Breaking and Fixing of an Identity Based Multi-Signcryption Scheme

S. Sharmila Deva Selvi, S. Sree Vivek * and C. Pandu Rangan*

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

Abstract. Signcryption is a cryptographic primitive that provides authentication and confidentiality simultaneously in a single logical step. It is often required that multiple senders have to signcrypt a single message to a certain receiver. Obviously, it is inefficient to signcrypt the messages separately. An efficient alternative is to go for multi-signcryption. The concept of multi-signcryption is similar to that of multi-signatures with the added property - confidentiality. Recently, Jianhong et al. proposed an identity based multi-signcryption scheme. They claimed that their scheme is secure against adaptive chosen ciphertext attack and it is existentially unforgeable. In this paper, we show that their scheme is not secure against chosen plaintext attack and is existentially forgeable, we also provide a fix for the scheme and prove formally that the improved scheme is secure against both adaptive chosen ciphertext attack and existential forgery.

Keywords. Identity Based Cryptography, Signcryption, Cryptanalysis, Multi-Signcryption, Bilinear Pairing, Provable Security , Random Oracle Model.

1 Introduction

Secure message transmission over an insecure channel like internet requires both confidentiality and authenticity. An encryption scheme is used to achieve confidentiality and digital signature is used to achieve unforgeability. Digital signcryption scheme is a cryptographic primitive that achieves both these properties together in an efficient way. In 1997, Zheng [17] proposed the first digital signcryption scheme with the aforementioned properties, lower computational cost and communication overhead than signing then encrypting (StE) or encrypting then signing (EtS) the message (i.e, signing and encrypting the same message independently). Since then, many signcryption schemes were proposed. Baek et al. [2] gave the formal security model for digital signcryption schemes and provided the security proof for Zheng's [17] scheme in the random oracle model. In 1984, Shamir [14] introduced the concept of identity based cryptography and proposed the first identity based signature scheme. The idea of identity based cryptography is to enable an user to use any arbitrary string that uniquely identifies him as his public key. Identity based cryptography serves as an efficient alternative to Public Key Infrastructure (PKI) based systems.

By combining identity based cryptography and signcryption, Malone-Lee [10] proposed the first identity based signcryption scheme. But Libert et al. [9] pointed out that Malone-Lee's scheme [10] is not semantically secure, since the signature of the message is visible in the signcryption. Chow et al. [6] proposed another identity based signcryption scheme that provides both public verifiability and forward security. Boyen [5] proposed yet another identity based signcryption scheme with ciphertext anonymity in the random oracle model. Following that, Libert et al. [8] modified Boyen's security model to a PKI based signcryption scheme and proposed a scheme as well. They claimed that their signcryption scheme was semantically secure against adaptive chosen ciphertext attacks, ciphertext anonymity and key invisibility. However, Tan [15] showed that the scheme by Libert et al. [8] did not satisfy the above properties. Till date, the most efficient identity based signcryption scheme was the one proposed by Barreto et al. [3].

* 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

Multi-signatures allow multiple signers to jointly authenticate a message using a single compact signature. Gentry et al [7] were the first to propose a provably secure identity based multi-signature scheme based on bilinear pairing. Following that, Bellare et al. [4] gave a provably secure identity based multi-signature scheme based on RSA. Multi-receiver signcryption scheme provides an efficient way to signcrypt a single message to an ad-hoc group of receivers chosen by the sender. In contrast to multi-receiver signcryption schemes, multi-signcryption schemes can be employed when multiple senders want to signcrypt a single message to a single receiver. Informally, a multi-signcryption scheme can be considered as a multi-signature scheme with message confidentiality. Jianhong et al. [16] were the first to propose an identity based multi-signcryption scheme. There are a few multi-signcryption schemes available in the literature [12, 11, 13] but to the best of our knowledge Jianhong et al.'s [16] scheme is the only identity based multi-signcryption scheme available to date.

Multi-signcryption schemes can be directly used in many applications, such as E-Business - for a joint signing of confidential contracts between two or more organizations, or E-Government - to signcrypt an electronic legal document by a number of higher authorities, or in membership - for access right authentication of confidential resources. In all these applications if the signcryptions are performed individually by the group of senders, the cost involved in unsigncryption is high. Multi-signcryption enables to perform unsigncryption with the cost equal to a single unsigncryption irrespective of the number of senders.

Our Contribution. In this paper, we point out that Jianhong et al.'s [16] scheme is insecure with respect to chosen plaintext attack and is existentially forgeable. We also propose an improvement to Jianhong et al.'s [16] identity based multi-signcryption scheme and formally prove the confidentiality of the scheme against adaptive chosen ciphertext and identity attack. We give a new security model for unforgeability, which is adapted from Bellare et al.'s security model for identity based multi-signature schemes. Thus our scheme turns out to be the first provably secure identity based multi-signcryption scheme. We also give the complexity figure of the improved scheme.

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 $P, Q, R \in \mathbb{G}_1$,
  - $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$
  - $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$
  - $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ [Where $a, b \in_R \mathbb{Z}_q^*$]
- ***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.

Computation Diffie-Hellman Problem (CDHP)

Definition 1. *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 Computational Bilinear Diffie-Hellman Problem (CBDHP)

Definition 2. *Given* $(P, aP, bP, cP) \in \mathbb{G}_1^4$ *for unknown* $a, b, c \in \mathbb{Z}_q^*$ *, the CBDHP in* $\mathbb{G}_1$ *is to compute* $\hat{e}(P, P)^{abc} \in \mathbb{G}_2$.

The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ *in solving the CBDHP in* $\mathbb{G}_1$ *is defined as*

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

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

2.4 Identity-Based Multi-Signcryption

A generic Identity-Based Multi-Signcryption scheme for signcrypting a single message from n senders to a single recipient consists of the following probabilistic polynomial time algorithms:

- ***Setup***(κ)***.*** Given a security parameter κ, the Private Key Generator (PKG) generates the public parameters *params* and master secret key *msk* of the system.
- ***Key Extract***(ID_i)***.*** Given an identity ID_i of an user, the PKG computes the corresponding private key S_i and transmits it to owner in a secure way.
- ***Signcryption***$(m, \mathcal{L} = \{ID_1, \ldots, ID_n\}, ID_X, S_1, \ldots, S_n)$***.*** To signcrypt a message m to the receiver ID_X, the set of senders $\mathcal{L}$ with identity $ID_1, \ldots, ID_n$ and private key $S_1, \ldots, S_n$ run this algorithm to obtain the signcryption σ.
- ***Unsigncryption***$(\sigma, \mathcal{L} = \{ID_1, \ldots, ID_n\}, ID_X, S_X)$***.*** When the receiver X with identity ID_X and private key S_X receives the signcrypted ciphertext σ from a set of senders $\mathcal{L}$ with identities $\mathcal{L} = \{ID_1, \ldots, ID_n\}$, X executes this algorithm to obtain either the plain text m or a message *invalid* with respect to the validity of σ.

2.5 Security Model for Identity-Based Multi-Signcryption

The notion of semantic security of public key encryption was extended to identity-based signcryption scheme by Malone-Lee in [10]. This was later modified by Sherman et al. in [6] which incorporates indistinguishability against adaptive chosen ciphertext and identity attacks and existential unforgeability against adaptive chosen message and identity attacks. We describe below the security models for *confidentiality* of an identity-based multi-signcryption scheme against adaptively chosen ciphertext and identity attack, and provide a new model for *unforgeability* based on unforgeability model of Bellare et al.'s [4] for identity based multi-signature scheme. These are the strongest security notions for this problem. It is to be noted that Jianhong et al. [16] have proved confidentiality in the chosen identity model where as we prove our scheme in the adaptive chosen identity model which is considered to be stronger than chosen identity model [5].

Confidentiality: An identity based multi-signcryption scheme is indistinguishable against adaptive chosen ciphertext attacks (IND-CCA2), if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage in the following game between the challenger $\mathcal{C}$ and $\mathcal{A}$:

Setup: The challenger $\mathcal{C}$ runs this algorithm to generate the master public and private keys, *params* and *msk* respectively. $\mathcal{C}$ gives *params* to $\mathcal{A}$ and keeps the master private key *msk* secret from $\mathcal{A}$.

Phase 1: $\mathcal{A}$ performs a series of queries in an adaptive fashion in this phase. The queries allowed are given below:

Key Extract queries: $\mathcal{A}$ chooses an identity ID_i and gives it to $\mathcal{C}$. $\mathcal{C}$ computes the corresponding private key S_i and sends it to $\mathcal{A}$.

Signcryption queries: $\mathcal{A}$ produces a list of senders $\mathcal{L} = \{ID_1, \ldots, ID_n\}$, the recipient identity ID_X and a message m to $\mathcal{C}$. $\mathcal{C}$ computes the signcryption σ of the message m with $\mathcal{L}$ as the group of senders, ID_X as the receiver and sends σ to $\mathcal{A}$.

Unsigncryption queries: Given a ciphertext $\sigma = Signcryption(m, \mathcal{L} = \{ID_1, \ldots, ID_n\}, ID_X, S_1, \ldots, S_n)$,

$\mathcal{C}$ unsigncrypts σ with the receiver private key S_X, the list of sender identities $\mathcal{L}$ and returns the corresponding message to $\mathcal{A}$ if σ is a valid signcryption from the list if senders $\mathcal{L}$ to the receiver ID_X, else, $\mathcal{C}$ returns "*invalid*".

Challenge: At the end of *Phase 1*, $\mathcal{A}$ sends to $\mathcal{C}$, plaintexts m_0 and m_1 of equal length, a receiver identity ID_X and a sender list $\mathcal{L} = \{ID_1, \ldots, ID_n\}$. Here, $\mathcal{A}$ should have obtained the public key corresponding to the user in the sender list $\mathcal{L}$ and the receiver ID_X from $\mathcal{C}$. However, $\mathcal{A}$ should not have queried the private key of ID_X to the key extract oracle. Now, $\mathcal{C}$ chooses $b \in_R \{0, 1\}$ and computes the challenge signcryption σ^* on the message m_b with $\mathcal{L}$ as the group of senders, ID_X as the receiver and returns σ^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ can perform polynomially bounded number of queries adaptively again as in ***Phase 1*** but $\mathcal{A}$ cannot ask the key extraction query on the receiver identity ID_X or cannot ask the unsigncryption query on the challenge signcryption (σ^*, ID_X).

Guess: $\mathcal{A}$ outputs a bit b' and wins the game if $b' = b$.

The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}} = |2Pr[b' = b] - 1|$, where $Pr[b' = b]$ denotes the probability that $b' = b$.

Existential Unforgeability: Jianhong et al. [16] did not provide the formal security model for unforgeability of identity based multi-signcryption scheme but they claim that, a slightly altered security model of Malone-Lee et al. [10] has been adapted for their scheme. Bellare et al., in [4] have given two security notions for unforgeability of identity based multi-signature schemes, namely, single-signer security and multi-signer security, and have proved that single-signer security implies multi-signer security [4]. We extend the notion of single-signer security of Bellare et al. to prove the unforgeability of our improved identity based multi-signcryption scheme.

An identity based multi-signcryption scheme is claimed to be existentially unforgeable under chosen message attack (EUF-CMA), if any polynomially bounded forger $\mathcal{F}$ has a negligible advantage in the following game:

Setup: The challenger $\mathcal{C}$ runs the ***Setup*** algorithm to generate the master public and private keys, *params* and *msk* respectively. $\mathcal{C}$ gives *params* to $\mathcal{F}$ and keeps the master private key *msk* secret from $\mathcal{F}$.

Training Phase: $\mathcal{F}$ asks polynomial number of queries to the various oracles provided by $\mathcal{C}$, without any restrictions.

Forgery: At the end of the ***Training Phase***, $\mathcal{F}$ chooses a message m and produces a signcryption σ^* on m with $\mathcal{L} = \{ID_1, \ldots, ID_n\}$ as the list of sender identities and ID_X as the receiver identity. $\mathcal{F}$ wins the game if the private keys of at least one of the user in the list of senders $\mathcal{L}$ was not queried during ***Training Phase***, σ^* is a valid signcryption and σ^* is not the output of any previous queries to the *Signcryption Oracle* with m as the message, $\mathcal{L}$ as the sender list and ID_X as the receiver. (note that the private key S_X of the receiver ID_X can be queried by $\mathcal{F}$ during the ***Training Phase***.)

3 Review and Attack of Jianhong et al.'s Identity Based Multi-Signcryption Scheme (JJ-IBMSC)

In this section we review the identity based multi-signcryption scheme of Jianhong et al.'s (JJ-IBMSC) presented in [16]. We also show that [16] does not provide both confidentiality and unforgeability.

3.1 Review of the scheme

This scheme has the following four algorithms.

1. **Setup:** Given κ the security parameter and l the length of the message, the PKG chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^\kappa$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a random generator P of $\mathbb{G}_1$. It then chooses a master private key $s \in_R \mathbb{Z}_q^*$, a system-wide public key $P_{pub} = sP$ and three cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^l$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. The public parameters *Params*= $\langle \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, H_1, H_2, H_3, P_{pub}, \kappa \rangle$.

2. **Key Extract:** Given an identity $ID_i \in \{0,1\}^*$, the PKG does the following to extract the private key corresponding to ID_i:
   - Computes $Q_i = H_1(ID_i) \in \mathbb{G}_1$ and
   - Computes the private key $S_i = sQ_i$.
3. **Signcryption:** Given a message m, a receiver identity ID_X and a list of n sender identities $\mathcal{L} = \{ID_1, \dots, ID_n\}$ Each sender i executes the following steps:
   - Chooses $x_i \in_R \mathbb{Z}q^*$ and computes $R_i = x_i P$.
   - Computes $\omega_i = \hat{e}(P_{pub}, Q_X)^{x_i}$.
   - Sends $\langle R_i, \omega_i \rangle$ to the other senders through a secure channel.
   - After receiving all other senders $\langle R_i, \omega_i \rangle$ values, each sender computes $\omega = \prod_{j=1}^{n} \omega_j$, $c = H_2(\omega) \oplus m$, $R = \sum_{j=1}^{n} R_j$ and $T_i = x_i H_1(m) + H_3(R) S_i$.

   The resultant signcryption is $\sigma = \langle c, T, R, \mathcal{L} \rangle$, where $T = \sum_{j=1}^{n} T_j$.
4. **Unsigncryption:** Given $\sigma = \langle c, T, R, \mathcal{L} \rangle$, the receiver with identity ID_X does the following:
   - Computes $\omega' = \hat{e}(R, S_X)$.
   - Retrieves the message $m = H_2(\omega') \oplus c$.
   - Accepts the message if $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(R, H_1(m))\hat{e}(P_{pub}, \sum_{j=1}^{n} Q_j)^{H_3(R)}$

3.2 Attacks on Jianhong et al.'s Identity Based Multi-Signcryption Scheme (JJ-IBMSC)

We launch two different attacks on the scheme to show both the weakness in confidentiality as well as unforgeability.

Attack on Confidentiality: During the confidentiality game the adversary $\mathcal{A}$ interacts with the challenger $\mathcal{C}$ in ***Phase 1*** by performing various queries on *Hash, Key Extraction, Signcryption* and *Unsigncryption* oracles. At the end of ***Phase 1***, $\mathcal{A}$ choose two messages m_0 and m_1, a list of sender identities $\mathcal{L} = \{ID_1, \dots, ID_n\}$ and a targeted receiver identity ID_X on which $\mathcal{A}$ wants to be challenged and sends them to $\mathcal{C}$ (note that, $\mathcal{A}$ should not have queried the private key for the targeted receiver identity ID_X throughout the game but is allowed to query the private keys corresponding to all the senders in $\mathcal{L}$). Now, $\mathcal{C}$ chooses a bit $b \in_R \{0,1\}$ and generates a challenge signcryption $\sigma^* = \langle c^*, T^*, R^* \rangle, \mathcal{L}$ and sends σ^* to $\mathcal{A}$.

We now show that up on receiving the challenge signcryption σ^*, $\mathcal{A}$ can perform the following to check whether σ^* is a valid signcryption of message m_0 or m_1.

- Computes $\mathcal{H}_0 = H_1(m_0)$.
- Checks whether $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(R, \mathcal{H}_0)\hat{e}(P_{pub}, \Sigma_{j=1}^{n} Q_j)^{H_3(R)}$.
- If the above check passes then σ^* is the signcryption of m_0, otherwise σ^* is the signcryption of the message m_1.

This shows that the scheme is not even secure against chosen plaintext attack because on receiving the challenge signcryption σ^*, $\mathcal{A}$ is capable of distinguishing whether σ^* is a signcryption of m_0 or m_1, without any further interaction with $\mathcal{C}$.

Remark 1: This attack is possible because $\mathcal{A}$ knows the messages m_0 and m_1 during the confidentiality game and during unsigncryption, the message is directly used for the verification test $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(R, H_1(m))\hat{e}(P_{pub}, \Sigma_{j=1}^{n} Q_j)^{H_3(R)}$. Therefore, $\mathcal{A}$ can check the validity of σ^* with respect to both the messages and find out for which message the equation holds.

Attack on Unforgeability: The forger $\mathcal{F}$ aims to generate the signcryption of a message m on a list of senders $\mathcal{L}^*$ to a receiver ID_Y. Here the forger does not know the private key of any of the identities in $\mathcal{L}^*$. In order to achieve this, $\mathcal{F}$ first selects a set of identities $\mathcal{L}' = \{ID_1, \dots, ID_{k'}\}$, where $\mathcal{F}$ knows the private key corresponding to all the identities in $\mathcal{L}'$. $\mathcal{F}$ then sets $\mathcal{L} = \mathcal{L}^* \cup \mathcal{L}'$. Now, $\mathcal{F}$ sends a message m, the list $\mathcal{L}$ and an identity ID_X (where $ID_X \neq ID_Y$) to the challenger $\mathcal{C}$ and obtains a signcryption $\sigma = \langle c, T, R, \mathcal{L} \rangle$ from $\mathcal{C}$ (Note that it is legal for $\mathcal{F}$ to ask this signcryption from $\mathcal{C}$). $\mathcal{F}$ constructs $\sigma^* = Signcryption(m, \mathcal{L}' = \{ID_1, \dots, ID_{k'}\}, ID_Y, S_1, \dots, S_{k'})$ as follows:

– As mentioned earlier let $\sigma = \langle c, T, R, \mathcal{L}\rangle$ be the signcryption on m from $\mathcal{L}$ to ID_X.
– $\mathcal{F}$ queries the private key S_Y, corresponding to ID_Y (Since the receivers private key is available in the unforgeability game).
– $\mathcal{F}$ takes the value R from σ and computes $\omega^* = \hat{e}(R, S_Y)$ and computes a new value $c^* = m \oplus H_2(\omega^*)$.
– Computes $T^* = T - H_3(R)\,(\Sigma_{ID_i \in \mathcal{L}'} S_i)$ and sets $R^* = R$
– Now, $\mathcal{F}$ produces $\sigma^* = \langle c^*, T^*, R^*, \mathcal{L}^*\rangle$ as a valid signcryption on the same message m as σ with $\mathcal{L}^*$ as the list of senders and ID_Y as the receiver to the challenger.

We show that the new signcryption $\sigma^* = \langle c^*, T^*, R^*, \mathcal{L}^*\rangle$ is valid because it passes the verification with respect to the private key of the receiver ID_Y.

– Compute $\omega' = \hat{e}(R^*, S_Y)$=$\hat{e}(R, S_Y)$.
– Retrieve the message $m' = H_2(\omega') \oplus c^*$.
– The message m' obtained during unsigncryption is a valid message because $\hat{e}(T^*, P) = \hat{e}(R^*, H_1(m'))$ $\hat{e}(P_{pub}, \sum_{j=1}^{n} Q_j)^{H_3(R^*)}$.

We show the correctness of the above check with respect to the forged signcryption.

$$
\begin{aligned}
\hat{e}(T^*, P) &= \hat{e}(\textstyle\sum_{i=1}^{n} T_i, P)\\
&= \hat{e}(\textstyle\sum_{i=1}^{n}\{x_i H_1(m') + H_3(R^*) S_i\}, P)\\
&= \hat{e}(\textstyle\sum_{i=1}^{n}\{x_i\} H_1(m'), P)\hat{e}(\sum_{i=1}^{n}\{H_3(R^*) s Q_i\}, P)\\
&= \hat{e}(H_1(m'), \textstyle\sum_{i=1}^{n}\{x_i P\})\hat{e}(H_3(R^*)\sum_{i=1}^{n}\{Q_i\}, sP)\\
&= \hat{e}(H_1(m'), R)\hat{e}(\textstyle\sum_{i=1}^{n}\{Q_i\}, P_{pub})^{H_3(R^*)}
\end{aligned}
$$

Remark 2: Informally, this forgery is possible because there is no binding between the signature component T_i and the receiver identity ID_X and the other senders in list $\mathcal{L}$ involved in the generation of signcryption σ. The component T_i acts as a signature because it is computed with the sender's private keys. So, the forger $\mathcal{F}$ can alter the receiver and generate the new signcryption σ' by changing the component c, the senders list and the receiver. The component c' can be computed using the private key of the new receiver, which is known to $\mathcal{F}$. (Note: It is well known that during unforgeability game for signcryption, $\mathcal{F}$ has access to the receiver's private key in order to ensure insider security.)

4 Improved Identity Based Multi-Signcryption Scheme (I-IBMSC)

The scheme by Jianhong et al. [16] can be fixed by modifying it suitably so that it satisfies the necessary conditions outlined by An, Dodis and Rabin [1]. In this section, we present a possible fix for [16]. We also prove the indistinguishability against adaptive chosen ciphertext attack (IND-CCA2) for confidentiality and existential unforgeablity against chosen message attack (EUF-CMA) for unforgeability of I-IBMSC in the random oracle model. These are the strongest security notions for any identity based multi-signcryption scheme. Our modifications are simple and subtle and the formal proofs are non-trivial.

4.1 The Scheme (I-IBMSC)

The improved scheme has the following four algorithms as in JJ-IBMSC [16].

1. **Setup:** This algorithm is similar to that of the **Setup** algorithm in JJ-IBMSC with a new definition for the hash function H_2. We define $H_2 : \{0,1\}^* \rightarrow \{0,1\}^l$ in I-IBMSC. We also add one more hash function H_4, which is defined as $H_4 : \{0,1\}^* \rightarrow \mathbb{G}_1$.
2. **Key Extract:** This algorithm is identical to that of the **Key Extract** algorithm in JJ-IBMSC.
3. **Signcryption:** Given a message m, a receiver identity ID_X and a list of n sender identities $\mathcal{L} = \{ID_1, \ldots, ID_n\}$, each sender indexed i, for $i = 1$ to n executes the following steps:
   – Chooses $x_i \in \mathbb{Z}q^*$ at random and computes $R_i = x_i P$.
   – Computes $\omega_i = \hat{e}(P_{pub}, Q_X)^{x_i}$.
   – Sends $\langle R_i, \omega_i\rangle$ to all other senders in the list $\mathcal{L}$, through a secure channel.

   After receiving all other senders $\langle R_i, \omega_i\rangle$ values, each sender continues to do the following;
   – Computes $\omega = \prod_{j=1}^{n} \omega_j$ and $h_2 = H_2(\omega, \mathcal{L}, ID_X)$.

– Sets $c = h_2 \oplus m$, computes $R = \sum_{j=1}^{n} R_j$, $M = H_4(m, \omega, R, \mathcal{L}, ID_X)$ and $h_3 = H_3(R, \mathcal{L}, ID_X)$.
– Computes $T_i = x_i M + h_3 S_i$.

Now, each sender ID_i sends their corresponding T_i, c and R values to a clerk (Note that one of the senders in the list $\mathcal{L}$ may act as clerk), who checks whether c and R values sent by all senders are identical; if so, the clerk computes $T = \sum_{j=1}^{n} T_j$ and outputs the resultant signcryption as $\sigma = \langle c, T, R, \mathcal{L} \rangle$.

4. **Unsigncryption:** Given $\sigma = \langle c, T, R, \mathcal{L} \rangle$, the receiver with identity ID_X does the following to unsigncrypt it:
   – Computes $\omega' = \hat{e}(R, S_X)$.
   – Computes $h_2' = H_2(\omega', \mathcal{L}, ID_X)$ and retrieves the message $m' = h_2' \oplus c$.
   – Computes $M' = H_4(m', \omega', R, \mathcal{L}, ID_X)$ and $h_3' = H_3(R, \mathcal{L}, ID_X)$.
   – Accepts the message m' if $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(R, M')\hat{e}(P_{pub}, \sum_{j=1}^{n}\{Q_j\})^{h_3'}$.

Correctness: In order to prove the correctness of I-IBMSC we prove the consistency of the check $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(R, M')\hat{e}(P_{pub}, \sum_{j=1}^{n} Q_j)^{h_3'}$ below:

$$\begin{aligned}
\hat{e}(T, P) &= \hat{e}(\textstyle\sum_{j=1}^{n} T_j, P) \\
&= \hat{e}(\textstyle\sum_{j=1}^{n}\{x_j M' + h_3' S_j\}, P) \\
&= \hat{e}(\textstyle\sum_{j=1}^{n}\{x_j\} M', P)\hat{e}(\sum_{j=1}^{n}\{h_3' s Q_j\}, P) \\
&= \hat{e}(M', \textstyle\sum_{j=1}^{n}\{x_j P\})\hat{e}(h_3' \sum_{j=1}^{n}\{Q_j\}, sP) \\
&= \hat{e}(M', R)\hat{e}(\textstyle\sum_{j=1}^{n}\{Q_j\}, P_{pub})^{h_3'}
\end{aligned}$$

Note that the acceptance test is same as that of Jianhong et al.'s [16] but the change in the definition of the hash functions H_2, H_3 and the introduction of the new hash function H_4 provide the needed security for our system.

4.2 Security of the Scheme (I-IBMSC)

We prove the security of I-IBMSC with respect to confidentiality (IND-I-IBMSC-CCA2) and unforgeability (EUF-I-IBMSC-CMA) in this section.

Confidentiality:

Theorem 1. *If an IND-I-IBMSC-CCA2 adversary $\mathcal{A}$ has an advantage ϵ against I-IBMSC scheme, asking q_{H_i} (i = 1, 2, 3, 4) hash queries to random oracles $\mathcal{O}_{H_i}$ (i = 1, 2, 3, 4), q_e extract queries ($q_e = q_{e_1} + q_{e_2}$, where q_{e_1} and q_{e_2} are the number of extract queries in the first phase and second phase respectively), q_{sc} signcryption queries and q_{us} unsigncryption queries, then there exist an algorithm $\mathcal{C}$ that solves the CBDH problem with advantage $\epsilon\left(\frac{1}{q_{H_1} q_{H_2}}\right)$.*

Proof: Let $\mathcal{C}$ be a challenger, who is challenged with an instance of CBDHP say, $(P, aP, bP, cP) \in \mathbb{G}_1^4$ for unknown $a, b, c \in \mathbb{Z}_q^*$. The aim of $\mathcal{C}$ is to compute $\alpha = \hat{e}(P, P)^{abc}$. Consider an adversary $\mathcal{A}$ who is capable of breaking the IND-I-IBMSC-CCA2 security of I-IBMSC. $\mathcal{C}$ can make use of $\mathcal{A}$ to solve the CBDHP instance with non-negligible advantage in polynomial time as described below.

It is assumed that in the IND-I-IBMSC-CCA2 game $\mathcal{A}$ queries $\mathcal{O}_{H_1}$ oracle with ID before using ID as input for any oracle query.

Setup: $\mathcal{C}$ starts the confidentiality game by setting up the system with $Params = \langle \kappa, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, P_{pub} \rangle$, where $P_{pub} = aP$ and models the hash functions H_i's as random oracles $\mathcal{O}_{H_i}$, for $i = 1$ to 4 and sends the public parameters to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ performs a series of queries to the oracles provided by $\mathcal{C}$. In-order to maintain the consistency of the responses given by the oracles $\mathcal{O}_{H_i}$, $\mathcal{C}$ maintain lists L_i, for $i = 1$ to 4. The oracles provided by $\mathcal{C}$ and the responses given to the corresponding queries by $\mathcal{C}$ are described below:

Oracle $\mathcal{O}_{H_1}(ID_i)$*:* We will make a simplifying assumption that $\mathcal{A}$ queries the $\mathcal{O}_{H_1}$ oracle with distinct identities in each query. There is no loss of generality due to this assumption, because, if the same identity is repeated, by definition the oracle consults the list L_1 and gives the same response. Thus, we assume that $\mathcal{A}$ asks q_{H_1} distinct queries for q_{H_1} distinct identities. Among this q_{H_1} identities, a random identity has to be selected as target identity and it is done as follows.

$\mathcal{C}$ selects a random index γ, where $1 \leq \gamma \leq q_{H_1}$. $\mathcal{C}$ does not reveal γ to $\mathcal{A}$. When $\mathcal{A}$ generates the γ^{th} query on ID_γ, $\mathcal{C}$ fixes ID_γ as target identity for the challenge phase and $\mathcal{C}$ responds to $\mathcal{A}$ as follows:

- If $i = \gamma$, $\mathcal{C}$ sets $Q_\gamma = bP$, returns Q_γ as the response to the query and stores $\langle ID_\gamma, Q_\gamma, * \rangle$ in the list L_1. Here, $\mathcal{C}$ does not know b since $\mathcal{C}$ uses the bP value given in the instance of the CBDHP.
- For all other queries, $\mathcal{C}$ chooses $y_i \in_R Z_q^*$ and sets $Q_i = y_iP$ and stores $\langle ID_i, Q_i, y_i \rangle$ in the list L_1.

$\mathcal{C}$ returns Q_i to $\mathcal{A}$. (Note that as the identities are assumed to be distinct, for each query, we create distinct entry and add in the list L_1).

Oracle $\mathcal{O}_{H_2}(\omega, \mathcal{L}, ID_X)$*:* When $\mathcal{A}$ makes this query, $\mathcal{C}$ checks for the tuple $\langle \omega, \mathcal{L}, ID_X, h_2 \rangle$ in the list L_2. If an entry was found then returns the corresponding h_2 as the response else chooses $h_2 \in_R \{0,1\}^l$ and adds the tuple $\langle \omega, \mathcal{L}, ID_X, h_2 \rangle$ to the list L_2 and returns h_2 as the response to $\mathcal{A}$.

Oracle $\mathcal{O}_{H_3}(R, \mathcal{L}, ID_X)$*:* When $\mathcal{A}$ makes this query, $\mathcal{C}$ checks for the tuple $\langle R, \mathcal{L}, ID_X, h_3 \rangle$ in the list L_3. If an entry was found then returns the corresponding h_3 as the response else chooses $h_3 \in_R \mathbb{Z}_q^*$ and adds the tuple $\langle R, \mathcal{L}, ID_X, h_3 \rangle$ to the list L_3 and returns h_3 as the response to $\mathcal{A}$.

Oracle $\mathcal{O}_{H_4}(m, \omega, R, \mathcal{L}, ID_X)$*:* When $\mathcal{A}$ makes this query, $\mathcal{C}$ checks for the tuple $\langle m, \omega, R, \mathcal{L}, ID_X, z, M \rangle$ in the list L_4. If an entry was found then returns the corresponding M as the response else chooses $z \in \mathbb{Z}_q^*$ randomly and computes $M = zP_{pub}$. $\mathcal{C}$ now adds the tuple $\langle m, \omega, R, \mathcal{L}, ID_X, z, M \rangle$ to the list L_4 and returns M as the response to $\mathcal{A}$.

Oracle $\mathcal{O}_{KeyExtract}(ID_i)$*:* $\mathcal{A}$ chooses an identity ID_i and queries the corresponding private key S_i to this oracle (Note that $\mathcal{A}$ should have performed the $\mathcal{O}_{H_1}(ID_i)$ query before performing this query). $\mathcal{C}$ responds to this query as follows:

- If $ID_i = ID_\gamma$, then $\mathcal{C}$ aborts.
- Else, $\mathcal{C}$ retrieves y_i corresponding to ID_i from L_1, computes $S_i = y_i(aP) = a(y_iP) = aQ_i$ and returns S_i to $\mathcal{A}$.

Oracle $\mathcal{O}_{Signcryption}(m, \mathcal{L}, ID_X)$*:* Here, m is the message to be signcrypted, $\mathcal{L}$ is the list of identities of n senders and ID_X is the identity of the receiver. If $ID_\gamma \notin \mathcal{L}$, then $\mathcal{C}$ can generate the signcryption on the message m because $\mathcal{C}$ knows the private key of all senders belonging to $\mathcal{L}$.

- If $ID_\gamma \in \mathcal{L}$ then $\mathcal{C}$ performs the following:
  - Chooses $z, r, h_3 \in_R \mathbb{Z}_q^*$.
  - Computes $M = z(P_{pub})$, $R = \left(\frac{1}{z}\right)(rP - h_3 \sum_{ID_i \in \mathcal{L}} Q_i)$.
  - Stores the tuple $\langle R, \mathcal{L}, ID_X, h_3 \rangle$ to the list L_3.
  - Computes $T = rP_{pub}$.
  - Stores $\langle m, \omega, R, \mathcal{L}, ID_X, z, M \rangle$ in list L_4. (Note: If this tuple is already available in list L_4 then $\mathcal{C}$ repeats the above steps with different values of r and z.)
  - Compute $\omega = \hat{e}(R, S_X)$ and $c = m \oplus \mathcal{O}_{H_2}(\omega, \mathcal{L}, ID_X)$.

Now, $\mathcal{C}$ returns the signcryption on message m as $\sigma = \langle c, T, R, \mathcal{L} \rangle$ to $\mathcal{A}$.

We show that the signcryption $\sigma = \langle c, T, R, \mathcal{L} \rangle$ produced by $\mathcal{C}$ passes the validity test. The validity check done during unsigncryption is $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(R, M)\hat{e}(P_{pub}, \sum_{i=1}^n Q_i)^{h_3}$

$$
\begin{aligned}
\text{R.H.S} &= \hat{e}(R, M)\hat{e}(P_{pub}, \textstyle\sum_{i=1}^n Q_i)^{h_3} \\
&= \hat{e}(zP_{pub}, \frac{1}{z}(rP - h_3 \textstyle\sum_{ID_i \in \mathcal{L}} Q_i))\hat{e}(P_{pub}, \textstyle\sum_{i=1}^n Q_i)^{h_3} \\
&= \hat{e}(P_{Pub}, rP)\hat{e}(P_{pub}, \textstyle\sum_{i=1}^n Q_i)^{h_3} \\
&= \hat{e}(P_{Pub}, rP)\hat{e}(P_{pub}, -\textstyle\sum_{i=1}^n Q_i)^{h_3}\hat{e}(P_{pub}, \textstyle\sum_{i=1}^n Q_i)^{h_3} \\
&= \hat{e}(P_{Pub}, rP) \\
&= \hat{e}(rP_{pub}, P) \\
&= \hat{e}(T, P) \\
&= \text{L.H.S}
\end{aligned}
$$

It is clear that L.H.S=R.H.S and thus σ is a valid signcryption on message m with $\mathcal{L}$ as the sender list, $ID_\gamma \in \mathcal{L}$ and ID_X as the receiver.

Oracle $\mathcal{O}_{Unsigncryption}(\sigma, \mathcal{L}, ID_X)$*:* Here, σ is the signcryption, $\mathcal{L}$ is the list of identities of n senders and ID_X is the receiver identity. To respond to this query, $\mathcal{C}$ checks whether $ID_X \stackrel{?}{=} ID_\gamma$.

- If $ID_X \neq ID_\gamma$, $\mathcal{C}$ proceeds as per the normal unsigncryption algorithm, since $\mathcal{C}$ knows the private key of the receiver ID_X.
- If $ID_X = ID_\gamma$ then $\mathcal{C}$ computes $h_3 = \mathcal{O}_{H_3}(R, \mathcal{L}, ID_X)$ and performs the following to unsigncrypt σ:
  1. Let Δ be the set of pairs (ω, h_2) from the list L_2 corresponding to $(\mathcal{L}, ID_X) \in \sigma$. (The tuples in L_2 will be of the form $\langle \omega, \mathcal{L}, ID_X, h_2 \rangle$.)
  2. For each $(\omega, h_2) \in \Delta$, $\mathcal{C}$ performs the following,
     (a) Computes $m' = c \oplus h_2$.
     (b) Computes $M = \mathcal{O}_{H_4}(m', \omega, \mathcal{L}, ID_X)$.
     (c) Checks whether $\hat{e}(M, R)^{1/z} \stackrel{?}{=} \omega$ and $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(M, R)\hat{e}(P_{pub}, \sum_{ID_i \in \mathcal{L}} Q_i)^{h_3}$. (Note that z is retrieved from the tuples in the list L_4 corresponding to $(m', \omega, \mathcal{L}, ID_X)$, which is of the form $\langle m', \omega, \mathcal{L}, ID_X, z, M \rangle \in L_4$.)
     (d) If true, return m'.
  3. If the test in step 2(c) fails for all $(\omega, h_2) \in \Delta$, $\mathcal{C}$ returns "*invalid*".

Challenge: At the end of ***Phase 1***, $\mathcal{A}$ generates and sends to $\mathcal{C}$, two plaintexts m_0 and m_1 of equal length, a sender list $\mathcal{L} = \{ID_1, \ldots, ID_{\overline{n}}\}$ and the receiver identity ID_X. Note that $\mathcal{A}$ should not have queried the private key corresponding to ID_X in ***Phase 1***. $\mathcal{C}$ checks whether $ID_X = ID_\gamma$. If not, $\mathcal{C}$ aborts, else, $\mathcal{C}$ chooses a bit $b \in_R \{0, 1\}$ and computes the challenge signcryption σ^* on the message m_b with $\mathcal{L}$ as the group of senders and ID_X as the receiver by performing the following:

- $\mathcal{C}$ sets $R = cP$, chooses randomly $T \in \mathbb{G}_1$ and $c \in \{0, 1\}^l$

Now, $\mathcal{C}$ outputs the signcryption on the message m_b as $\sigma = \langle c, T, R, \mathcal{L} \rangle$.

Phase 2: $\mathcal{A}$ can perform polynomially bounded number of queries adaptively again as in *Phase 1* with the restriction that $\mathcal{A}$ cannot query

- The key extract oracle for the private key of ID_X.
- The unsigncryption oracle with the challenge signcryption σ^*, ID_X.

Guess: At the end of ***Phase 2***, $\mathcal{A}$ outputs a guess b' to $\mathcal{C}$. Similar to the argument in [5], $\mathcal{C}$ ignores the response by $\mathcal{A}$, picks a random ω from the list L_2 and returns it as the solution to the CBDHP instance. Since the challenge ciphertext σ^* given to $\mathcal{A}$ is randomly distributed in the ciphertext space, $\mathcal{A}$ cannot gain any advantage in the IND-I-IBMSC-CCA2 game. Thus, any adversary that has advantage ϵ in the real IND-I-IBMSC-CCA2 game must necessarily recognize with probability at least ϵ that the challenge ciphertext provided by $\mathcal{C}$ is incorrect. For $\mathcal{A}$ to find that σ^* is not a valid ciphertext, $\mathcal{A}$ should have queried the $\mathcal{O}_{H_2}$ oracle with $\omega = \hat{e}(R, S_\gamma)$ (Since the receiver in σ^* is $ID_X = ID_\gamma$). Here S_γ is the private key of the target identity and it is $a(Q_\gamma) = abP$. Also, $\mathcal{C}$ has set $R = cP$. Hence $\omega = \hat{e}(R, S_\gamma) = \hat{e}(cP, abP) = \hat{e}(P, P)^{abc}$. Therefore, one of the entries in the list L_2 should be the value $\hat{e}(P, P)^{abc}$. With probability $\frac{1}{q_{H_2}}$, the value of ω chosen by $\mathcal{C}$ from list L_2 will be the solution to CBDHP instance.

Now, we analyze the probability of success of $\mathcal{C}$. The events in which $\mathcal{C}$ aborts the IND-I-IBMSC-CCA2 game are,

1. E_1 - when $\mathcal{A}$ queries the private key of the target identity ID_γ and $Pr[E_1] = \dfrac{q_{e_1}}{q_{H_1}}$.
2. E_2 - when $\mathcal{A}$ does not choose the target identity ID_γ as the receiver during the challenge and $Pr[E_2] = \left(1 - \dfrac{1}{q_{H_1} - q_{e_1}}\right)$.

The probability that, $\mathcal{C}$ does not abort the IND-I-IBMSC-CCA2 game is given by

$$(Pr[\neg E_1 \wedge \neg E_2]) = \left(1 - \frac{q_{e_1}}{q_{H_1}}\right)\left(\frac{1}{q_{H_1} - q_{e_1}}\right) = \frac{1}{q_{H_1}}$$

The probability that, the ω chosen randomly from L_2 by $\mathcal{C}$, being the solution to CBDHP is $\left(\frac{1}{q_{H_2}}\right)$. Therefore, the probability of $\mathcal{C}$ solving CBDHP is given by,

$$Pr[\mathcal{C}(P, aP, bP, cP | a, b, c \in_R \mathbb{Z}_q^*) = \hat{e}(P,P)^{abc}] = \epsilon\left(\frac{1}{q_{H_1}q_{H_2}}\right)$$

Since ϵ is non-negligible, the probability of $\mathcal{C}$ solving CBDHP is also non-negligible. □

Unforgeability:

Theorem 2. *If an EUF-I-IBMSC-CMA forger $\mathcal{F}$ exists against I-IBMSC scheme, asking q_{H_i} (i = 1, 2, 3, 4) hash queries to random oracles H_i (i = 1, 2, 3, 4), q_e extract secret key queries, q_{sc} signcryption queries and q_{us} unsigncryption queries, then there exist an algorithm $\mathcal{C}$ that solves the CDHP with advantage.*

Proof: Let $\mathcal{C}$ be a challenger who is challenged with an instance of CDH problem say, $(P, aP, bP) \in \mathbb{G}_1^3$ for unknown $a, b \in \mathbb{Z}_q^*$. The aim of $\mathcal{C}$ is to compute the value abP. Consider a forger $\mathcal{F}$ who is capable of breaking the EUF-I-IBMSC-CMA security of I-IBMSC. $\mathcal{C}$ can make use of $\mathcal{F}$ to solve the CDH problem instance with non-negligible advantage in polynomial time as described below.

Setup: $\mathcal{C}$ starts the unforgeability game by setting up the system with $P_{pub} = aP$, models the hash functions H_i's as random oracles $\mathcal{O}_{H_i}$, for $i = 1$ to 4 and sends the public parameters $Params = \langle \kappa, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, P_{pub}\rangle$ to $\mathcal{F}$. The elements of $Params$ are set identical to that of the CDH instance, $\mathcal{C}$ has received.

It is assumed in the EUF-I-IBMSC-CMA game that $\mathcal{A}$ queries $\mathcal{O}_{H_1}$ oracle with ID before using ID as input for any other query.

Training Phase: $\mathcal{F}$ performs a series of queries to the oracles provided by $\mathcal{C}$. In-order to maintain the consistency of the responses given by the oracles $\mathcal{O}_{H_i}$, $\mathcal{C}$ maintain lists L_i, for $i = 1$ to 4. The oracles $\mathcal{O}_{H_i}, i = 1, 2, 3$ and $\mathcal{O}_{KeyExtract}$ provided by $\mathcal{C}$ and the responses given to the corresponding queries by $\mathcal{C}$ are identical to that of the IND-I-IBMSC-CCA2 game. The description of the other oracles are given below,

Oracle $\mathcal{O}_{H_4}(m, \omega, R, \mathcal{L}, ID_X)$*:* When $\mathcal{A}$ makes this query, $\mathcal{C}$ checks for the tuple $\langle m, \omega, R, \mathcal{L}, ID_X, z, M\rangle$ in the list L_4. If an entry was found then returns the corresponding M as the response else chooses $z \in \mathbb{Z}_q^*$ randomly and if $ID_X = ID_\gamma$, $\mathcal{C}$ computes $M = zP$ else computes $M = zP_{pub}$. $\mathcal{C}$ now adds the tuple $\langle m, \omega, R, \mathcal{L}, ID_X, z, M\rangle$ to the list L_4 and returns M as the response to $\mathcal{A}$.

Oracle $\mathcal{O}_{Signcryption}(m, \mathcal{L}, ID_X)$*:* Here, m is the message to be signcrypted, $\mathcal{L}$ is the list of identities of n senders and ID_X is the receiver identity (Note that the receiver identity can also be the target identity ID_X). Even in the case of $ID_X = ID_\gamma$, $\mathcal{C}$ can generate the signcryption on the message m because $\mathcal{C}$ knows the private key of all senders belonging to $\mathcal{L}$. In order to respond to this query, $\mathcal{C}$ checks for each $ID_i \in \mathcal{L}$, whether $ID_i = ID_\gamma$. If $ID_\gamma \notin \mathcal{L}$, $\mathcal{F}$ follows the signcrypt protocol as $\mathcal{C}$ knows the private key of all senders in $\mathcal{L}$. If $ID_\gamma \in \mathcal{L}$ then $\mathcal{C}$ performs the following:

- Chooses z, h_3 randomly from $\mathbb{Z}_q^*$
- Computes $M = zQ_\gamma$ and $R = \left(\dfrac{-h_3}{z}\right) P_{pub}$.
- Computes $\omega = \hat{e}(R, S_X)$ and $c = m \oplus \mathcal{O}_{H_2}(\omega, \mathcal{L}, ID_X)$.
- Computes $T = h_3(\sum_{ID_i \in \mathcal{L} \wedge ID_i \neq ID_\gamma} S_i)$.
- Stores $\langle m, \omega, R, \mathcal{L}, ID_X, z, M\rangle$ in list L_4. (Note: If this tuple is already available in list L_4 then $\mathcal{C}$ repeats the above steps with different values.)

Now, $\mathcal{C}$ stores $\langle m, \sigma = \langle c, T, R, \mathcal{L}\rangle, ID_X\rangle$ in list L_5 returns the signcryption on message m as σ to $\mathcal{F}$.

We now show that the signcryption $\sigma = \langle c, T, R, \mathcal{L}\rangle$ produced by $\mathcal{C}$ is valid because it passes the verification test $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(R, M)\hat{e}(P_{pub}, \sum_{i=1}^{n} Q_i)^{h_3}$ as shown below.

$$\begin{aligned}
\text{R.H.S} &= \hat{e}(R, M)\hat{e}(P_{pub}, \textstyle\sum_{ID_i \in L} Q_i)^{h_3} \\
&= \hat{e}(\left(\frac{-h_3}{z}\right) P_{pub}, zP)\hat{e}(P_{pub}, \textstyle\sum_{ID_i \in L} Q_i)^{h_3} \\
&= \hat{e}(P_{pub}, \textstyle\sum_{ID_i \in L \wedge ID_i \neq ID_\gamma} Q_i)^{h_3} \\
&= \hat{e}(P, h_3 \textstyle\sum_{ID_i \in L \wedge ID_i \neq ID_\gamma} S_i) \\
&= \hat{e}(T, P) \\
&= \text{L.H.S}
\end{aligned}$$

It is clear that L.H.S=R.H.S and thus σ is a valid signcryption on message m with $\mathcal{L}$ as the sender list, $ID_\gamma \in \mathcal{L}$ and ID_X as the receiver.

Oracle $\mathcal{O}_{Unsigncryption}(\sigma, \mathcal{L}, ID_X)$*:* Here, σ is the signcryption, $\mathcal{L}$ is the list of identities of n senders and ID_X is the receiver identity. To respond to this query, $\mathcal{C}$ does the following.

- If $ID_X \neq ID_\gamma$ then $\mathcal{C}$ knows the private key of ID_X and can perform the unsigncryption as per the protocol.
- If $ID_X = ID_\gamma$, then $\mathcal{C}$ performs the following to unsigncrypt σ
- If $\sigma \in L_5$, then $\mathcal{C}$ returns m to $\mathcal{F}$,
- Otherwise, $\mathcal{C}$ does the following:
  1. Computes $h_3 = \mathcal{O}_{H_3}(R, L, ID_X)$.
  2. Let Δ be the set of pairs (ω, h_2) from the list L_2 corresponding to $(\mathcal{L}, ID_X) \in \sigma$. (The tuples in L_2 will be of the form $\langle \omega, \mathcal{L}, ID_X, h_2 \rangle$.)
  3. For each $(\omega, h_2) \in \Delta$, $\mathcal{C}$ performs the following,
     (a) Computes $m' = c \oplus h_2$.
     (b) Computes $M = \mathcal{O}_{H_4}(m', \omega, \mathcal{L}, ID_X)$.
     (c) Checks whether $\hat{e}(M, R)^{1/z} \stackrel{?}{=} \omega$ and $\hat{e}(T, P) \stackrel{?}{=} \hat{e}(M, R)\hat{e}(P_{pub}, \sum_{ID_i \in \mathcal{L}} Q_i)^{h_3}$. (Note that z is retrieved from the tuples in the list L_4 corresponding to $(m', \omega, \mathcal{L}, ID_X)$, which is of the form $\langle m', \omega, \mathcal{L}, ID_X, z, M \rangle \in L_4$).
     (d) If true, return m'.
  4. If the test in step 3(c) fails for all $(\omega, h_2) \in \Delta$, $\mathcal{C}$ returns "*invalid*".

Forgery: Eventually, $\mathcal{F}$ outputs a forged signcryption $\sigma^* = \langle c^*, T^*, R^*, \mathcal{L} \rangle$ on some message m^* with $\mathcal{L}$ as the list of senders and an arbitrary receiver, say ID_X. The restriction in generating σ^* is, $\mathcal{F}$ should not have generated σ^* by querying the signcryption oracle $\mathcal{O}_{Signcryption}$ in any previous queries on the message m^* with $\mathcal{L}$ as the list of senders and ID_X as the receiver. $\mathcal{C}$ can very well unsigncrypt and verify the validity of the forged signcryption σ^* because $\mathcal{C}$ knows the secret key of the receiver during the unforgeability game. If the forged signcryption σ^* passes the verification and at least one of the identities in $\mathcal{L}$ say ID_A was not the output of any previous key extract queries, then $\mathcal{C}$ can obtain the solution for the CDH problem instance by performing the following steps:

- $\mathcal{C}$ obtains ω^* and the message m^* during unsigncryption.
- It then checks the list L_4 for the tuple of the form $\langle m^*, \omega^*, R^*, \mathcal{L}, ID_X, z, M \rangle$.
- Computes $\sum_{i=1}^{n}\{x_i M\} = \sum_{i=1}^{n} x_i(zP) = z\sum_{i=1}^{n}(x_i)P = zR$.
- Computes $X = T - \sum_{i=1}^{n}\{x_i M\} = \sum_{i=1}^{n} h_3 S_i$.
- Computes $Y = \sum_{i=1}^{n}\{h_3 S_i\} - \sum_{i=1, i \neq A}^{n} h_3 S_i = h_3 S_A$. This is possible because $\mathcal{C}$ can retrieve the value h_3 from the list L_3 which corresponds to the tuple $\langle R, \mathcal{L}, ID_X, h_3 \rangle$ and $\mathcal{C}$ knows the private key corresponding to all the identities of the sender list $\mathcal{L}$ except ID_A.

  Therefore, $\mathcal{C}$ can compute:
  $$\begin{aligned}(h_3)^{-1}Y &= (h_3)^{-1}h_3 S_A \\ &= (h_3)^{-1}h_3 abP \\ &= abP\end{aligned}$$

Thus, $\mathcal{C}$ is capable of finding abP value which is the solution for the CDH problem instance. So, if there exists a forger who can forge a valid signcryption with non-negligible advantage, then there exists an algorithm to solve the CDH problem with non-negligible advantage. Since this is not possible, no forger can forge a valid signcryption with non-negligible advantage. Hence, I-IBMSC is secure against any EUF-I-IBMSC-CMA attack. Now we analyse the probability of success of $\mathcal{C}$ in solving the CDHP.

$$Pr[\mathcal{C}(aP, bP) = abP | a, b \in \mathbb{Z}_q^*] = Pr[\mathcal{F}(\sigma^*, ID_X) = Valid].Pr[\neg Abort]$$

The events in which $\mathcal{C}$ aborts are:

- E_1 - $\mathcal{F}$ queries the private key of the target identity to the key extract oracle and $Pr[E_1] = \left(\frac{q_e}{q_{H_1}}\right)$.
- E_2 - $\mathcal{F}$ doesnot choose ID_γ as sender in $\mathcal{L}$ which is used for forgery and $Pr[E_2] = \left(\frac{q_{H_1} - q_e - n}{q_{H_1} - q_e}\right) = \left(1 - \frac{n}{q_{H_1} - q_e}\right)$.

Therefore, the probability that $\mathcal{C}$ does not abort the game is given by

$$Pr[\neg Abort] = Pr[\neg E_1 \wedge \neg E_2]$$

$$= \left(1 - \frac{q_e}{q_{H_1}}\right)\left(1 - \left(1 - \frac{n}{q_{H_1} - q_e}\right)\right)$$

$$= \left(\frac{q_{H_1} - q_e}{q_{H_1}}\right)\left(\frac{n}{q_{H_1} - q_e}\right)$$

$$= \frac{n}{q_{H_1}}$$

Thus, the probability of success of $\mathcal{C}$ is given by $Pr[\mathcal{C}(aP, bP) = abP | a, b \in \mathbb{Z}_q^*] = \epsilon \dfrac{n}{q_{H_1}}$ $\square$

5 Conclusion

As the only existing identity based multi-signcryption scheme is cryptanalized for its confidentiality and unforgeability, we do not compare the efficiency of our scheme with any other scheme but we present the complexity figure of I-IBMSC scheme below:

Scheme	Signcrypt				Designcrypt			
	PA	SM	GE	MG	PA	SM	GE	MG
I-IBMSC	1/sender	3/sender	1/sender	1/sender	4	-	1	1

Table-1: Complexity figure for I-IBMSC

PA - Pairing, SM - Scalar Multiplication, GE - Exponentiation in $\mathbb{G}_2$, GM - Mapping to $\mathbb{G}_1$.

We have provided an improved identity based multi-signcryption scheme, which is an extension of Jianhong et al.'s scheme with the proper binding, that provides adequate security to the scheme. We have also proved the security of the improved scheme formally under the random oracle model.

References

1. Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer, 2002.
2. Joonsang Baek, Ron Steinfeld, and Yuliang Zheng. 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, 2002.
3. Paulo S. L. M. Barreto, Benoît Libert, Noel McCullagh, and Jean-Jacques Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Advances in Cryptology - ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2005.
4. Mihir Bellare and Gregory Neven. Identity-based multi-signatures from rsa. In *Topics in Cryptology - CT-RSA 2007*, volume 4377 of *Lecture Notes in Computer Science*, pages 145–162. Springer, 2007.
5. Xavier Boyen. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography). In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–399. Springer, 2003.
6. Sherman S. M. Chow, Siu-Ming Yiu, Lucas Chi Kwong Hui, and K. P. Chow. Efficient forward and provably secure id-based signcryption scheme with public verifiability and public ciphertext authenticity. In *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 352–369. Springer, 2003.
7. Craig Gentry and Zulfikar Ramzan. Identity-based aggregate signatures. In *Public Key Cryptography - PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 257–273. Springer, 2006.
8. Benoît Libert and Jean-Jacques Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. In *Public Key Cryptography - PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2004.
9. Benot Libert and Jean-Jacques Quisquater. A new identity based signcryption scheme from pairings. In *In IEEE Information Theory Workshop*, pages 155–158, 2003.

10. John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002.
11. Shirow Mitomi and Atsuko Miyaji. A multisignature scheme with message flexibility, order flexibility and order verifiability. In *Information Security and Privacy, ACISP - 2000*, volume 1841 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2000.
12. Xiaolin Pang, Barbara Catania, and Kian-Lee Tan. Securing your data in agent-based p2p systems. In *Eighth International Conference on Database Systems for Advanced Applications - DASFAA 2003*, pages 55–64. IEEE Computer Society, 2003.
13. Seung-Hyun Seo and Sang-Ho Lee. A secure and flexible multi-signcryption scheme. In *Computational Science and Its Applications, ICCSA - 2004*, volume 3046 of *Lecture Notes in Computer Science*, pages 689–697. Springer, 2004.
14. Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, CRYPTO - 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
15. Chik How Tan. On the security of signcryption scheme with key privacy. *IEICE Transactions*, volume 88-A(4):pages 1093–1095, 2005.
16. Jianhong Zhang and Jian Mao. A novel identity-based multi-signcryption scheme. *Computer Communications*, volume 32(1):pages 14–18, 2009.
17. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) $< <$ cost(signature) + cost(encryption). In *Advances in Cryptology, CRYPTO - 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 1997.