AN IDENTITY BASED RING SIGNCRYPTION SCHEME WITH PUBLIC VERIFIABILITY

S. Sharmila Deva Selvi, S. Sree Vivek, Sakhi S. Anand and C. Pandu Rangan
Theoretical Computer Science Lab, Indian Institute of Technology, 600036, Chennai, India

Keywords: Ring signcryption, Adaptive chosen ciphertext attack, Bilinear pairing, Random oracle model, Public verifiability.

Abstract: Signcryption is a cryptographic primitive which offers authentication and confidentiality simultaneously with a cost lower than signing and encrypting the message independently. Ring signcryption enables a user to anonymously signcrypt a message on behalf of a set of users including himself. Thus a ring signcrypted message has anonymity in addition to authentication and confidentiality. Ring signcryption schemes have no centralized coordination: any user can choose a ring of users, that includes himself and signcrypt any message without any assistance from the other group members. Ring Signcryption is useful for leaking trustworthy secrets in an anonymous, authenticated and confidential way.
To the best of our knowledge, ten identity based ring signcryption schemes are reported in the literature. Three of them were proved to be insecure in (Li et al., 2008a), (Zhang et al., 2009a) and (Vivek et al., 2009). Four of them were proved to be insecure in (Selvi et al., 2009). In this paper, we show that one among the remaining three schemes, (Zhang et al., 2009b) is not secure against confidentiality, existential unforgeability and anonymity attacks. We propose a new anonymous ring signcryption scheme which is an extension to (Selvi et al., 2009) and give formal security proofs for our system in the random oracle model. Our scheme is publicly verifiable which none of the existing unbroken schemes can achieve.

1 INTRODUCTION

Let us consider a scenario, where a member of the cabinet wants to leak a very important and juicy information, regarding the president of the nation to the press. He has to leak the secret in an anonymous way, else he will be black spotted in the cabinet. The press will not accept the information unless it is authenticated by one of the members of the cabinet. Here, if the information is so sensitive and should not be leaked until the authorities in the press receives it, we should have confidential transmission of information. Thus, we require anonymity to safeguard the cabinet member who sends the information, authentication for the authorities in the press to believe the information and confidentiality until the information reaches the hands of the right person in the press. All the three properties are together achieved by a single primitive called "Ring Signcryption". The first identity based ring signcryption scheme was proposed by Huang et al.(Huang et al., 2005). Subsequently, several schemes appeared in the literature((Zhang et al., 2008), (Li et al., 2008b), (Li et al., 2008a), (Yu et al., 2008), (Zhu et al., 2008), (Zhun and Zhang, 2008), (Huang et al., 2005), (Selvi et al., 2009), (Zhang et al., 2009a) and (Zhang et al., 2009b)). However, the weaknesses of (Zhang et al., 2008), (Li et al., 2008b), (Zhang et al., 2009a) were shown in (Li et al., 2008a), (Vivek et al., 2009) and (Zhang et al., 2009b) respectively. The insecurities of the schemes (Li et al., 2008a), (Yu et al., 2008), (Zhu et al., 2008) and (Zhun and Zhang, 2008) were shown in (Selvi et al., 2009). In this paper, we show that (Zhang et al., 2009b) is insecure against confidentiality, unforgeability and anonymity attacks.

Typically, signcryption σ is generated by a sender to a specific receiver and only the receiver can verify the validity of σ and recover the message from σ. However, in some practical scenarios, verification may have to be carried out by an entity other than the receiver but the verifier should not obtain the message. We call this property as public verifiability. For instance, firewalls are one of the most useful and versatile tools available for securing a LAN and other applications such as constructing secure private virtual networks. They are typically operated as a fil-

Sharmila Deva Selvi S., Sree Vivek S., S. Anand S. and Pandu Rangan C. (2010).
AN IDENTITY BASED RING SIGNCRYPTION SCHEME WITH PUBLIC VERIFIABILITY.
In *Proceedings of the International Conference on Security and Cryptography*, pages 362-371
DOI: 10.5220/0002981203620371

tering gateway at the LAN-WAN interface, usually a router. A signcryption scheme used in a LAN should satisfy the public verifiability property. This requires that any third party should be able to verify the origin of the signcryption without knowledge of the message and without getting any additional information from the intended recipient. Even, in the scenario mentioned above, a press authority may receive several ring signed messages and it is only appropriate that the filtering gateway is equipped with public verifying capabilities of the ring signcryptions.

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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following properties.

- **Bilinearity.** For all $P, Q, R \in_R \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$
  - $\hat{e}(P+Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$
  - $\hat{e}(P, Q+R) = \hat{e}(P, Q)\hat{e}(P, R)$
  - $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **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 Bilinear Diffie-Hellman Problem (CBDHP)

Given $(P, aP, bP, cP) \in \mathbb{G}_1^4$, for unknown $a, b, c \in \mathbb{Z}_q^*$, the CBDH problem 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 CBDH problem in $\mathbb{G}_1$ is defined as

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

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

2.3 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.

3 IDENTITY BASED RING SIGNCRYPTION

3.1 Framework

A generic identity based ring signcryption scheme consists of the following four algorithms.

Setup(κ). Given a security parameter κ, the private key generator (PKG) generates the systems public parameters *params* and the corresponding master private key *msk* that is kept secret by PKG.

Extract(ID_i). Given a user identity ID_i by user $\mathcal{U}_i$, the PKG computes the corresponding private key D_i and sends D_i to ID_i through a secure channel.

Signcrypt(m, $\mathcal{L}$, $ID_{\mathbb{S}}$, $D_{\mathbb{S}}$, $ID_{\mathbb{R}}$). This algorithm takes a message $m \in \mathcal{M}$, an ad-hoc group of ring members $\mathcal{L} = \{\mathcal{U}_1, \mathcal{U}_2, ...\mathcal{U}_n\}$ with identities $\{ID_1, \ldots, ID_n\}$, the sender identity $ID_{\mathbb{S}}$, the sender private key $D_{\mathbb{S}}$ and the receiver identity $ID_{\mathbb{R}}$ as input and outputs the ring signcryption C. This algorithm is executed by the sender with identity $ID_{\mathbb{S}} \in \mathcal{L}$. $ID_{\mathbb{R}}$ may or may not be in $\mathcal{L}$.

Unsigncrypt(C, $\mathcal{L}$, $ID_{\mathbb{R}}$, $D_{\mathbb{R}}$). This algorithm takes the ring signcryption C, the ring members $\mathcal{L} = \{\mathcal{U}_1, \mathcal{U}_2, ...\mathcal{U}_n\}$ and the private key $D_{\mathbb{R}}$ of the receiver $\mathcal{U}_{\mathbb{R}}$ with identity $ID_{\mathbb{R}}$ as input and produces the plaintext m, if C is a valid ring signcryption of m from the ring $\mathcal{L}$ to $ID_{\mathbb{R}}$ or "Invalid", if C is an invalid ring signcryption.

3.2 Security Notion

The formal security definition of signcryption was given by Baek et al.(Baek et al., 2002). The security of ID-based signcryption scheme was first defined by Malone-Lee (Malone-lee, 2002) that satisfies indistinguishability against adaptive chosen ciphertext attacks and unforgeability against adaptive chosen message attacks.

Definition 1 (Confidentiality). *An identity based ring signcryption (IRSC) is indistinguishable against*

adaptive chosen ciphertext attacks (IND-IRSC-CCA2) if there exists no polynomially bounded adversary having non-negligible advantage in the following game:

1. ***Setup Phase.*** *The challenger $\mathcal{C}$ runs the* ***Setup*** *algorithm with the security parameter κ as input and sends the system parameters* ***params*** *to the adversary $\mathcal{A}$ and keeps the master private key* ***msk*** *secret.*
2. ***Phase-I.*** *$\mathcal{A}$ performs polynomially bounded number of queries to the oracles provided to $\mathcal{A}$ by $\mathcal{C}$. The description of the queries in the phase-I are listed below:*

   Key Extraction Query. *$\mathcal{A}$ produces an identity ID_i corresponding to $\mathcal{U}_i$ and receives the private key D_i corresponding to ID_i.*

   Signcryption Query(m, $\mathcal{L}$, $ID_{\mathbb{S}}$, $ID_{\mathbb{R}}$). *$\mathcal{A}$ produces a message* ***m****, a sender group $\mathcal{L} = \{\mathcal{U}_i\}_{(i=1\ to\ n)}$, a sender identity $ID_{\mathbb{S}}$ and a receiver identity $ID_{\mathbb{R}}$ to the challenger $\mathcal{C}$. Then $\mathcal{C}$ signcrypts* ***m*** *from $ID_{\mathbb{S}}$ to $ID_{\mathbb{R}}$ with $D_{\mathbb{S}}$ and sends the result to $\mathcal{A}$.*

   Unsigncryption Query(C, $\mathcal{L}$, $ID_{\mathbb{R}}$). *$\mathcal{A}$ produces the sender group $\mathcal{L} = \{\mathcal{U}_{i,(i=1\ to\ n)}$, a receiver identity $ID_{\mathbb{R}}$, and a ring signcryption C. $\mathcal{C}$ generates the private key $D_{\mathbb{R}}$ by querying the* ***Key Extraction oracle****. $\mathcal{C}$ unsigncrypts C using $D_{\mathbb{R}}$ and returns* ***m*** *if C is a valid ring signcryption from $\mathcal{L}$ to $ID_{\mathbb{R}}$, else outputs "Invalid".*

   $\mathcal{A}$ queries the various oracles adaptively, i.e. the current oracle requests may depend on the response to the previous oracle queries.
3. ***Challenge.*** *$\mathcal{A}$ chooses two plaintexts $\{m_0, m_1\} \in \mathcal{M}$ of equal length, a set of $\overline{n}$ users $\mathcal{L}^* = \{\mathcal{U}_i{}^*\}_{(i=1\ to\ \overline{n})}$ and a receiver identity ID_{R^*} and sends them to $\mathcal{C}$. $\mathcal{A}$ should not have queried the private key corresponding to ID_{R^*} in the Phase-I. $\mathcal{C}$ now chooses a bit $\delta \in_R \{0, 1\}$ and computes the challenge ring signcryption C^* of m_δ and sends C^* to $\mathcal{A}$.*
4. ***Phase-II.*** *$\mathcal{A}$ performs polynomially bounded number of requests just like the Phase-1, with the restrictions that $\mathcal{A}$ cannot make* ***Key Extraction query*** *on $ID_{\mathbb{R}^*}$ and should not query for unsigncryption query on C^*. It should be noted that $ID_{\mathbb{R}^*}$ can be included as a ring member in $\mathcal{L}^*$, but $\mathcal{A}$ cannot query the private key of $ID_{\mathbb{R}^*}$.*
5. ***Guess.*** *Finally, $\mathcal{A}$ produces a bit δ' and wins the game if $\delta' = \delta$. The success probability is defined by:*

$$\mathrm{Succ}_{\mathcal{A}}^{\ IND-IRSC-CCA2}(\kappa) = \frac{1}{2} + \varepsilon.$$

Here, ε is called the advantage for the adversary in the above game.

Remark. The security model described here deals with insider security, since the adversary is assumed to have access to the private key of a user who belong to to ring $\mathcal{U}^*$ chosen for Challenge phase.

Definition 2 (Unforgeability). *An identity based ring signcryption scheme (IRSC) is said to be existentially unforgeable against adaptive chosen message attack (EUF-IRSC-CMA), if no polynomially bounded adversary has non-negligible advantage in the following game:*

1. ***Setup Phase.*** *The challenger $\mathcal{C}$ runs the* ***Setup*** *algorithm with the security parameter κ to generate the system parameters* ***params*** *and the master secret key msk. $\mathcal{C}$ gives params to the adversary $\mathcal{A}$ and keeps* ***msk*** *secret.*
2. ***Training Phase.*** *$\mathcal{A}$ performs polynomially bounded number of queries as described in Phase-I of* ***Definition 1****.*
3. ***Existential Forgery.*** *Finally, $\mathcal{A}$ produces a new triple $(\mathcal{U}^*, ID^*_{\mathbb{R}}, C^*)$ (i.e. this triple that was not produced as output by the signcryption oracle), where the private keys of the users in the ring $\mathcal{L}^*$ were not queried during the* ***training phase****. $\mathcal{A}$ wins the game if the result of the Unsigncryption $(\mathcal{L}^*, ID^*_{\mathbb{R}}, C^*)$ is not "Invalid" in other words, C^* is a valid signcryption of some message $m \in \mathcal{M}$. It should be noted that $ID^*_{\mathbb{R}}$ can also be member of the ring $\mathcal{L}$ and in that case, the private key of $ID^*_{\mathbb{R}}$ should not be queried by $\mathcal{A}$. However, if $ID^*_{\mathbb{R}} \notin \mathcal{L}^*$, $\mathcal{A}$ may query the private key of $ID^*_{\mathbb{R}}$.*

Remark. The security model described here deals with insider security since the adversary is assumed to have access to the private key of the receiver of a signcryption used for generation of C^*. This means that the unforgeability is preserved even if a receiver′s private key is compromised.

Definition 3 (Anonymity). *An ID-based ring signcryption scheme is unconditionally anonymous if for any group of n members($n \geq 3$) with identities $\mathcal{L} = ID_i$ ($1 \leq i \leq n$), any message m and Ciphertext C, any adversary cannot identify the actual signcrypter with probability better than a random guess.*
That is, $\mathcal{A}$ outputs the identity of actual signcrypter with probability 1/n if he is not the member of $\mathcal{L}$, and with probability 1/(n - 1) if he is the member of $\mathcal{L}$.

Definition 4 (Public Verifiability). *An ID-based ring signcryption scheme is publicly verifiable if given a*

ciphertext C, ring $\mathcal{L}$, and receiver $\mathbb{R}$, anyone can verify that C is a valid signcryption by some member of the ring $\mathcal{L}$ to the specified receiver $\mathbb{R}$, without knowing the receiver's private key.

4 ATTACK ON SIGNCRYPTION SCHEME BY ZHANG ET AL. (Zhang et al., 2009b)

In this section, we review the scheme in (Zhang et al., 2009b) and demonstrate various attacks on the scheme. We propose attacks on confidentiality, unforgeability and anonymity of (Zhang et al., 2009b).

4.1 Overview of the Scheme in (Zhang et al., 2009b)

Here, we review the ring signcryption scheme proposed in (Zhang et al., 2009b), which was proposed as an improvement to the scheme in (Zhang et al., 2009a). They claim that the scheme remedies the weaknesses of J.H Zhang et al.'s scheme (Zhang et al., 2009a) and it satisfies the semantic security, unforgeability, sender identity's ambiguity, and public authenticity. The scheme (Zhang et al., 2009b) consists of the following algorithms.

1. ***Setup.*** Given a security parameter κ, the PKG chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^k$ (with $\mathbb{G}_1$-additive group and $\mathbb{G}_2$-multiplicative group), bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, a generator P of $\mathbb{G}_1$. PKG randomly picks the master key $s \in Z_q^*$ and computes $P_{pub} = sP$. Next, PKG chooses three cryptographic hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^{n_1}$, $H_3 : \{0,1\}^l \times \mathbb{G}_1 \rightarrow Z_q^*$, where n_1 and l are the sizes of plaintext and ciphertext respectively. The PKG keeps the master private key s secret and publishes the system parameters $params = \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3 \rangle$.
2. ***Key Extract.*** Given an identity ID_i, PKG computes user public key $Q_i = H_1(ID_i)$ and corresponding private key $D_i = sQ_i$.
3. ***AnonymousSigncrypt.*** Let $\mathcal{L} = \{\mathcal{U}_i\}_{(i=1,\ldots,n)}$ be a set of n users including the actual signcrypter $ID_{\mathbb{S}}$. To signcrypt a message m on behalf of the group $\mathcal{L}$ to receiver $ID_{\mathbb{R}}$, $ID_{\mathbb{S}}$ performs the following:
   (a) For $i = 1,\ldots,n (i \neq \mathbb{S})$, randomly picks $x_i \in Z_q^*$ and computes $R_i = x_i P$.
   (b) For the actual sender $\mathbb{S}$, randomly picks $x_{\mathbb{S}} \in Z_q^*$ and computes $\omega = e(P_{pub}, \sum_{i=1}^{n} x_i Q_{\mathbb{R}})$, and sets $c = H_2(\omega) \oplus m$.
   (c) Computes $R_{\mathbb{S}} = x_{\mathbb{S}} Q_{\mathbb{S}} - \sum_{i=1, i\neq\mathbb{S}}^{n} (H_3(c, R_i)Q_i + R_i)$ and $U = \sum_{i=1}^{n} x_i P$.
   (d) Computes $S = (x_{\mathbb{S}} + H_3(c, R_{\mathbb{S}}))D_{\mathbb{S}}$.
   (e) Finally, outputs the ciphertext $C = \langle c, S, U, R_1, \ldots, R_n \rangle$.
4. ***UnSigncrypt.*** Upon receiving the ciphertext $C = (c, S, U, R_1,\ldots,R_n)$, the receiver $U_{\mathbb{R}}$ with identity $ID_{\mathbb{R}}$ uses his private key $D_{\mathbb{R}}$ to recover and verify the message as follows:
   (a) Checks whether $\hat{e}(S, P) \stackrel{?}{=} \hat{e}(P_{pub}, \sum_{i=1}^{n} (R_i + H_3(c, R_i)Q_i))$. If the test passes, computes $\omega' = e(U, D_B)$, then recovers plaintext $m = c \oplus H_2(\omega')$; otherwise outputs "Invalid".

Attack on Confidentiality of the Scheme. During the Challenge phase, let $\{m_0, m_1\}$ be the messages chosen by the adversary $\mathcal{A}$ and sent to the challenger $\mathcal{C}$. Assume that $\mathcal{C}$ chooses $\delta \in_R \{0,1\}$ and computes challenge ring signcryption on m_δ as $C^* = (c^*, S^*, U^*, R_1{}^*, \ldots, R_n{}^*)$ for the receiver $ID_{\mathbb{R}}^*$ and sends C^* to $\mathcal{A}$. Now $\mathcal{A}$ can find the message used for generating C^* by generating a new C' derived from C^* but with a different sender group. $\mathcal{A}$ performs the following steps to find if C^* is a signcryption of m_0 or m_1, during the second phase of oracle queries.

1. $\mathcal{A}$ forms a new group $\mathcal{L}' = \{\mathcal{U}'_1, \ldots, \mathcal{U}'_\eta\}$ with η members who are totally different from the users in $\mathcal{L}^*$ present in the challenge ring signcryption. The private key $D'_{\mathbb{E}}$ of user $\mathcal{U}'_{\mathbb{E}}$, is known to $\mathcal{A}$, where $\mathcal{U}'_{\mathbb{E}} \in \mathcal{L}'$.
2. For $i = 1, \ldots, \eta (i \neq \mathbb{E})$, $\mathcal{A}$ randomly picks $x'_i \in Z_q{}^*$ and computes $R'_i = x'_i P$.
3. For $i = \mathbb{E}$, $\mathcal{A}$ randomly picks $x'_{\mathbb{E}} \in Z_q{}^*$, computes $R'_{\mathbb{E}} = x'_{\mathbb{E}} Q'_{\mathbb{E}} - \sum_{i=1, i\neq\mathbb{E}}^{\eta} (H_3(c^*, R'_i)Q'_i + R'_i)$.
4. $\mathcal{A}$ computes $S' = (x'_{\mathbb{E}} + H_3(c^*, R'_{\mathbb{E}}))D'_{\mathbb{E}}$.
5. $\mathcal{A}$ constructs a ring signcryption $C' = (c^*, S', U^*, R'_1, \ldots, R'_\eta)$ generated by $\mathcal{U}'_{\mathbb{E}}$ using the ring $\mathcal{L}'$ to the receiver $ID_{\mathbb{R}}^*$.
6. During the second phase of training, $\mathcal{A}$ requests the unsigncryption of C' to $\mathcal{C}$. Now $\mathcal{C}$ computes $\omega' = e(U^*, D_{\mathbb{R}}^*)$, then recovers plaintext $m_\delta = c^* \oplus H_2(\omega')$. Note that c^* and U^* components of C^* are not altered in C'.
7. $\mathcal{C}$ responds with $M = m_\delta$ as the output to $\mathcal{A}$.
8. $\mathcal{A}$ now obtains M and thus correctly identifies the message in the challenge ring signcryption C^*.

The new C' will pass the validation test as a valid signcryption of m_δ from ring $\mathcal{L}'$ to the same receiver $ID^*_{\mathbb{R}}$. This can be shown by

$$\begin{aligned}\text{RHS} &= \hat{e}(P_{pub}, \sum_{i=1}^{n}(R'_i + H_3(c^*, R'_i)Q'_i)) \\ &= \hat{e}(sP, \sum_{i=1,i\neq\mathbb{E}}^{n}(R'_i + H_3(c^*, R'_i)Q'_i) + R'_{\mathbb{E}} + \\ &H_3(c^*, R'_{\mathbb{E}})Q'_{\mathbb{E}}) \\ &= \hat{e}(sP, (x'_{\mathbb{E}}Q'_{\mathbb{E}} + H_3(c^*, R'_{\mathbb{E}})Q'_{\mathbb{E}}) \\ &= \hat{e}(P, S') \\ &= LHS\end{aligned}$$

This clearly shows that S' will pass the verification test during unsigncryption.

Attack on unforgeability of the scheme. The scheme in (Zhang et al., 2009b) is not secure against forgeability attacks. The forger $\mathcal{F}$ aims to generate the signcryption of the message m by $ID_{\mathbb{S}}$ using the ring $\mathcal{L} = \{ID_1, ID_2, ..., ID_{\mathbb{S}}, ...ID_n\}$ to a receiver $ID_{\mathbb{R}}$. The details of the attack are as follows.

1. During the training phase, $\mathcal{F}$ queries the H_1 oracle for the identities ID_B, $ID_{\mathbb{R}}$ and $\{ID_1, ID_2, ...ID_n\}$ and $\mathcal{F}$ does not execute **KeyExtract** queries on the above identities.
2. $\mathcal{F}$ gives a signcryption query on a message m from the sender $ID_{\mathbb{S}}$ to the receiver $ID_{\mathbb{B}}$.
3. If the signcryption oracle returns σ as the result of the previous query, $\mathcal{F}$ can submit σ as a valid signcryption from $ID_{\mathbb{S}}$ to $ID_{\mathbb{R}}$.

This attack is possible due to the lack of binding between the signature part of the signcryption and the receiver.

Attack on Anonymity of the Scheme. We show that the scheme in (Zhang et al., 2009b) does not provide anonymity. Any passive observer including the receiver, who is in possession of a ring signcryption can correctly identify the sender of the ring signcryption. This can be demonstrated as follows.
Let $C = \langle c, S, U, R_1, ..., R_n\rangle$ be the ring signcryption on some message m from the ring $\mathcal{L} = \{ID_1, ID_2, ...ID_n\}$ to $ID_{\mathbb{R}}$ and let $ID_{\mathbb{S}} \in \mathcal{L}$ be the actual sender. On receiving the ring signcryption C, anyone can do the following operations to identify the actual sender $ID_{\mathbb{S}} \in \mathcal{L}$.

1. Compute $R = \sum_{i=1}^{n}(R_i + (h_iQ_i))$, *where* $h_i = H_3(c, R_i)$
2. For *j = 1 to n*, compute $M_j = R - h_jQ_j$

$$M_j = \begin{cases} x_{\mathbb{S}}Q_{\mathbb{S}}; & \text{if j} = \mathbb{S}; \\ x_{\mathbb{S}}Q_{\mathbb{S}} + h_{\mathbb{S}}Q_{\mathbb{S}} - h_jQ_j; & \text{if j} \neq \mathbb{S}; \end{cases}$$

.

3. For *j = 1 to n*, compute $N_j = \sum_{i=1,i\neq j}^{n} R_i$

$$N_j = \begin{cases} \sum_{i=1,i\neq j}^{n} x_iP; & \text{if j} = \mathbb{S}; \\ x_{\mathbb{S}}Q_{\mathbb{S}} - x_jP - \sum_{j=1,j\neq\mathbb{S}}^{n} h_jQ_j; & \text{if j} \neq \mathbb{S}; \end{cases}$$

4. For *j = 1 to n*, compute $X_j = \hat{e}(M_j, P)$ and $Y_j = \hat{e}(N_j, Q_j)$.
5. For *j = 1 to n*, compute $Z_j = X_jY_j$.

$$Z_j = \begin{cases} \hat{e}(U, Q_{\mathbb{S}}); \text{if j} = \mathbb{S}; \\ \hat{e}(Q_{\mathbb{S}}, P)^{x_{\mathbb{S}}+h_{\mathbb{S}}}\hat{e}(Q_j, P)^{-(x_j+h_j)} \\ \quad \hat{e}(Q_j, Q_{\mathbb{S}})^{x_{\mathbb{S}}}\prod_{i\neq\mathbb{S}}\hat{e}(Q_i, Q_j)^{-h_i}; \text{if j} \neq \mathbb{S}; \end{cases}$$

Using the steps 1 to 5, the sender $\mathcal{U} \in \mathcal{L}$ can be identified.

5 IDENTITY BASED RING SIGNCRYPTION SCHEME WITH PUBLIC VERIFIABILITY

In this section, we present a new identity based ring signcryption scheme incorporating public verifiability property. The scheme consists of the following algorithms.

1. **Setup**(κ)**.** This algorithm is executed by the PKG to setup the system by taking the security parameter κ as input.
   (a) Selects $\mathbb{G}_1$ an additive cyclic group and $\mathbb{G}_2$ a multiplicative cyclic group, both with same prime order $q > 2^\kappa$ and a random generator $P \in \mathbb{G}_1$.
   (b) Selects $s \in_R \mathbb{Z}_q{}^*$ as the master private key and sets $P_{pub} = sP$ as the master public key.
   (c) Selects a CPA-secure symmetric key encryption system (E, D).
   (d) Picks a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
   (e) Selects five cryptographic hash functions
      i. $H_1 : \{0,1\}^* \to \mathbb{G}_1$
      ii. $H_2 : \mathbb{G}_2 \to \{0,1\}^*$
      iii. $H_3 : \{0,1\}^{|\mathcal{M}|} \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \{0,1\}^* \to \mathbb{Z}_q{}^*$
      iv. $H_4 : \mathbb{G}_1 \times \mathbb{G}_2 \times \{0,1\}^{|\mathcal{M}|} \to \{0,1\}^{|\mathcal{M}|}$
      v. $H_5 : \{0,1\}^{|\mathcal{M}|} \times \mathbb{G}_1 \times \mathbb{G}_1 \times \{0,1\}^* \to \mathbb{G}_1$
   (f) The public parameters of the scheme are set to be *params* = $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3, H_4, H_5, q \rangle$.
2. **Keygen**(ID_i)**.** This algorithm takes ID_i, the identity of a user $\mathcal{U}_i$ as input. The *PKG* who executes this algorithm computes the private key and public key for the user with identity ID_i as follows:

(a) The public key is computed as $Q_i = H_1(ID_i)$.
(b) The corresponding private key, $D_i = sQ_i$.
(c) PKG sends D_i to user $\mathcal{U}_i$ via a secure channel.

3. **Signcrypt**(m, $\mathcal{L}$, $ID_{\mathbb{S}}$, $D_{\mathbb{S}}$, $ID_{\mathbb{R}}$)**.** Let $\mathcal{L} = \{\mathcal{U}_i\}$(i = 1, 2,..., n) be a set of n users including the actual signcrypter $ID_{\mathbb{S}}$. To signcrypt a message m on behalf of the group $\mathcal{L}$ to receiver $ID_{\mathbb{R}}$, $ID_{\mathbb{S}}$ executes as follows:
   (a) Picks a random $x \in Z_q^*$ and computes $U = xP$.
   (b) Computes $\omega = \hat{e}(xP_{pub}, Q_{\mathbb{R}})$, $k = H_2(\omega)$ and sets $\sigma_1 = E_k(m)$.
   (c) For $i = 1$ to n, $i \neq \mathbb{S}$
   - Chooses $R_i \in_R \mathbb{G}_1$
   - Computes $h_i = H_3(\sigma_1, R_i, U, Q_{\mathbb{R}}, \mathcal{L})$.

   (d) For i = $\mathbb{S}$
   - Chooses $x_{\mathbb{S}} \in_R \mathbb{Z}_q^*$
   - Computes $R_{\mathbb{S}} = x_{\mathbb{S}}Q_{\mathbb{S}} - \sum_{i=1, i\neq\mathbb{S}}^{n} (R_i + h_iQ_i)$
   - Computes $h_{\mathbb{S}} = H_3(\sigma_1, R_{\mathbb{S}}, U, Q_{\mathbb{R}}, \mathcal{L})$.

   (e) Computes $R = \sum_{i=1}^{n} R_i$, $\sigma_2 = H_4(R, \omega, m)$, $S_1 = (x_{\mathbb{S}} + h_{\mathbb{S}})D_{\mathbb{S}}$, and $S_2 = xH_5(\sigma_1, R, Q_{\mathbb{R}}, \mathcal{L})$.
   (f) Finally the sender outputs the ciphertext as $C = \langle \sigma_1, \sigma_2, S_1, S_2, U, R_1, ..., R_n \rangle$ to the receiver.

4. **Unsigncrypt**($C = \langle \sigma_1, \sigma_2, S_1, S_2, U, R_1,...,R_n \rangle$, $\mathcal{L}$, $ID_{\mathbb{R}}$, $D_{\mathbb{R}}$)**.** Upon receiving the ciphertext C, $ID_{\mathbb{R}}$ uses his private key $D_{\mathbb{R}}$ to recover the message and verify the signcryption as follows.
   - $\omega' = \hat{e}(U, D_{\mathbb{R}})$, $k' = H_2(\omega')$, $m' = D_{k'}(\sigma_1)$.
   - Check $\sigma_2 \stackrel{?}{=} H_4(R, \omega', m')$

5. **Public-verifiability**($C = \langle \sigma_1, \sigma_2, S_1, S_2, U, R_1,...,R_n \rangle$, $\mathcal{L}$)**.** Upon receiving the ciphertext C, the receiver or any third-party can verify the signcryption for sender authenticity as follows:
   - For $i = 1$ to n ,$h_i = H_3(\sigma_1, R_i, U, Q_{\mathbb{R}}, \mathcal{L})$
   - $H = H_5(\sigma_1, R, Q_{\mathbb{R}}, \mathcal{L})$
   - $\hat{e}(S_1, P) \stackrel{?}{=} \hat{e}(P_{pub}, \sum_{i=1}^{n} (R_i + h_iQ_i))$
   - $\hat{e}(S_2, P) \stackrel{?}{=} \hat{e}(U, H)$
   - If the above validity checks fail, outputs "Invalid" ;

6 CORRECTNESS AND SECURITY ANALYSIS

6.1 Correctness

If the ciphertext C is generated in the way described as above algorithm, it has

$$\begin{aligned}\omega' &= \hat{e}(U, D_{\mathbb{R}}) \\ &= \hat{e}(xP, sQ_{\mathbb{R}}) \\ &= \hat{e}(xP_{pub}, Q_{\mathbb{R}}) \\ &= \omega.\end{aligned}$$

Furthermore,

$$\begin{aligned}&\hat{e}(P_{pub}, \sum_{i=1}^{n} (R_i + h_iQ_i)) \\ &\quad = \hat{e}(sP, \sum_{i=1, i\neq\mathbb{S}}^{n} (R_i + h_iQ_i) + R_{\mathbb{S}} + h_{\mathbb{S}}Q_{\mathbb{S}}) \\ &\quad = \hat{e}(sP, (x_{\mathbb{S}} + h_{\mathbb{S}})Q_{\mathbb{S}}) \\ &\quad = \hat{e}(P, S_1).\end{aligned}$$

6.2 Security Analysis

Theorem 1 (Confidentiality). *If an IND-IBRSC-CCA2 adversary $\mathcal{A}$ has an advantage ε against IBRSC scheme, asking q_{H_i} $(i = 1,2,3,4,5)$ hash queries to random oracles O_{H_i} $(i = 1,2,3,4,5)$, q_e extract queries ($q_e = q_{e1} + q_{e2}$, where q_{e1} and q_{e2} 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 $\varepsilon(\frac{1}{q_{H_1}q_{H_2}})$.*

Proof. The challenger $\mathcal{C}$ is challenged with an instance *(P, aP, bP, cP)* of the *CBDHP*. Assume that there is an adversary $\mathcal{A}$ capable of breaking the $IND-IBRSC-CCA2$ security of *IBRSC* with non-negligible advantage. $\mathcal{C}$ makes use of $\mathcal{A}$ to solve the *CBDHP* instance. $\mathcal{C}$ simulates the system with the various oracles O_{H_1}, O_{H_2}, O_{H_3}, O_{H_4}, O_{H_5}, $O_{Signcryption}$, $O_{Unsigncryption}$ and allows $\mathcal{A}$ to make polynomially bounded number of queries, adaptively to these oracles. The game between $\mathcal{C}$ and $\mathcal{A}$ is demonstrated below:

Setup Phase. $\mathcal{C}$ simulates the system by setting up the system parameters in the following way.

- $\mathcal{C}$ chooses the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and the generator $P \in \mathbb{G}_1$ as given in *CBDHP* instance.
- Sets the master public key $P_{pub} = aP$, here $\mathcal{C}$ does not know a. $\mathcal{C}$ is using the aP value given in the instance of the *CBDHP*.
- Models the five hash functions as random oracles O_{H_1}, O_{H_2} ,O_{H_3}, O_{H_4} and O_{H_5}.
- Selects a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
- Delivers $\mathbb{G}_1$, $\mathbb{G}_2$, $\hat{e}$, P, P_{pub} to $\mathcal{A}$.

Phase I . To handle the oracle queries, $\mathcal{C}$ maintains five lists L_i, $(i = 1,2,3,4,5)$ which keeps track of the responses given by $\mathcal{C}$ to the corresponding oracle (O_{H_1}, O_{H_2}, O_{H_3}, O_{H_4}, O_{H_5}) queries. $\mathcal{A}$ adaptively

queries the various oracles in the first phase, which are handled by $\mathcal{C}$ as given below:

O_{H_1} **Oracle Query.** Assume that $\mathcal{A}$ queries the 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, 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 $\mathcal{A}$ asks the γ^{th} query on ID_γ, $\mathcal{C}$ decides to fix ID_γ as target identity for the challenge phase. $\mathcal{C}$ responds to $\mathcal{A}$ as follows:

- If it is the γ^{th} query, then $\mathcal{C}$ 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. $\mathcal{C}$ is simply using the value bP given in the instance of the *CBDHP*.
- For all other queries, $\mathcal{C}$ chooses $x_i \in_R Z_q^*$ and sets $Q_i = x_i P$ and stores $\langle ID_i, Q_i, x_i \rangle$ in the list L_1 .

$\mathcal{C}$ returns Q_i to $\mathcal{A}$.

O_{H_2} **Oracle Query.** When $\mathcal{A}$ makes a query to this oracle with ω as input, $\mathcal{C}$ retrieves h_2 from list L_2 and returns h_2 to $\mathcal{A}$, if the tuple exists in the list; else, chooses a new h_2 randomly, stores $\langle \omega, h_2 \rangle$ in L_2 and returns h_2 to $\mathcal{A}$.

O_{H_3} **Oracle Query.** When $\mathcal{A}$ makes a query to this oracle with $(c, R_i, U, Q_R, \mathcal{L})$ as input, $\mathcal{C}$ retrieves $h_i^{(3)}$from list L_3 and returns $h_i^{(3)}$ to $\mathcal{A}$ if the tuple exists in the list ; else, chooses a new $h_i^{(3)} \in_R Z_q^*$ randomly, stores $\langle c, R_i, U, Q_R, \mathcal{L}, h_i^{(3)} \rangle$, in the list L_3 and returns $h_i^{(3)}$ to $\mathcal{A}$.

O_{H_4} **Oracle Query.** When $\mathcal{A}$ makes a query to this oracle with (R, ω, m) as input, $\mathcal{C}$ retrieves ψ from list L_4 and returns ψ to $\mathcal{A}$ if the tuple exists in the list; else, chooses $\psi \in \{0,1\}^{|\mathcal{M}|}$, stores $\langle R, \omega, m, \psi \rangle$ in L_4 and returns ψ to $\mathcal{A}$.

O_{H_5} **Oracle Query.** When $\mathcal{A}$ makes a query to this oracle with $(\sigma_1, R, Q_i, \mathcal{L})$ as input, $\mathcal{C}$ retrieves h from list L_5 and returns h to $\mathcal{A}$ if the tuple exists in the list; else, chooses $r \in_R \mathbb{G}_1$, computes $h = r$, if $ID_i \neq ID_\gamma$, and computes $h = rP_{pub}$ if $ID_i = ID_\gamma$. The tuple $\langle \sigma_1, R, Q_i, \mathcal{L}, h \rangle$ is stored in list L_5 and returns h to $\mathcal{A}$.

Extract Query. On getting a request for the private key of user U_i with identity ID_i, $\mathcal{C}$ aborts if $ID_i = ID_\gamma$. Else, $\mathcal{C}$ retrieves $\langle Q_i, x_i \rangle$ from list L_1 and returns $D_i = aQ_i = x_i aP$ to $\mathcal{A}$.

$O_{Signcryption}$ **Query.** $\mathcal{A}$ chooses a message m, a set of n potential senders and forms an ad-hoc group $\mathcal{L}$ by fixing a sender $ID_\mathbb{S}$ and a receiver $ID_\mathbb{R}$ and sends them to $\mathcal{C}$. To respond correctly to the signcryption query on the plaintext m chosen by $\mathcal{A}$, $\mathcal{C}$ does the following:

$\mathcal{C}$ proceeds according to the signcryption algorithm when $ID_\mathbb{S} \neq ID_\gamma$. This is possible as $\mathcal{C}$ knows the private key $D_\mathbb{S}$ of the sender $ID_\mathbb{S}$.
If the sender's identity $ID_\mathbb{S} = ID_\gamma$ (i.e. when $\mathcal{C}$ does not know the private key corresponding to $ID_\mathbb{S}$), $\mathcal{C}$ cooks up a response as explained below:

- Chooses a random $x \in Z_q^*$, computes $U = xP$, $\omega = \hat{e}(xP_{pub}, Q_\mathbb{R})$ and sets $\sigma_1 = E_k(m)$.
- For $i = 1$ to n, $i \neq \mathbb{S}$, chooses $R_i \in \mathbb{G}_1$ and computes $h_i^{(3)} = H_3(c, R_i, U, Q_\mathbb{R}, \mathcal{L})$.
- For $i = \mathbb{S}$,
  - Chooses $x_\mathbb{S}$, $h_\mathbb{S}^{(3)} \in Z_q^*$.
  - Computes $R_\mathbb{S} = x_\mathbb{S} P - h_\mathbb{S}^{(3)} Q_\mathbb{S} - \sum_{i=1, i\neq\mathbb{S}}^{n} (R_i + h_i Q_i)$.
  - Adds the tuple $\langle c, R_\mathbb{S}, U, Q_\mathbb{S}, \mathcal{L}, h_\mathbb{S}^{(3)} \rangle$ to the list L_3.

  (**Note.** Here $h_\mathbb{S}^{(3)}$ is not computed by $\mathcal{C}$, instead it is chosen at random and set as the output for the random oracle query $h_\mathbb{S}^{(3)} = H_3(c, R_\mathbb{S}, U, Q_\mathbb{R}, \mathcal{L})$ This is possible because the random oracles are manipulated by $\mathcal{C}$).
- Computes $S_1 = x_\mathbb{S} P_{pub}$ and $S_2 = xH_5(\sigma_1, R, Q_\mathbb{R}, \mathcal{L})$.
- Computes $R = \Sigma R_i$ and queries σ_2 from O_{H_4}.

Finally, $\mathcal{C}$ outputs the ring signcryption $C = \langle \sigma_1, \sigma_2, S_1, S_2, U, R_1,...,R_n \rangle$ to $\mathcal{A}$ as the signcryption of m. The signcryption $C = \langle \sigma_1, \sigma_2, S_1, S_2, U, R_1,...,R_n \rangle$ is considered as valid by $\mathcal{A}$ because $\mathcal{C}$ passes the verification tests as shown below.
From the definition of $R_\mathbb{S}$, $\sum_{i=1}^{n} (R_i + h_i Q_i) = x_\mathbb{S} P$. Thus,

$$\begin{aligned} \hat{e}(P_{pub}, \sum_{i=1}^{n} (R_i + h_i Q_i)) &= \hat{e}(aP, x_\mathbb{S} P) \\ &= \hat{e}(P, x_\mathbb{S} aP) \\ &= \hat{e}(P, x_\mathbb{S} P_{pub}) \\ &= \hat{e}(P, V) \end{aligned}$$

$O_{Unsigncryption}$ **Query.** Upon receiving an unsigncryption query on a ring signcryption $C = \langle \sigma_1, \sigma_2, S_1, S_2, U, R, R_1,..., R_n \rangle$ with $ID_\mathbb{R}$ as receiver, $\mathcal{C}$ proceeds as follows:

$\mathcal{C}$ proceeds as per the unsigncryption algorithm, when $ID_{\mathbb{R}} \neq ID_{\gamma}$. Here, $\mathcal{C}$ can directly use the unsigncryption algorithm because, $\mathcal{C}$ knows the private key $D_{\mathbb{R}}$ of the receiver $ID_{\mathbb{R}}$.
If the receiver identity $ID_{\mathbb{R}} = ID_{\gamma}$ (i.e. When $\mathcal{C}$ does not know the private key corresponding to $ID_{\mathbb{R}}$), $\mathcal{C}$ generates the response as explained below:

1. For $i = 1$ to n, Compute $h_i = \mathcal{O}_{H_3}$ (c, R_i, U, $Q_{\mathbb{R}}$, $\mathcal{L}$) and check whether $\hat{e}(S_1, P) \stackrel{?}{=} \hat{e}(P_{pub}, \sum_{i=1}^{n}(R_i + h_i Q_i))$.
2. If the above equation holds,then for each pair (m, ω) in the list L_4, the challenger $\mathcal{C}$ performs the following:
   (a) Computes $k' = \mathcal{O}_{H_2}(\omega)$.
   (b) Computes $R = \Sigma R_i$
   (c) Retrieves the message as $m' = E_{k'}(c)$.
   (d) Checks whether $\omega \stackrel{?}{=} \hat{e}(S_1, Q_R)$.
   (e) Checks whether $m' \stackrel{?}{=} m$ and $\sigma \stackrel{?}{=} \mathcal{O}_{H_4}(R, \omega', m')$.
3. The first time when all the above checks passes, $\mathcal{C}$ outputs the corresponding m' and halts.
4. If every (m, ω) pair fails the check in step(2) then $\mathcal{C}$ outputs "Invalid" and halts.

Challenge Phase. Finally, $\mathcal{A}$ chooses two plaintexts $m_0, m_1 \in \mathcal{M}$, the set of ring members $\mathcal{L} = ID_i$ ($i = 1$ to $\overline{n}$), a sender identity $ID_{\mathbb{S}} \in \mathcal{L}$ and a receiver identity $ID_{\mathbb{R}}$ on which $\mathcal{A}$ wants to be challenged and sends them to $\mathcal{C}$. $\mathcal{A}$ should not have queried the private key corresponding to $ID_{\mathbb{R}}$ in the first phase. $\mathcal{C}$ aborts, if $ID_{\mathbb{R}} \neq ID_{\gamma}$; else, $\mathcal{C}$ chooses a bit $\delta \in_R \{0,1\}$ and computes the challenge ring signcryption C of m_{δ} as follows :

- Sets $U^* = cP$. Here $\mathcal{C}$ is using the value cP given in the instance of *CBDHP*.
- Chooses $\{R_i^*\}_{(i=1\ to\ \overline{n})}$, S_1^*, $S_2^* \in_R \mathbb{G}_1$ and $\sigma_1{}^* \in_R \{0,1\}^{|M|}$, $\sigma_2{}^* \in_R \mathbb{Z}_q^*$, and outputs $C^* = \langle \sigma_1^*, \sigma_2^*, S_1^*, S_2^*, U^*, R^*, R_1^*, ..., R_n^* \rangle$.

Phase II. On getting the challenge ring signcryption C^*, $\mathcal{A}$ is allowed to interact with $\mathcal{C}$ as in the first phase. But this time, $\mathcal{A}$ is not given access to the private key of $ID_{\mathbb{R}}$ and is also restricted from querying the decryption oracle for the ring unsigncryption of C^*.
Guess. At the end of the Phase II, $\mathcal{A}$ returns its guess. $\mathcal{C}$ ignores the answer from $\mathcal{A}$, picks a random tuple (ω, h_2) from list L_2 and returns the corresponding ω as the solution to the *CBDHP* instance. Thus, any adversary that has advantage in the real $IND-IBRSC-CCA2$ game must necessarily recognize with probability ε at least that the challenge ciphertext provided by $\mathcal{C}$ is incorrect. For $\mathcal{A}$ to find that C^* is not a valid ciphertext, $\mathcal{A}$ should have queried the $\mathcal{O}_{H_2}$ oracle with $\omega = \hat{e}(U^*, D_{\gamma})$. Here D_{γ} is the private key of the target identity and it is $a(Q_{\gamma}) = abP$. Also $\mathcal{C}$ has set $U^* = cP$. Hence $\omega = \hat{e}(U^*, D_{\gamma}) = \hat{e}(cP, abP) = \hat{e}(P,P)^{abc}$. 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.

We now consider $\mathcal{C}$'s probability of success. The events in which $\mathcal{C}$ aborts the $IND-IBRSC-CCA2$ game are,

1. E_1 - when $\mathcal{A}$ queries the private key of the target identity ID_{γ} and its probability, $\Pr[E_1] = \frac{q_e}{q_{H_1}}$.
2. E_2 - when $\mathcal{A}$ does not choose the target identity ID_{γ} as the receiver during the challenge phase and its probability, $\Pr[E_2] = \left(1 - \frac{1}{q_{H_1} - q_e}\right)$.

The probability that $\mathcal{C}$ does not abort the $IND-IBRSC-CCA2$ game is given by
$(Pr[\neg E_1 \wedge \neg E_2]) = \left(1 - \frac{q_e}{q_{H_1}}\right)\left(\frac{1}{q_{H_1} - q_e}\right) = \frac{1}{q_{H_1}}$.
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, $Pr[C(P, aP, bP, cP) = \hat{e}(P,P)^{abc}] = \varepsilon\left(\frac{1}{q_{H_1} q_{H_2}}\right)$.
Since ε is non-negligible, the probability of $\mathcal{C}$ solving *CBDHP* is also non-negligible.

Theorem 2 (Unforgeability). *If an $EUF-IBRSC-CMA$ forger $\mathcal{A}$ exists against IBRSC scheme, then there exist an algorithm $\mathcal{C}$ that solves the CDHP with advantage $\varepsilon\frac{1}{q_{H_1}}$.*

Proof. The challenger $\mathcal{C}$ is challenged to solve an instance of the *CDHP*. $\mathcal{C}$ interacts with adversary $\mathcal{A}$ which is capable of breaking the $EUF-IBRSC-CMA$ security of the new scheme, to solve the *CDHP* instance. On receiving the instance (P, aP, bP) of the *CDHP* as input, $\mathcal{C}$ begins the interaction with $\mathcal{A}$ to compute the value abP. $\mathcal{C}$ simulates the system with the various oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{H_4}$, $\mathcal{O}_{H_5}$, $\mathcal{O}_{Signcryption}$, $\mathcal{O}_{Unsigncryption}$ and allows $\mathcal{A}$ to adaptively ask polynomially bounded number of queries to these oracles.
Setup Phase. $\mathcal{C}$ simulates the system by setting up the system parameters in the following way.

- $\mathcal{C}$ chooses the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and the generator $P \in \mathbb{G}_1$ as given in *CDHP* instance.
- Sets the master public key $P_{pub} = aP$, here $\mathcal{C}$ does not know a. $\mathcal{C}$ is using the aP value given in the instance of the *CDHP*.
- Models the five hash functions as random oracles O_{H_1}, O_{H_2}, O_{H_3}, O_{H_4} and O_{H_5}.
- Selects a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
- Delivers $\mathbb{G}_1$, $\mathbb{G}_2$, $\hat{e}$, P, P_{pub} to $\mathcal{A}$.

Training Phase. $\mathcal{A}$ adaptively performs polynomially bounded number of queries to the various oracles in this phase. The queries may be Hash Queries, Extract Queries, $O_{Signcryption}$ Queries and $O_{Unsigncryption}$ Queries, which are handled by $\mathcal{C}$.

All Hash oracle queries are same as that in the confidentiality game discussed above.

Forgery. Finally, $\mathcal{A}$ produces a forged signcryption $C^* = \langle \sigma_1{}^*, \sigma_2{}^*, S_1{}^*, S_2{}^*, U^*, R_1{}^*, ..., R_n{}^* \rangle$ on the message m^* (i.e. C^* was not produced by the Signcryption Oracle as an output for the ring signcryption query on the message m with an ad-hoc set of users $\mathcal{U}^*$ and the receiver $ID_{\mathbb{R}}$),where the private keys of the users who are in the group $\mathcal{U}^*$ were not queried in the training phase. $\mathcal{C}$ aborts if $\mathcal{U}^*$ do not contain the target identity. Else, $\mathcal{C}$ can very well unsigncrypt and verify the validity of the forged ring signcryption C^* (as done in unsigncrypt oracle).

Using forking lemma, we obtain two valid ring signcryptions $C^* = \langle \sigma_1{}^*, \sigma_2{}^*, S_1{}^*, S_2{}^*, U^*, R^*, R_1{}^*, ..., R_n{}^* \rangle$ and $C' = \langle \sigma_1{}', \sigma_2{}', S_1{}', S_2{}', U^*, R^*, R_1{}^*, ..., R_n{}^* \rangle$. On getting two valid ring signcryptions on m^*, $\mathcal{C}$ will be able to retrieve $D_{\mathbb{S}} = abP$ as follows:

- Here $S_1{}' = (x_{\mathbb{S}} + h_{\mathbb{S}}{}')D_{\mathbb{S}}$ and $S_1{}^* = (x_{\mathbb{S}} + h_{\mathbb{S}}{}^*)D_{\mathbb{S}}$ (since they have the same randomness)
- Thus, $S_1{}'$ - $S_1{}^* = (h_{\mathbb{S}}{}' - h_{\mathbb{S}}{}^*)D_{\mathbb{S}}$

Since $\mathcal{C}$ knows the hash values $h_{\mathbb{S}}{}'$ and $h_{\mathbb{S}}{}^*$, $\mathcal{C}$ can compute $D_{\mathbb{S}} = (S_1{}' - S_1{}^*)(h_{\mathbb{S}}{}' - h_{\mathbb{S}}{}^*)^{-1}$. This means, $\mathcal{C}$ can compute abP because $D_{\mathbb{S}} = abP$. In other words, $\mathcal{C}$ is capable of solving *CDHP*, which is not possible. Hence, *IBRSC* is secure against $EUF - IBRSC - CMA$.

Theorem 3 (Anonymity). *The IBRSC scheme is fully anonymous.*

The proof is based on the approach used in (Chow et al., 2005).

Since $\bigcup_{i \neq \mathbb{S}} \{R_i\}$ and $x'_{\mathbb{S}}$ is randomly generated, $\bigcup_{i=1}^{n} \{R_i\}$ values are uniformly distributed. All other components of C except S_1 does not contain any identity information bound to them. So we need to check only whether $S_1 = (x_{\mathbb{S}} + h_{\mathbb{S}})D_{\mathbb{S}}$ leaks information about the actual signer. We have $S_1 - h_{\mathbb{S}}D_{\mathbb{S}} = x_{\mathbb{S}}D_{\mathbb{S}}$. Anyone can compute the value of $x_{\mathbb{S}}Q_{\mathbb{S}}$ by $x_{\mathbb{S}}Q_{\mathbb{S}} = R_{\mathbb{S}} + \sum_{i=1, i \neq \mathbb{S}}^{n} (R_i + h_i Q_i)$. As bilinearity can relate $x_{\mathbb{S}}D_{\mathbb{S}}$ and $x_{\mathbb{S}}Q_{\mathbb{S}}$, by checking whether $\hat{e}(x_{\mathbb{S}}D_{\mathbb{S}}, P) = \hat{e}(x_{\mathbb{S}}Q_{\mathbb{S}}, P_{pub})$, it may be possible to see if ID_j is the actual signcrypter by checking whether the following equality holds:

$$\hat{e}(R_j + \sum_{i \neq j} (R_i + h_i Q_i), P_{pub}) \stackrel{?}{=} \hat{e}(S_1, P)/\hat{e}(h_j Q_j, P_{pub}).$$

But this method is of no use, as the above equality holds $\forall$ j values. i.e. the signature is symmetric. The above equality is just the same as the equality to be checked in the verification algorithm.

$$\begin{aligned}
&\hat{e}(R_j + \sum_{i \neq j} (R_i + h_i Q_i), P_{pub}) \\
&= \hat{e}(\sum_{i \neq \mathbb{S}} R_i + R_{\mathbb{S}} + \sum_{i \neq j} h_i Q_i, P_{pub}) \\
&= \hat{e}(\sum_{i \neq \mathbb{S}} R_i + x_{\mathbb{S}} Q_{\mathbb{S}} - \sum_{i \neq \mathbb{S}} \{R_i + h_i Q_i\} + \sum_{i \neq j} h_i Q_i, P_{pub}) \\
&= \hat{e}(x_{\mathbb{S}} Q_{\mathbb{S}} - \sum_{i \neq \mathbb{S}} \{h_i Q_i\} + \sum_{i \neq j} h_i Q_i, P_{pub}) \\
&= \hat{e}(x_{\mathbb{S}} Q_{\mathbb{S}} + h_{\mathbb{S}} Q_{\mathbb{S}} - h_j Q_j, sP) \\
&= \hat{e}(x_{\mathbb{S}} D_{\mathbb{S}} + h_{\mathbb{S}} D_{\mathbb{S}} - h_j D_j, P) \\
&= \hat{e}(S_1 - h_j D_j, P) \\
&= \hat{e}(S_1, P)/\hat{e}(h_j D_j, P) \\
&= \hat{e}(S_1, P)/\hat{e}(h_j Q_j, P_{pub})
\end{aligned}$$

So, we can conclude that even an adversary with unbounded computing power has no advantage in identifying the actual signcrypter over random guessing.

7 CONCLUSIONS

As a concluding remark we summarize the work in this paper. In this paper we showed the security weakness of an identity based ring signcryption scheme in the literature. We showed that (Zhang et al., 2009b) does not provide security against adaptive chosen ciphertext attacks (CCA2), existential unforgeability attacks and anonymity attacks. We proposed a new identity based ring signcryption scheme as an extension to (Selvi et al., 2009) for which we proved the security against chosen ciphertext attack and existential unforgeability in the random oracle model. We also proved anonymity property of our scheme. Future research direction includes designing an identity based ring signcryption scheme with constant ciphertext length. We provide the comparison of our Identity Based Ring Signcryption Scheme with the existing secure schemes in the following tables.

Table 1: Efficiency comparison - Signcryption.

Scheme	Signcryption			
	SPM	*BP*	$\mathbb{G}_2M$	*PA*
A^*	$2n+2$	$n+2$	1	$2n$
B	$n+2$	1	–	$2n-2$
C	$n+3$	1	–	$2n-2$

Table 2: Efficiency comparison - Unsigncryption.

Scheme	Unsigncryption			
	SPM	*BP*	$\mathbb{G}_2M$	*PA*
A^*	n	3	$n+1$	n
B	n	3	–	$2n-1$
C	n	5	–	$2n-1$

Table 3: Ciphertext size and public verifiability.

Scheme	Ciphertext Size	PV
A	$2\|\mathcal{M}\|+(n+1)\|\mathbb{G}_1\|+n\|\mathbb{Z}_q^*\|$	No
B	$2\|\mathcal{M}\|+(n+2)\|\mathbb{G}_1\|$	No
C	$2\|\mathcal{M}\|+(n+4)\|\mathbb{G}_1\|$	**Yes**

A-Huang et al.(Huang et al., 2005), *B*- Sharmi et al.(Selvi et al., 2009), *C*-IBRSC, *PV*- Public Verifiability, *SPM* - Scalar Point Multiplication, *BP* - Bilinear Pairing, $\mathbb{G}_2M$ - Multiplication of two $\mathbb{G}_2$ elements and *PA* - Point Addition.

** This scheme cannot be considered as a provably secure scheme as the proof given for the model is incorrect.*

REFERENCES

Baek, J., Steinfeld, R., and Zheng, Y. (2002). Formal proofs for the security of signcryption. In *PKC 2002: Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 80–98. Springer-Verlag.

Chow, S. S. M., Yiu, S.-M., and Hui, L. C. K. (2005). Efficient identity based ring signature. In *ACNS*, volume 3531, pages 499–512.

Huang, X., Susilo, W., Mu, Y., and Zhang, F. (2005). Identity-based ring signcryption schemes: Cryptographic primitives for preserving privacy and authenticity in the ubiquitous world. In *AINA '05: Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 649–654. IEEE Computer Society.

Li, F., Shirase, M., and Takagi, T. (2008a). Analysis and improvement of authenticatable ring signcryption scheme. In *International Conference ProvSec-08, Paper appears in Journal of Shanghai Jiaotong University (Science)*, volume 13-6, pages 679–683.

Li, F., Xiong, H., and Yu, Y. (2008b). An efficient id-based ring signcryption scheme. In *International Conference on Communications, Circuits and Systems, 2008. ICCCAS 2008.*, pages 483–487.

Malone-lee, J. (2002). Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098.

Selvi, S. S. D., Vivek, S. S., and Rangan, C. P. (2009). On the security of identity based ring signcryption schemes. In *ISC, 12th International Conference, ISC 2009, Pisa, Italy, September 7-9, 2009. Proceedings*, volume 5735 of *Lecture Notes in Computer Science*, pages 310–325. Springer.

Vivek, S. S., Selvi, S. S. D., and Rangan, C. P. (2009). On the security of two ring signcryption schemes. In *SECRYPT 2009*, pages 219–224. INSTICC Press.

Yu, Y., Li, F., Xu, C., and Sun, Y. (2008). An efficient identity-based anonymous signcryption scheme. *Wuhan University Journal of Natural Sciences*, Volume: 13, Number: 6, December, 2008:670–674.

Zhang, J., Gao, S., Chen, H., and Geng, Q. (2009a). A novel id-based anonymous signcryption scheme. In *APWeb/WAIM*, volume 5446 of *Lecture Notes in Computer Science*, pages 604–610. Springer.

Zhang, M., Yang, B., Zhu, S., and Zhang, W. (2008). Efficient secret authenticatable anonymous signcryption scheme with identity privacy. In *PAISI, PACCF and SOCO '08: Proceedings of the IEEE ISI 2008 PAISI, PACCF, and SOCO international workshops on Intelligence and Security Informatics*, pages 126–137. Springer-Verlag.

Zhang, M., Zhong, Y., Yang, B., and Zhang, W. (2009b). Analysis and improvement of an id-based anonymous signcryption model. In *ICIC (1)*, volume 5754 of *Lecture Notes in Computer Science*, pages 433–442. Springer.

Zhu, Z., Zhang, Y., and Wang, F. (Pages 649-654, http://dx.doi.org/10.1016/j.csi.2008.09.023, 2008). An efficient and provable secure identity based ring signcryption scheme. In *Computer Standards & Interfaces*.

Zhun, L. and Zhang, F. (2008). Efficient identity based ring signature and ring signcryption schemes. In *International Conference on Computational Intelligence and Security, 2008. CIS '08.*, volume 2, pages 303–307.