On the Provable Security of Multi-Receiver Signcryption Schemes

S. Sharmila Deva Selvi, S. Sree Vivek*, Ragavendran Gopalakrishnan,
Naga Naresh Karuturi, C. Pandu Rangan*

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

Abstract. In ATC 2007, an identity based signcryption scheme for multiple receivers was proposed by Yu et al. They formally proved confidentiality of their scheme and also claim unforgeability without any proof. In this paper, we show that Yu et al.'s signcryption scheme is insecure by demonstrating an 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. Also, we point out a subtle flaw in the proof of confidentiality given by Yu et al. and show that the scheme does not provide confidentiality. Further, we propose a corrected version of Yu et al.'s scheme and formally prove its security (confidentiality and unforgeability) under the existing security model for signcryption.

In another direction, Fagen Li et al. have proposed a pairing based multi-recipient signcryption scheme. We also show that, the scheme proposed by Fagen Li et al. is not adaptive chosen ciphertext secure. We propose an improvement for Fagen Li et al.'s scheme and formally prove confidentiality under adaptive chosen ciphertext attack. Since all the previously reported multi-receiver schemes are shown to have flaws either here or else where, the schemes reported in this paper are the only correct and efficient schemes (identity based and pairing-based) for multi-receiver signcryption.

Keywords. Signcryption, Cryptanalysis, Identity Based Cryptography, PKI, 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 variety 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 authenticity. For achieving this, encryption schemes and signature schemes can be combined together. This was shown to be complex by An et al. in [2]. Signcryption, introduced by Zheng in 1997 [22], 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], [16], [19], [20], [5], [8], [13]). The security notion for signcryption was first formally defined in 2002 by Baek et al. in [3]. 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 identity based (ID-based) cryptosystem was introduced by Shamir [1] in 1984. The distinguishing characteristic of identity 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 user private key

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

derivation. An identity 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 (PKI).

Identity based signcryption schemes achieve the functionality of signcryption with the added advantage that identity based cryptography provides. In [14], Malone-Lee gave the first identity based signcryption scheme. Later it was found that Malone-Lee's scheme was not semantically secure. Since then, quite a few identity based signcryption schemes have been proposed ([11], [5], [13], [8], [17], [6]). To date, some of the most efficient identity based signcryption schemes are that of Chen et al. [6], and Barreto et al. [17]

Related Work and Our Contribution: 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, it results in huge computation and communication overhead. 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 identity based signcryption schemes till date. Duan et al. [9] were the first to come up with an identity based scheme for multi-receiver signcryption. Their scheme requires just one pairing operation to signcrypt a single message for multiple receivers. Chik How Tan [7] proved that, in spite of its efficiency and cleaver construct, [9] lacks adaptive chosen ciphertext security. Yu et al. [21] came up with another scheme with improved efficiency in the unsigncryption phase (their scheme requires one less pairing operation than Dual et al.'s). However, in this paper, we show that Yu et al.'s scheme [21] is insecure with respect to unforgeability and confidentiality, 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 Yu et al.'s scheme and prove its security (confidentiality and unforgeability) under the existing security model for signcryption. Thus, it turns out that ours is the only existing correct and provably secure identity based multi-receiver signcryption scheme.

To the best of our knowledge, three PKI based multi-receiver signcryption schemes are reported in the literature [12, 18, 10]. Zheng has given a construct for multi-receiver signcryption in [12]. However, it is known that Zheng's [12] signcryption scheme is not forward secure, anyone who obtains the sender's private key can recover the original message of a signcrypted ciphertext, which was shown in [23], following that Duan et al. [18] proposed a multi-receiver signcryption scheme, which is a combination of Zheng's multi-receiver signcryption and Bellare's concepts on multi-receiver setting for public key encryption [15]. However, [18] is insecure with respect to insider security, i.e. during the confidentiality game the senders private key is known to the adversary, knowing it the adversary can distinguish the message signcrypted in the ciphertext (Since the work is not published and is only available in the authors web page, we do not review and provide the formal attack on the scheme in [18]). Recently, Fagen Li et al. [10] proposed a multi-receiver signcryption scheme which depends on bilinear pairing. We show that [10] is not adaptive chosen ciphertext secure, also we propose a new multi-receiver signcryption scheme and formally prove the confidentiality and unforgeability of the new scheme. Thus, all the previously reported schemes are flawed ones and the only correct PKI based multi-receiver signcryption scheme is the scheme presented in this paper.

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 identity based and PKI based signcryption schemes for multiple receivers and the security models for those schemes. Next, in Section 3, we review Yu et al.'s multi-receiver identity based signcryption scheme and present the attacks on the scheme. In section 4, we propose the improved multi-receiver identity based signcryption scheme and the formal security proof for it. In Section 5, we review Fagen Li et al.'s multi-receiver signcryption scheme and show that it is not adaptive chosen ciphertext secure. Following that in section 6 we lay out the details of our new multi-receiver signcryption scheme and give the formal proof for confidentiality and unforgeability of the new scheme and in section 7 we conclude 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 $P, Q, R \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$
  - $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$
  - $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$
  - $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- ***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.

Definition 1. *(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. 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.

Definition 2. *(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 $\hat{e}(P, P)^{abc}$. 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.

Definition 3. *(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 $\alpha = \hat{e}(P, P)^{abc}$. 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.

2.3 ID-Based Signcryption for Multiple Receivers

A generic multi-receiver IBSC 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 a set of receivers with identities ID_1, ID_2, ..., ID_t, *Alice* with identity ID_{Alice} and private key D_{Alice} runs this algorithm to obtain the signcrypted ciphertext σ.
- ***Unsigncrypt***$(\sigma, ID_{Alice}, ID_{Bob}, D_{Bob})$. When *Bob* with identity ID_{Bob} and private key D_{Bob} receives the signcrypted ciphertext σ from *Alice* with identity ID_{Alice}, Bob runs this algorithm to obtain either the plain text m or *invalid* according as whether σ was a valid signcryption from Alice to Bob or not.

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

2.4 Security Model for Multi-Receiver ID-Based Signcryption (MIBSC)

We describe below the security models for *confidentiality* and *unforgeability* given by [5] and these are the strongest security notions for MIBSC schemes.

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.

Setup Phase: The challenger $\mathcal{C}$ runs the *Setup* algorithm and sends the system public parameters to the adversary $\mathcal{A}$.

Phase I: In this phase, $\mathcal{A}$ makes polynomial number of queries to the following oracles.

1. *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 $\mathcal{A}$.
2. *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 $\mathit{Keygen}(ID_A)$ and returns to $\mathcal{A}$, the signcrypted ciphertext σ.
3. *Unsigncrypt 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 $\mathit{Keygen}(ID_B)$ and returns the message m to $\mathcal{A}$. It returns *invalid* if σ is an invalid signcryption from ID_A to ID_B.

Challenge: $\mathcal{A}$ 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 $\sigma^* = \mathit{Signcrypt}(m_b, ID_A, \{ID_1, ID_2, \ldots, ID_t\}, D_A)$. σ^* is returned to $\mathcal{A}$ as the challenge signcrypted ciphertext.

Phase II: $\mathcal{A}$ is allowed to make polynomial number of new queries as in ***Phase I:*** with the restrictions that it should not query the *Unsigncryption Oracle* for the unsigncryption 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 $ID_1, ID_2, \ldots, ID_t$.

Guess: 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-CMA) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

Setup Phase: 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 $\mathcal{A}$.

Training Phase: The adversary $\mathcal{A}$ makes polynomial number of queries to the oracles as described in ***Phase I:*** of the confidentiality game.

Forgery: $\mathcal{A}$ produces a signcrypted ciphertext σ and wins the game if the private key of sender identity ID_A was not queried in the ***Training Phase:***, the output of $\mathit{Unsigncrypt}(\sigma, ID_A, ID_B, D_B)$ is not *invalid* and σ is not the output of any previous queries to the *Signcrypt Oracle* with ID_A as sender.

2.5 Security Model for PKI Based Multiple Receiver Signcryption for (MSC)

We describe below the security models for *confidentiality* and *unforgeability* for PKI based multi-receiver signcryption scheme and these are the strongest security notions for MSC schemes.

Confidentiality: A multi-receiver signcryption scheme is semantically secure against adaptive chosen ciphertext attack (IND-MSC-CCA2), if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

The challenger $\mathcal{C}$, takes the security parameter κ as input and runs *Keygen* to generate multiple key pairs (sk_{R_i}, pk_{R_i}), $(i = 1, \ldots, n)$ for the n receivers. All sk_{R_i} are kept secret while pk_{R_i} the corresponding public keys are given to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ performs a series of queries in an adaptive fashion in this phase. The queries allowed are given below:
Signcryption oracle: $\mathcal{A}$ produces a message $m \in \mathcal{M}$ and requests the result of the operation *Signcrypt*(m, sk_S, pk_{R_1}, ..., pk_{R_n}).
Unsigncryption oracle: $\mathcal{A}$ produces a ciphertext σ and an arbitrary sender public key pk_S and requires the result of the operation $Unsigncryt(\sigma, pk_S, sk_{R_i})$.
These queries may be asked adaptively, i.e. each query may depend on the answers to previous ones.
Challenge: At the end of *Phase 1*, $\mathcal{A}$ generate two equal length plaintexts m_0 and m_1 and sends it to $\mathcal{C}$. Now, $\mathcal{C}$ flips $b \in_R \{0,1\}$ and computes $\sigma^* = signcrypt(\ m_b, sk_S, pk_{R_1}, \ldots, pk_{R_n})$ and returns it to $\mathcal{A}$.
Phase 2: $\mathcal{A}$ can perform polynomial number of queries adaptively again as in *Phase 1* but he cannot make an unsigncryption query on σ^*.
Guess: $\mathcal{A}$ outputs a bit b' and wins the game if $b' = b$.

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

The challenger $\mathcal{C}$, takes the security parameter κ as input and runs *Keygen* to generate multiple key pairs (sk_{R_i}, pk_{R_i}), $(i = 1, \ldots, n)$ for the n receivers. All sk_{R_i} are kept secret while pk_{R_i} the corresponding public keys are given to $\mathcal{A}$.

Training Phase: The adversary $\mathcal{A}$ makes polynomial number of queries to the oracles as described in ***Phase I:*** of the confidentiality game.
Forgery: $\mathcal{A}$ produces a signcrypted ciphertext σ^* and wins the game if the private key of sender S was not queried in the ***Training Phase:***, the output of $Unsigncryt(\sigma^*, pk_S, sk_{R_i})$ is not *invalid* and σ^* is not the output of any previous queries to the *Signcrypt Oracle* with S as the sender.

3 Yu et al.'s ID-Based Multi-Receiver Signcryption Scheme (Y-MIBSC)

In this section, we review Yu et al.'s identity based multi-receiver signcryption scheme (Y-MIBSC) and show that the scheme does not provide both unforgeability as well as confidentiality.

3.1 Review of Y-MIBSC

The Y-MIBSC scheme in [21] has the following algorithms.

Setup(*k*): The security parameter of the scheme is k, $\mathbb{G}_1$, $\mathbb{G}_2$ are two cyclic groups of prime order q, 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 $R \in \mathbb{G}_1 \setminus \{\mathbf{0}_{\mathbb{G}_1}\}$ 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 $\langle \mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, R, \theta, \hat{e}, H_1, H_2, H_3 \rangle$.

Keygen(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.

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. Chooses $r \in_R \mathbb{Z}_q^*$
2. Computes 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 $\sigma = \langle y, U, X, W, T_1, T_2, \ldots, T_n, \mathcal{L} \rangle$, where $\mathcal{L}$ is the list of receivers who can decrypt the message. Here, T_i is meant for the receiver ID_i.

Unsigncrypt(σ, ID_A, ID_i, D_i) : A receiver with identity ID_i uses his secret key D_i to unsigncrypt $\sigma = \langle y, U, X, W, T_i, \mathcal{L} \rangle$ from ID_A as follows.

1. Computes 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 $\omega' = \hat{e}(P_{pub}, X + h_2' Q_A)$, returns m'. Otherwise, returns *invalid*.

3.2 Attack on Y-MIBSC

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

Attack on Unforgeability: 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 $ID_1, ID_2, \ldots, ID_n$, *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 $1 \leq j \leq n$
3. $\sigma^* = \langle y^*, U^*, X^*, W^*, T_1^*, T_2^*, \ldots, T_n^*, \mathcal{L}^* \rangle$ is the signature of *Alice* on message m^* generated by *Eve* for the list of users $\mathcal{L}^*$ with identities $\{ID_j\}_{1 \leq j \leq n}$

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

Unsigncrypt($\sigma^* = \langle y^*, U^*, X^*, W^*, T_1^*, T_2^*, \ldots, T_n^*, L^* \rangle, ID_{Alice}, ID_j, D_j$). A receiver with identity ID_j uses his secret key D_j to unsigncrypt 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 $\omega' \stackrel{?}{=} \hat{e}(P_{pub}, X^* + h_2' Q_{Alice})$ is performed. We show below that this test will succeed and hence message m^* will be returned.

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

From this it is clear that *Eve* can succeed in generating a signcryption of message m^* with Alice as sender and identities ID_j, $1 \leq j \leq n$ 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.

Attack on Confidentiality : The scheme in [21] does not provide confidentiality. This can be shown by the following:

Let m_0 and m_1 be the two messages given by the adversary to the challenger during the challenge phase of the confidentiality game. On seeing the challenge ciphertext $\sigma^* = \langle y^*, U^*, X^*, W^*, T_i^*, \mathcal{L}^* = \{ID_1, ID_2, \ldots, ID_n\}\rangle$, the adversary will be able to compute $h_2^0 = H_2(X^* \| m_0)$ and $w^0 = \hat{e}(X^* + h_2^0 Q_{ID_1}, P_{pub})$. Then, he can compute $m' = y^* \oplus H_3(w^0)$. If $m' = m_0$ then adversary knows that σ^* is signcryption of m_0, else, σ^* is signcryption of m_1.

4 Improved Multi-Receiver ID-Based Signcryption Scheme (I-MIBSC)

In this section, we propose an improved version of Y-MIBSC, which we formally prove to be secure.

4.1 Scheme

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.

Setup(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 $R \in \mathbb{G}_1 \backslash \{\mathbf{0}_{\mathbb{G}_1}\}$ 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 $\langle \mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, R, \theta, \hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2, H_1, H_2, H_3\rangle$.

Keygen(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.

Signcrypt($m, ID_A, ID_1, ID_2, \ldots, ID_n, D_A$): 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) $y = (m \| Z \| X) \oplus H_3(\omega)$
   (g) $W = \theta^{r_1} \omega$
   (h) $T_i = r_1(Q_i + R)$, for $1 \le i \le n$
3. The signcrypted ciphertext is $\sigma = \langle y, U, W, T_1, T_2, \ldots, T_n, \mathcal{L} \rangle$, where $\mathcal{L}$ is the list of receivers who can decrypt the message. Here, T_i is meant for the receiver ID_i.

Unsigncrypt(σ, ID_A, ID_i, D_i): A receiver with identity ID_i uses his secret key D_i to unsigncrypt $\sigma = \langle y, U, W, T_i, \mathcal{L} \rangle$ 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) $m' \| Z' \| X' = y \oplus H_3(\omega')$
   (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 *invalid*.

We prove the correctness of our scheme in Appendix-A and confidentiality of our scheme in Appendix-B.

4.2 Proof of Unforgeability of I-MIBSC

Theorem 1. *Our multi-receiver identity based signcryption scheme I-MIBSC is secure against any EUF-MIBSC-CMA 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-CMA 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 $\mathcal{O}_{Unsigncrypt}$. 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 unsigncryption queries. First, the adversary $\mathcal{A}$ outputs the identity ID_A of the sender whose signcryption he claims to be able to forge.
Setup Phase: $\mathcal{C}$ gives $\mathcal{A}$ the system parameters *params*, consisting of P, $P_{pub} = bP$, R, $\theta = \hat{e}(R, P_{pub} = \hat{e}(R, bP)$.

Training Phase: $\mathcal{A}$ interacts with $\mathcal{C}$ by accessing the various oracles provided by $\mathcal{C}$ The descriptions of these oracles are presented below.

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 $ID_i \ne ID_A$, 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.

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

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 *invalid*.
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 $\mathcal{L} = \{ID_1, ID_2, \ldots, ID_n\}$ 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) $y = \mathcal{O}_{H_3}(\omega) \oplus (m\|Z\|X)$
   (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 $\sigma = \langle y, U, W, T_1, T_2, \ldots, T_n, \mathcal{L}\rangle$.

Oracle $\mathcal{O}_{\mathbf{Unsigncrypt}}(\sigma, \mathbf{ID_i}, \mathbf{ID_j})$***.*** On receiving this query, where the signcryption $\sigma = \langle\, y, U, W, T_1, T_2, \ldots, T_n, \mathcal{L}\rangle$, $\mathcal{C}$ checks if $ID_j = ID_A$. If not, $\mathcal{C}$ computes D_j using $\mathcal{O}_{KeyExtract}(ID_j)$, unsigncrypts σ in the normal way and returns what the unsigncryption 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 $\omega = \hat{e}(P_{pub}, X + h_2\mathcal{O}_{H_1}(ID_i))$, $(m\|Z\|X) = h_3 \oplus y$, and $\omega = \hat{e}(Z, P)$. If no such entries are found, the oracle returns *invalid*. Otherwise, m is returned.

Forgery: Eventually, $\mathcal{A}$ outputs a forged signcryption $\sigma' = \langle y', U', W', T_1', T_2', \ldots, T_n', \mathcal{L}'\rangle$ on some message m' from the sender ID_A to users in the set $\mathcal{L}' = \{ID_1, ID_2, \ldots, ID_n\}$, with $ID_A \notin \mathcal{L}'$.

Now, $\mathcal{C}$ unsigncrypts the ciphertext σ' with the private key of 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 $\sigma_1 = \langle y_1, U_1, W_1, T_1'^1, T_2'^1, \ldots, T_n'^1, \mathcal{L}'\rangle$ and $\sigma_2 = \langle y_2, U_2, W_2, T_1'^2, T_2'^2, \ldots, T_n'^2, \mathcal{L}'\rangle$ on some message m' from the sender ID_A to users in the set $\mathcal{L}' = \{ID_1, ID_2, \ldots, ID_n\}$, with $ID_A \notin \mathcal{L}'$. $\mathcal{C}$ unsigncrypts 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 $\mathcal{C}$ 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 $\mathcal{C}$ 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-CMA attack. □

5 Li et al.'s Multi-receiver Signcryption Scheme (L-MSC)

In this section, we review Li et al.'s multi-receiver signcryption scheme (L-MSC) as described in [10] and show that the scheme is not adaptive chosen ciphertext secure.

5.1 Review of L-MSC

This scheme has the following three algorithms. Given κ and l as the two security parameters, the sender and the receiver agrees up on two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^\kappa$ (the number of bits required to represent $\mathbb{G}_1$ is l), a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a generator $P \in_R \mathbb{G}_1$. They also choose three cryptographic hash functions $H_1 : \mathbb{G}_1 \rightarrow \{0,1\}^{n_1}$, $H_2 : \{0,1\}^{n_1+(n+1)l} \rightarrow \mathbb{G}_1$ and $H_3 : \mathbb{G}_1^3 \rightarrow \{0,1\}^l$.

Extract: User U does the following to extract the private/public key pair:

- Choose $x_U \in_R \mathbb{Z}_q^*$ and sets it as his private key.
- Sets the public key as $Y_U = x_U P$.

The sender is represented by S and the set of receivers are denotes as R_i, where ($i = 1$ to n)

Signcrypt: Given a message m, a set of receivers $R_1, R_2, \ldots, R_n$ and the sender S executes the following steps:

- Chooses $r \in_R \mathbb{Z}q^*$ and $R \in_R \mathbb{G}_1$.
- Computes $U = rP$.
- Computes $c = m \oplus H_1(R)$.
- Computes $V = x_S H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$.
- Computes $Z_i = R \oplus H_3(U, Y_{R_i}, rY_{R_i})$ for $i = 1, \ldots, n$.

The ciphertext is $\sigma = (U, c, V, Z_1, \ldots, Z_n)$.

Unsigncrypt: On receiving a ciphertext $\sigma = (U, c, V, Z_1, \ldots, Z_n)$, each receiver R_i performs the following steps.

- Computes $R = Z_i \oplus H_3(U, Y_{R_i}, x_{R_i} U)$.
- Computes $m = c \oplus H_1(R)$.
- Computes $H = H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$.
- Accepts the message if and only if $\hat{e}(P, V) \stackrel{?}{=} \hat{e}(Y_S, H)$, return *invalid* otherwise.

5.2 Attack on Li et al.'s Multi-receiver Signcryption Scheme (L-MSC)

The above scheme is insecure against adaptive chosen ciphertext security, we launch the attack on the confidentiality of the scheme as follows.

Attack on Confidentiality The crucial argument in the confidentiality proof of [10] is that the adversary $\mathcal{A}$ will not realize that the challenge ciphertext σ^* is not a valid signcryption for the sender's private key and public keys $Y_{R_1}, \ldots, Y_{R_n}$ of the set of receivers $R_1, \ldots, R_n$ unless it asks for the hash value $H_3(aP, bP, abP)$, which reduces the problem to CDH. We prove that this is not the only means for $\mathcal{A}$ to decrypt the challenge ciphertext. $\mathcal{A}$ can manipulate the ciphertext by attaching an arbitrary message and generating the signature ($\mathcal{A}$ knows the secret key of the sender in confidentiality game) for the manipulated ciphertext. Now, $\mathcal{A}$ can make use of the oracles to decrypt the altered ciphertext and thus decrypts without solving any hard problem.

During the IND-MSC-CCA2 game, the adversary $\mathcal{A}$, on getting the challenge ciphertext $\sigma^* = (U^*, c^*, V^*, Z_1^*, \ldots, Z_n^*)$, can do the following to identify whether σ^* is a signcryption of m_0 or m_1 with out solving any hard problem.

- $\mathcal{A}$ computes $c' = c^* \oplus m'$.
- Chooses an arbitrary sender, for which it knows the private key (let the private key be x_A).
- Computes $V' = x_A H_2(c', U^*, Y_{R_1}, \ldots, Y_{R_n})$ (note: $\mathcal{A}$ can choose the receivers of the newly generated ciphertext as any subset of receivers from the challenge ciphertext), where all values except c' are the same as in the challenge ciphertext.

- Now, $\sigma' = (U^*, c', V', Z_1^*, \ldots, Z_n^*)$ is a valid signcryption from user A to multiple receivers R_i, where $i = 1$ to n.
- Since it is a valid ciphertext and is also not the exact challenge ciphertext, $\mathcal{A}$ can obtain the unsigncryption of σ' during *Phase 2* from $\mathcal{C}$.

$Unsigncryt(\sigma')$ produces $m_b \oplus m'$. As m' is selected by $\mathcal{A}$ and it also knows m_0 and m_1, it can easily identify whether c^* is a signcryption of m_0 or m_1.

6 New Multi-receiver Signcryption Scheme (N-MSC)

The bug identified in this scheme is not a trivial one but it can be extirpated by altering the scheme according to the guideline of An et al. [2]. We propose a new multi-receiver signcryption scheme and prove the confidentiality against adaptive chosen ciphertext attack and unforgeability against chosen message attack in the random oracle model in this section.

6.1 Scheme.

The improved scheme also has three algorithms. First, given κ as the security parameter, the sender and the receiver agrees up on two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^\kappa$ (Let the number of bits required to represent a message m be n_1), a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a generator $P \in_R \mathbb{G}_1$. They also choose four cryptographic hash functions $H_1 : \mathbb{G}_1 \rightarrow \{0,1\}^{n_1+n_2}$, $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_3 : \mathbb{G}_1^3 \rightarrow \mathbb{G}_1$. and $H_4 : \{0,1\}^* \rightarrow \{0,1\}^{n_2}$

Extract: User U does the following to extract the private/public key pair:

- Chooses $x_U \in_R \mathbb{Z}_q^*$ and sets it as his private key.
- Sets the public key as $Y_U = x_U P$.

The sender is represented by S and the set of receivers are denotes as R_i, where ($i = 1$ to n)

Signcrypt: Given a message m, a set of receivers $R_1, R_2, \ldots, R_n$, the sender S executes the following steps to perform signcryption:

- Chooses $r \in_R \mathbb{Z}q^*$ and $R \in_R \mathbb{G}_1$.
- Computes $U = rP$ and $h = H_4(m, R, Y_S, Y_{R_1}, \ldots, Y_{R_n})$.
- Computes $c = (m \| h) \oplus H_1(R)$.
- Computes $V = x_S H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$.
- Computes $Z_i = R \oplus H_3(U, Y_{R_i}, rY_{R_i})$, for $i = 1, \ldots, n$.

The ciphertext is $\sigma = (U, c, V, Z_1, \ldots, Z_n)$.

Unsigncrypt: On receiving a ciphertext $\sigma = (U, c, V, Z_1, \ldots, Z_n)$, each receiver R_i performs the following steps.

- Computes $R = Z_i \oplus H_3(U, Y_{R_i}, x_{R_i} U)$.
- Computes $h' = H_4(m, R, Y_S, Y_{R_1}, \ldots, Y_{R_n})$.
- Retrieves the message m and h as $(m \| h) = c \oplus H_1(R)$.
- Computes $H = H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$.
- Accepts the message if $\hat{e}(P, V) \stackrel{?}{=} \hat{e}(Y_S, H)$ and $h \stackrel{?}{=} h'$, otherwise rejects the ciphertext σ.

Remark 1: For a signcryption scheme to be secure in multi-user setting it is required to have the following binding in the *Encrypt-then-Sign* (*EtS*) paradigm.

- Encryption should involve the identity of sender,
- The signature should involve the identity of the receiver.

This key issue was proved by An, Dodis and Rabin in [2]. The scheme by Fagen Li et al. [10] also uses the (EtS) paradigm, but it fails to achieve the above said property. Thus, during the confidentiality game, the adversary is able to alter the signature part of the challenge ciphertext and produce a valid ciphertext as if it is signcrypted by a legitimate user for some other message (It can be the signature of the actual sender itself, as the secret key of the sender is known to the adversary during the confidentiality game to prove the insider security). This led to the weakness on adaptive chosen ciphertext security of [10] as mentioned in the attack, but we counter it by following the guidelines in [2].

6.2 Proof of Confidentiality of N-MSC

Theorem 2. *Our multi-receiver signcryption scheme N-MSC is secure against any IND-N-MSC-CCA2 adversary $\mathcal{A}$ under the random oracle model if CDHP is hard in $\mathbb{G}_1$.*

The challenger $\mathcal{C}$ uses the adversary $\mathcal{A}$, who is capable of breaking the IND-N-MSC-CCA2 security of N-MSC to solve the CDH problem in polynomial time. Let (P, aP, bP) be a random instance of the CDH problem $\mathcal{C}$ has received. $\mathcal{C}$ starts the game by choosing a receiver $R^* \in \{R_1, \ldots R_n\}$ and sets the public key of the user R^* as $Y^* = bP$, which is the challenge public key and gives the public parameters to $\mathcal{A}$.

Phase I: $\mathcal{A}$ then adaptively performs queries on the various oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{H_4}$, $\mathcal{O}_{Signcryption}$ and $\mathcal{O}_{Unsigncryption}$.

To handle these queries, $\mathcal{C}$ maintains lists L_i that keeps track of the answers given to oracle queries by $\mathcal{A}$ on queries to H_i for $(i = 1, 2, 3, 4)$. Upon a query by $\mathcal{A}$ on the hash oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$ and $\mathcal{O}_{H_4}$, $\mathcal{C}$ responds in the following way: $\mathcal{C}$ first checks in the respective lists L_i, whether the oracle is queried previously for the same input; if so, retrieves and returns the corresponding value; if not queried previously, randomly generate an element from the output range of the corresponding hash function, returns it to $\mathcal{A}$ and stores the input and output values in the corresponding list.

$\mathcal{O}_{Signcryption}$ *queries:* To face the signcryption query on a plaintext m chosen by $\mathcal{A}$, $\mathcal{C}$ does the following:

- If the public key of the sender is not the target public key, i.e. $Y_S \neq Y^*$ then $\mathcal{C}$ proceeds as per the **Signcrypt** algorithm.
- If the public key of the sender is the target public key i.e. $Y_S = Y^*$ then $\mathcal{C}$ proceeds as follows:
  - Chooses $r \in_R \mathbb{Z}q^*$ and $R \in_R \mathbb{G}_1$.
  - Computes $U = rP$ and queries $h_4 = \mathcal{O}_{H_4}(m, R, Y_S, Y_{R_1}, \ldots, Y_{R_n})$ and $h_1 = \mathcal{O}_{H_1}(R)$. If entries $(m, R, Y_S, Y_{R_1}, \ldots, Y_{R_n}, h_4)$ and (R, h_1) already exists in the list L_4 and L_1 respectively, $\mathcal{C}$ uses them.
  - Computes $c = (m \| h_4) \oplus h_1$.
  - Chooses $x' \in_R \mathbb{Z}_q^*$, sets $H_2' = x'P$ and stores the tuple $\langle c, U, Y_{R_1}, \ldots, T_{R_n}, H_2' \rangle$ in the list L_2.
  - Computes $V = x'bP$.
  - Queries $h_{3_i} = \mathcal{O}_{H_3}(U, Y_{R_i}, rY_{R_i})$ for $i = 1, \ldots, n$. If all entries $(U, Y_{R_i}, rY_{R_i}, h_{3_i})$, for $i = 1, \ldots, n$ already exists in the list L_3, $\mathcal{C}$ uses them.
  - Computes $Z_i = R \oplus h_{3_i}$ for $i = 1, \ldots, n$.
  - The ciphertext $\sigma = (U, c, V, Z_1, \ldots, Z_n)$ is then returned as the signcryption of the message m with Y^* as the sender to $\mathcal{A}$.

$\mathcal{O}_{Unsigncryption}$ *queries:* Upon receiving an unsigncryption query on a ciphertext $\sigma = (U, c, V, Z_1, \ldots, Z_n)$ and a senders public key Y_S both chosen by $\mathcal{A}$, $\mathcal{C}$ proceeds as follows:

- If the public key of the receiver is not the target public key, i.e. $Y_R \neq Y^*$ then $\mathcal{C}$ proceeds as per the **Unsigncrypt** algorithm.
- If the public key of the receiver is the target public key i.e. $Y_R = Y^*$ then $\mathcal{C}$ proceeds as follows:
  - Retrieves $(U, Y_{R_i}, rY_{R_i}, h_{3_i})$, where $(0 \leq i \leq q_{H_3})$ from list L_3, if the tuple does not exist return *invalid*.
  - Computes $R = Z_i \oplus h_{3_i}$.
  - Retrieves the message m' and h_4' by computing $(m' \| h_4') = c \oplus h_1$ where h_1 is retrieved from the list L_1 by searching for a tuple (R, h_1) in it, if not present returns *invalid*.
  - Retrieves $(m, R, Y_S, Y_{R_1}, \ldots, Y_{R_n}, h_4)$ from the list L_4, if the tuple does not exist returns *invalid*.
  - Retrieves $(c, U, Y_{R_1}, \ldots, Y_{R_n}, H_2')$ from the list L_2, if the tuple does not exist returns *invalid*.

- Returns the message m to $\mathcal{A}$ if and only if $\hat{e}(P, V) \stackrel{?}{=} \hat{e}(Y_S, H_2')$ and $h_4 \stackrel{?}{=} h_4'$, otherwise return *invalid* and *Abort*.

Challenge: At the end of ***Phase I***, $\mathcal{A}$ produces two plaintexts m_0 and m_1 to $\mathcal{C}$ and requires a challenge ciphertext encrypted with the receivers public keys that includes the challenge public key Y^*. $\mathcal{C}$ chooses a random bit $b \in_R \{0,1\}$ and signcrypts m_b as follows.
- Computes $U^* = aP$ and chooses $R^* \in_R \mathbb{G}_1$.
- Queries the oracles $\mathcal{O}_{H_4}$ and $\mathcal{O}_{H_1}$ to obtain $h_4^* = \mathcal{O}_{H_4}(m, R, Y_S, Y_{R_1}, \ldots, Y_{R_n})$ and $h_1^* = \mathcal{O}_{H_1}(R)$ respectively.
- Computes $c^* = (m_b \| h_4^*) \oplus h_1^*$.
- Queries the oracle $\mathcal{O}_{H_2}$ and obtains $H_2'^* = \mathcal{O}_{H_2}(c^*, U^*, Y_{R_1}, \ldots, Y_{R_n})$.
- Computes $V^* = x_S H_2'$.

$\mathcal{C}$ then chooses $\{Z_1^*, \ldots, Z_n^*\} \in_R \mathbb{G}_1$ and sends the challenge ciphertext $\sigma^* = (U^*, c^*, V^*, Z_1^*, \ldots, Z_n^*)$ to $\mathcal{A}$.

Phase II: $\mathcal{A}$ adaptively performs series of queries in this phase also but with the restriction that, it is not allowed to get the decryption of the challenge ciphertext σ^*. These queries are handled by $\mathcal{C}$ as those in the first stage.

(Note that $\mathcal{A}$ cannot realize that σ^* is not a valid signcryption for the senders private key x_S and the receiver public key Y^* unless it asks for the hash value $H_3(U^*, Y^*, aY^*)= H_3(aP, bP, abP)$. In that case, the solution of the Computational Diffie-Hellman problem would be inserted in the list L_3 and it does not matter to the challenger, even if the simulation of $\mathcal{A}$'s view is no longer perfect.)

Guess: At the end of ***Phase II***, $\mathcal{A}$ outputs a bit b'.

$\mathcal{C}$ ignores the result of $\mathcal{A}$. $\mathcal{C}$ is only interested in the tuple in the list L_3 which is of the form $(aP, bP, X, .)$. $\mathcal{C}$ now checks whether $\hat{e}(P, X) \stackrel{?}{=} \hat{e}(aP, bP)$ for all entries of the list L_3 and if this relation holds, stops and outputs X as the solution of the CDH problem instance it has received. If no tuple of this kind satisfies the equality, $\mathcal{C}$ stops and outputs *invalid*. The probability that $\mathcal{C}$'s answer to the CDH problem is correct, is same as the probability that $\mathcal{A}$ queries $\mathcal{O}_{H_3}(aP, bP, abP)$ and this implies that C can solve the CDH problem with non-negligible advantage and this is a contradiction. □

6.3 Proof of Unforgeability of N-MSC

Theorem 3. *Our multi-receiver signcryption scheme N-MSC is secure against any EUF-N-MSC-CMA adversary $\mathcal{A}$ under the random oracle model if CDHP is hard in $\mathbb{G}_1$.*

The challenger $\mathcal{C}$ uses the adversary $\mathcal{A}$, who is capable of breaking the IND-N-MSC-CMA security of N-MSC to solve the CDH problem in polynomial time. Let (P, aP, bP) be a random instance of the CDH problem. $\mathcal{C}$ simulates $\mathcal{A}$'s queries in the game of unforgeability as defined in the confidentiality game. It starts the game by choosing a sender S^* and sets $Y^* = aP$ as the public key of the user S^*, which is the challenge public key.

Training Phase: $\mathcal{A}$ is allowed to adaptively perform queries on the various oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{H_4}$, $\mathcal{O}_{Signcryption}$ and $\mathcal{O}_{Unsigncryption}$ (Note that the definition of these oracles are same as that in the confidentiality proof in section 6.2).

Forgery: Finally, $\mathcal{A}$ produces a forged signcryption $\sigma^* = (U^*, c^*, V^*, Z_1^*, \ldots, Z_n^*)$ on the message m^* (i.e. σ^* was not produced by signcryption oracle $\mathcal{O}_{Signcryption}$ as an output for the signcryption query on the message m^* with S^* as sender). $\mathcal{C}$ can very well unsigncrypt and verify the validity of the forged signcryption σ^* because $\mathcal{C}$ knows the secret key of all the receivers.

If the forged signcryption passes the verification then $\mathcal{C}$ can obtain the solution for CDH problem by performing the following steps:

- $\mathcal{C}$ checks list L_2 whether $\langle c^*, U^*, Y_{R_1}, \ldots, Y_{R_n}, H_2' \rangle$ was previously queried by $\mathcal{A}$ during the *Training Phase*. If not queried by $\mathcal{A}$, $\mathcal{C}$ *aborts* the game else if it was queried, the corresponding H_2' value was set by $\mathcal{C}$ to be bP.

– Thus, V^* which is obtained from the forged signcryption is nothing but $V^* = x_{S^*} H_2' = abP$, which is the result of the CDH instance that $\mathcal{C}$ has received.

So, we can see that $\mathcal{C}$ 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, it 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, N-MSC is secure against any EUF-N-MSC-CMA attack

7 Conclusion

In this paper, we presented the cryptanalysis of the multi-receiver identity based signcryption scheme by Yu et al. [21] and showed an universal forgeability attack on the 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. Also, we showed that the scheme does not provide confidentiality, i.e. it is not indeed adaptive chosen ciphertext secure. We have also proposed an improved scheme and proved its security formally in the existing security model for multi-receiver identity based signcryption schemes.

We have also cryptanalyzed a PKI based multi-receiver signcryption scheme by Fagen Li et al. [10] by demonstrating an attack on the confidentiality of the scheme. We have also proposed a new multi-receiver signcryption scheme and have proved both confidentiality and unforgeability formally in the random oracle model.

References

1. Adi Shamir: *Identity-Based Cryptosystems and Signature Schemes*. In: CRYPTO 1984, Lecture Notes in Computer Science, pp. 47-53,1984.
2. An J. H., Dodis Y., Rabin T: *On the security of joint signature and encryption.* In: Proceedings of Advances in Cryptology- EUROCRYPT 2002, LNCS, vol. 2332, Springer-Verlag, pp. 83-107, 2002.
3. 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.
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. 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.
6. Chen L., Malone-Lee J.: *Improved identity-based signcryption.* In: Vaudenay,S.(ed.) PKC 2005. LNCS, vol. 3386, pp. 362379. Springer, Heidelberg 2005.
7. Chik How Tan: *On the Security of Provably Secure Multi-Receiver ID-Based Signcryption Scheme.* In: IEICE-Transaction on Fundamentals of Electronics, Communication & Computer Science, vol. E91-A, Number 7, pp. 1836-1838. 2008.
8. 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.
9. 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.
10. Fagen Li, Yupu Hu, Shuanggen Liu: *Efficient and provably secure multi-recipient signcryption from bilinear pairings.* In: Wuhan University Journal of Natural Sciences, vol. 12, Number 1, pp. 17-20, January, 2007.
11. Libert B., Quisquator J.J.: *A new identity based signcryption scheme from pairings.* In: 2003 IEEE information theory workshop. Paris, France, pp. 155158, 2003.
12. 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.
13. 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).
14. Malone-Lee J.: *Identity based signcryption.* In: Cryptology ePrint Archive. Report 2002/098, 2002.
15. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. *Public-key encryption in a multi-user setting: Security proofs and improvements* In: Advances in Cryptology - EUROCRYPT 2000, LNCS, vol. 1807, Springer-Verlag, pp 259-274, 2000.

16. Mu Y., Varadharajan V.: *Distributed signcryption.* In Roy, B., Okamoto, E. (eds.) INDOCRYPT 2000. LNCS, vol. 1977, pp. 155-164. Springer, Heidelberg (2000)
17. 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.
18. Shanshan Duan, Zhenfu Cao *Efficient and Secure Multi-Receiver Signcryption Scheme* In: http://tdt.sjtu.edu.cn/ dss/, 2006.
19. 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)
20. 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).
21. 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.
22. Zheng Y.: *Digital signcryption or How to achieve cost(signature & Encrytpion) ≪ cost(signature) + cost(encryption).* In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165-179. Springer, Heidelberg 1997.
23. Jung H.Y., Lee D.H., Lim J.I., and Chang K.S.: *Signcryption schemes with forward secrecy.* In: The Second Workshop on Information Security Application - WISA 2001, pp. 463-475, Seoul, Korea, 2001.

Appendix-A

Proof of Correctness of I-MIBSC In this section, we show that our improved scheme is consistent. If $\sigma = \langle y, U, W, T_i \rangle$ is a valid signcryption for a user with identity ID_i, then ***Unsigncrypt***$(\sigma, ID_A, ID_i, D_i)$ 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 $\omega' \stackrel{?}{=} \hat{e}(Z', P)$ and $\omega' \stackrel{?}{=} \hat{e}(X + h_2' Q_A, P_{pub})$ 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***

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

Appendix-B

Proof of Confidentiality of I-MIBSC

Theorem 4. *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 $\alpha = \hat{e}(P,P)^{abc}$ 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 $\mathcal{O}_{Unsigncrypt}$. 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 unsigncryption 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 *invalid*.
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 $\mathcal{O}_{\mathbf{Signcrypt}}(\mathbf{m}, \mathbf{ID_A}, \mathcal{L}_1)$***.*** 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 $\mathcal{O}_{KeyExtract}(ID_A)$, 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) $y = \mathcal{O}_{H_3}(\omega) \oplus (m \| Z \| X)$
   (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 $\sigma = \langle y, U, W, T_1, T_2, \ldots, T_t, \mathcal{L}_1 \rangle$.

d

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

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

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) $y^* = \mathcal{O}_{H_3}(\omega) \oplus (m_b \| Z^* \| X^*)$
   (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 $\sigma^* = \langle y^*, U^*, W^*, T_1, T_2, \ldots, T_t, \mathcal{L}^* \rangle$ to the adversary.

$\mathcal{A}$ can perform queries as above. However, it cannot query the unsigncryption 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. □