ID_i, U, X, m, h_2) to L_2 and returns h_2.

Oracle $\mathcal{O}_{\mathbf{H_3}}(\omega)$***.*** $\mathcal{C}$ checks if there exists a tuple (ω, h_3) in L_3. If such a tuple exists, $\mathcal{C}$ returns h_3. Otherwise, $\mathcal{C}$ chooses a new $h_3 \in_R \{0,1\}^{n_1+n_3}$, adds the tuple (ω, h_3) in L_3 and returns h_3.

Oracle $\mathcal{O}_{\mathbf{KeyExtract}}(\mathbf{ID_i})$***.*** $\mathcal{C}$ does the following.

1. If $ID_i \in \mathcal{L}$ return $\perp$.

[2] By new, we mean that the random value chosen must not have been already chosen during an earlier execution.

2. If $ID_i \notin \mathcal{L}$, recover the tuple (ID_i, Q_i, x_i) from L_1 and return $D_i = x_i P_{pub} = cQ_i$.

Oracle $\mathcal{O}_{\mathbf{Signcrypt}}\left(\mathbf{m}, \mathbf{ID_A}, \mathcal{L}_1\right)$**.** On receiving this query, where $\mathcal{L}_1 = \{ID_1, ID_2, \ldots, ID_t\}$ is the list of intended receivers, $\mathcal{C}$ checks if $ID_A \in \mathcal{L}$. If not, $\mathcal{C}$ computes D_A using $\mathcal{O}_{KeyExtract}\left(ID_A\right)$, generates the signcryption in a normal way and returns it. Otherwise, that is, if $ID_A \in \mathcal{L}$, it chooses r, r' and a new $h_2 \in_R \mathbb{Z}_q^*$ and does the following.

1. Compute $U = r'P$
2. Compute $X = rP - h_2 \mathcal{O}_{H_1}(ID_A)$ and add the tuple (ID_A, U, X, m, h_2) to L_2.
3. Compute the following.
   (a) $Z = rP_{pub}$
   (b) $\omega = \hat{e}(Z, P)$
   (c) $y = \mathcal{O}_{H_3}(\omega) \oplus (m \| Z)$
   (d) For all $ID_j \in \mathcal{L}_1, T_j = r'(\mathcal{O}_{H_1}(ID_j) + R)$.
   (e) $W = \theta^{r'} \omega$
4. Return the signcrypted ciphertext $\sigma = \langle y, U, X, W, T_1, T_2, \ldots, T_t, \mathcal{L}_1 \rangle$.

Oracle $\mathcal{O}_{\mathbf{Designcrypt}}\left(\sigma, \mathbf{ID_A}, \mathbf{ID_j}\right)$**.** On receiving this query, where the signcryption $\sigma = \langle y, U, X, W, T_1, T_2, \ldots, T_t, \mathcal{L}_1 \rangle$, $\mathcal{C}$ checks if $ID_j \in \mathcal{L}$. If not, then $\mathcal{C}$ computes D_j using $\mathcal{O}_{KeyExtract}(ID_j)$, designcrypts σ in the normal way and returns what the designcryption algorithm returns. Otherwise, that is, if $ID_j \in \mathcal{L}$, then $\mathcal{C}$ tries to locate entries $(ID_A, U, X, m, h_2) \in L_2$ and $(\omega, h_3) \in L_3$ for some h_2, h_3, and ω under the constraints that $\omega = \hat{e}(P_{pub}, X + h_2 \mathcal{O}_{H_1}(ID_A))$, $(m \| Z) = h_3 \oplus y$, and $\omega = \hat{e}(Z, P)$. If no such entries are found, the oracle returns $\perp$. Otherwise, m is returned.

After the first query stage, $\mathcal{A}$ outputs two plaintext messages m_0 and m_1 of equal length, together with a senders's identity ID_A on which he wishes to be challenged. $\mathcal{A}$ now waits for a challenge signcrypted ciphertext built under the receivers' identities $ID_1, ID_2, \ldots, ID_t \subseteq \mathcal{L}$. Now, $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and signcrypts message m_b as follows.

1. Choose a new h_2 and $r \in_R \mathbb{Z}_q^*$.
2. Compute $U^* = aP$
3. Compute $X^* = rP - h_2 \mathcal{O}_{H_1}(ID_A)$ and add the tuple $(ID_A, U^*, X^*, m_b, h_2)$ to the list L_2.
4. Compute the following.
   (a) $Z^* = rP_{pub} = rcP$
   (b) $\omega = \hat{e}(Z^*, P)$
   (c) $y^* = \mathcal{O}_{H_3}(\omega) \oplus (m_b \| Z^*)$
   (d) $T_j^* = x_j aP$ for $1 \le j \le t$
   (e) $W^* = \alpha\omega$
5. Create a new label $\mathcal{L}^* = \{ID_1, ID_2, \ldots, ID_t\}$ and send the signcrypted ciphertext as $\sigma^* = \langle y^*, U^*, X^*, W^*, T_1, T_2, \ldots, T_t, \mathcal{L}^* \rangle$ to the adversary.

$\mathcal{A}$ can perform queries as above. However, it cannot query the designcryption oracle with the challenge signcrypted ciphertext or the signcryption oracle with messages m_0 or m_1 and ID_A as the sender. At the end of the simulation, $\mathcal{A}$ outputs a bit b' for which he believes that the challenge signcryption ciphertext is the signcryption of $m_{b'}$ from ID_A to $\mathcal{L}^*$. If the relation $b = b'$ holds, then $\mathcal{C}$ outputs 1 as the answer to the DBDH problem. Otherwise, it outputs 0. We have,

σ^* is a valid signcryption of m_b from ID_A to the receivers in $\mathcal{L}^*$

$$\begin{aligned}
&\Leftrightarrow \omega = W^* \hat{e}(T_j, P_{pub})^{-1} \hat{e}(U^*, D_j) \\
&\Leftrightarrow \alpha \hat{e}(T_j, P_{pub})^{-1} \hat{e}(U^*, D_j) = 1 \qquad \text{(because we have } W^* = \alpha\omega) \\
&\Leftrightarrow \alpha \hat{e}(x_j aP, cP)^{-1} \hat{e}(aP, (x_j - b)cP) = 1 \\
&\Leftrightarrow \alpha \hat{e}(x_j aP, cP)^{-1} \hat{e}(aP, x_j cP) \hat{e}(aP, -bcP) = 1 \\
&\Leftrightarrow \alpha \hat{e}(P, -abcP) = 1 \\
&\Leftrightarrow \alpha = \hat{e}(P, P)^{abc}
\end{aligned}$$

These calculations show that we get a correct ω if and only if $\alpha = \hat{e}(P, P)^{abc}$.

So, we can see that the challenger $\mathcal{C}$ has the same advantage in solving the DBDH problem as the adversary $\mathcal{A}$ has in distinguishing a valid signcrypted ciphertext from a random string. So, if there exists an adversary who can succeed in such a CCA2 attack with non-negligible advantage, that means there exists an algorithm to solve the DBDH problem with non-negligible advantage. Since this is not possible, no adversary can distinguish a valid signcrypted ciphertext from a random string with non-negligible advantage. Hence I-MIBSC is secure against any IND-MIBSC-CCA2 attack. □