On the security of Identity Based Ring Signcryption Schemes

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

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

Abstract. Signcryption is a cryptographic primitive which offers authentication and confidentiality simultaneously with a cost lower than signing and encrypting the message independently. The need for ring signcryption is to make it possible for an user to signcrypt a message along with the identities of a set of potential senders (that includes him) without revealing who in the set has actually produced the signcryption. Thus a ring signcrypted message has anonymity in addition to authentication and confidentiality. Ring signcryption schemes have no group managers, no setup procedures, no revocation procedures and no coordination: any user can choose any set of users that includes himself and signcrypt any message by using his private key as well as other users public keys, without getting any approval or assistance from them. Ring Signcryption is used to provide a graceful way to leak trustworthy secrets in an anonymous, authenticated and confidential way.

To the best of our knowledge, seven identity based ring signcryption schemes are reported in the literature. Two of them were already proved to be insecure in [11] and [7]. In this paper, we show that four among the remaining five do not provide confidentiality. We show that two schemes among the four do not resist chosen plaintext attack and the other two schemes do not provide adaptive chosen ciphertext security, i.e. the adversary can perform some easy test to distinguish the ciphertext during the confidentiality game. We then propose a new scheme and formally prove our scheme to be correct. A comparison of our scheme with the only existing correct scheme by Huang et al. shows that our scheme is much more efficient than the scheme by Huang et al.

Keywords: Ring Signcryption, Cryptanalysis, Provable Security, Confidentiality, Chosen Plaintext Attack, Adaptive Chosen Ciphertext Attack, Bilinear Pairing, Random Oracle Model.

1 Introduction

Identity based cryptography (IBC) was introduced by Shamir [8] in the year 1984. It aims in reducing the over head of public key certification which is inherent in the public key infrastructure (PKI). The public key of an user in IBC is not a random string as in PKI, instead it is an unique identifier of an user such as email id, IP address, social security number etc. The user of an identity based cryptosystem is not required to obtain a certificate for his public key, since his identity is well known in public or available in a public directory. IBC employs a trusted third party, namely the private key generator (PKG) to generate the private key for an user, corresponding to its identity at the time of registration of an user with the PKG. Thus, the private key of all users registered with the IBC is known to the PKG.

Signcryption - the cryptographic primitive, proposed by Zheng [12] provides both authenticity and confidentiality with a lower computational cost when compared to signing and encrypting the message independently. Ring signature, that was first proposed by Rivest et al. [6] provides authenticity for a message in an anonymous way, i.e. the verifier does not know who has signed the message but he can verify that one of the person from the ring (group), formed by the signer during signing has done it. Ring signcryption enables an user to send an authentic message confidentially and anonymously to a specified receiver.

Motivation. Ring signcryption as a primitive can be motivated from the following scenario: Let us consider the same example given by Rivest et al. [6], where a member of a cabinet wants to leak a very important

[*] 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

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 a spotted person 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, the information should be authenticated for the authorities in the press to consider it and it should be confidential until it reaches the hands of the right person in the press. All the three properties are together achieved by the single primitive - "Ring Signcryption".

Related Work. The combination of identity based cryptography and ring signcryption yields a scheme which confidentially transmits an authenticated message anonymously to a specific receiver with the advantages of IBC. The first identity based ring signcryption scheme was proposed by Huang et al. [3]. Subsequently identity based ring signcryption schemes are reported in [9, 13, 11, 5, 4, 14].

Huang et al.'s scheme [3] was considered to be inefficient because the sender has to compute $n+2$ pairing for signcrypting a message and to unsigncrypt a ciphertext, the receiver has to compute 3 pairings. In an attempt to improve [3], Yu et al. [9] proposed a scheme entitled as identity based anonymous signcryption scheme which is essentially a ring signcryption scheme. The authors have claimed that their scheme [9] is adaptive chosen ciphertext (CCA2) secure but we show in this paper that their scheme is not at all secure even with respect to chosen plaintext attack (CPA). Fagen Li et al. proposed yet another scheme in [5], where they reduce the total number of pairing operations to 4 (one for signcryption and three for unsigncryption) but their scheme was reported to be faulty with respect to adaptive chosen ciphertext attack in [7]. Following that, Zhang et al. [11] proposed an authenticatable identity based anonymous signcryption scheme, which is also a ring signcryption scheme where the actual sender can prove to a valid verifier that the signcrypt was indeed produced by him. In [4], Fagen Li et al. have shown that Zhang et al.'s scheme does not resist adaptive chosen ciphertext attack and have proposed an improved authenticatable identity based anonymous signcryption scheme. We show that the improvement proposed by Fagen Li et al. [4] is also not adaptive chosen ciphertext secure. Lijun et al. proposed an identity based ring signature and ring signcryption scheme in [14], whose work was followed by Zhu et al. [13] who too proposed an identity based ring signcryption scheme. In this paper we show that the former one [14] is not secure against chosen plaintext attack and the latter one [13] is not adaptive chosen ciphertext secure.

Our Contribution. Anonymous signcryption is another nomenclature for ring signcryption, so both have the same functionalities and authenticatable anonymous signcryption is a ring signcryption scheme, which has to satisfy the security notions of ring signcryption with an additional property that an actual sender can expose himself with a brief interaction with the verifier at a later point of time. We consider these two variants of ring signcryption too in our paper. We show that the schemes reported in [9] and [14] does not withstand chosen plaintext attack, moreover, the schemes reported in [4] and [13] does not resist chosen ciphertext attack, by demonstrating the attack on each scheme independently. We also provide a new scheme which is IND-New-IBRSC-CCA2 (indistinguishable against adaptive chosen ciphertext attack) and EUF-New-IBRSC-ACMA (existentially unforgeable against adaptive chosen message attack) secure. Note that these are the strongest security requirements for encryption and signature schemes. The formal proof of our new scheme is given in the random oracle model. Finally, a comparison with the only existing correct scheme by Huang et al. [3] shows that our scheme is the most efficient identity based ring signcryption scheme available till date.

2 Preliminaries

2.1 Bilinear Pairing

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

- ***Bilinearity.*** For all $P, Q, R \in \mathbb{G}_1$,
  - $\hat{e}(P+Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$
  - $\hat{e}(P, Q+R) = \hat{e}(P, Q)\hat{e}(P, R)$

- $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$

- ***Non-Degeneracy.*** There exist $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq I_{\mathbb{G}_2}$, where $I_{\mathbb{G}_2}$ is the identity in $\mathbb{G}_2$.
- ***Computability.*** There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

2.2 Decisional Bilinear Diffie-Hellman Problem (DBDHP)

Definition 1. *Given $(P, aP, bP, cP, \alpha) \in \mathbb{G}_1^4 \times \mathbb{G}_2$ for unknown $a, b, c \in \mathbb{Z}_q^*$, the DBDH problem in $\mathbb{G}_1$ is to decide if $\alpha = \hat{e}(P, P)^{abc}$.*

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

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

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

2.3 Computation Diffie-Hellman Problem (CDHP)

Definition 2. *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.

2.4 Notations used in his paper

To have a better understanding and to enhance the readability and clarity, we use the following notations throughout the paper.

$\mathcal{U}_i$ - User with identity ID_i.
$\mathcal{U} = \{\mathcal{U}_i\}_{(i = 1\, to\, n)}$ - Group of users in the ring (including the actual sender).
$\mathcal{M}$ - Message space.
m - Message.
l - Number of bits used to represent m.
Q_i - Public key corresponding to ID_i.
D_i - Private key corresponding to ID_i.
$ID_{\mathbb{S}}$ - Identity of the sender.
$ID_{\mathbb{R}}$ - Identity of the receiver.
$Q_{\mathbb{S}}$ - Public key of the sender.
$Q_{\mathbb{R}}$ - Private key of the receiver.
$D_{\mathbb{S}}$ - Public key of the sender.
$D_{\mathbb{R}}$ - Private key of the receiver.

3 Formal Security Model for Identity Based Ring Signcryption

3.1 Generic Scheme

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 $params$, which includes a master public key P_{pub} and a corresponding master private key s that is kept secret.
- ***Extract(ID_i):*** Given an identity ID_i, the PKG computes the corresponding private key D_i and sends it to its owner via a secure channel.

- ***Signcrypt(m,$\mathcal{U}$, $D_{\mathbb{S}}$, $ID_{\mathbb{R}}$):*** This algorithm takes a message $m \in \mathcal{M}$, a receiver with identity $ID_{\mathbb{R}}$, the senders private key $D_{\mathbb{S}}$ and an ad-hoc group of ring members $\mathcal{U}$ with identities $\{ID_1, \ldots, ID_n\}$ as input and outputs a ciphertext C on behalf of the ad-hoc group. This algorithm is executed by a sender with identity $ID_{\mathbb{S}} \in \mathcal{U}$. $ID_{\mathbb{R}}$ may or may not be in $\mathcal{U}$.
- ***Unsigncrypt(C,$\mathcal{U}$, $D_{\mathbb{R}}$):*** This algorithm takes the ciphertext C, The ad-hoc group of user identities from $\mathcal{U}$ and the private key of the receiver $D_{\mathbb{R}}$ as input and produces the plaintext m, if C is a valid ciphertext for m or the symbol '$\perp$', if C is an invalid ciphertext. This algorithm is executed by a receiver $ID_{\mathbb{R}}$.

In other words, make the consistency constraint that, if $C = Signcrypt(m,\mathcal{U}, D_{\mathbb{S}}, ID_{\mathbb{R}})$,then $m = Unsigncrypt$ $(C,\mathcal{U}, D_{\mathbb{R}})$

3.2 Security Notion

The formal security definition of signcryption was given by Baek et al. in [1]. The security requirements for identity based ring signcryption is defined as follows.

Definition 3. *An identity based ring signcryption (IRSC) is indistinguishable against adaptive chosen ciphertext ring attacks(IND-IRSC-CCA2) if there exists no polynomially bounded adversary that has non-negligible advantage in the following game:*

1. ***Setup Phase:*** *The challenger $\mathcal{C}$ runs the Setup algorithm with a security parameter κ and sends the system parameters params to the adversary $\mathcal{A}$ and keeps the master private key secret.*
2. ***First Phase:*** *$\mathcal{A}$ performs polynomially bounded number of queries to the oracles provided to it by $\mathcal{C}$. The description of the queries and responses allowed in the first phase are listed below:*
   - ***Key Extraction query****: $\mathcal{A}$ produces an identity ID_i corresponding to U_i and receives the private key D_i.*
   - ***Signcryption query:*** *$\mathcal{A}$ produces a set of users $\mathcal{U}$, a receiver identity $ID_{\mathbb{R}}$ and a plaintext $m \in_R \mathcal{M}$ to the challenger $\mathcal{C}$. $\mathcal{A}$ also specifies the sender $\mathcal{U}_{\mathbb{S}} \in \mathcal{U}$ whose identity is $ID_{\mathbb{S}}$ and secret key is $D_{\mathbb{S}}$. Then $\mathcal{C}$ signcrypts m and sends the result to $\mathcal{A}$.*
   - ***Unsigncryption query:*** *$\mathcal{A}$ produces a set of users $\mathcal{U}$, a receiver identity $ID_{\mathbb{R}}$, and a ciphertext C. The challenger $\mathcal{C}$ generates the private key $D_{\mathbb{R}} = Keygen(ID_{\mathbb{R}})$ and retrieves m from C and verifies the validity of the ciphertext. If it gets verified then return m to $\mathcal{A}$ else return $\perp$.*

   $\mathcal{A}$ 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 $\{m_0, m_1\} \in \mathcal{M}$, a set of n users $\mathcal{U}$ and an identity $ID_{\mathbb{R}}$ on which he wants to be challenged and produces them to $\mathcal{C}$. $\mathcal{A}$ should not have queried the private key corresponding to any user in the group $\mathcal{U}$ or even $ID_{\mathbb{R}}$ in the first stage. $\mathcal{C}$ now chooses a bit $b \in_R \{0, 1\}$ and computes the challenge ciphertext C^* of m_b, which is sent to $\mathcal{A}$.*
4. ***Second Phase:*** *$\mathcal{A}$ performs polynomially bounded number of requests just like those in the first stage, with the restrictions that it cannot make Key Extraction queries on any user in the group $\mathcal{U}$ or even $ID_{\mathbb{R}}$ and should not query for unsigncryption query on C^*.*
5. ***Guess:*** *Finally, $\mathcal{A}$ produces a bit b' and wins the game if $b' = b$. The adversary's success probability is defined as:*

$$Succ_{\mathcal{A}}^{IND-IRSC-CCA2}(\kappa) = \frac{1}{2} + \epsilon$$

We require that ϵ to be negligible with respect to κ and ϵ is called the advantage for the adversary in the attack.

Definition 4. *An identity based ring signcryption scheme (IRSC) is said to be existentially unforgeable against adaptive chosen messages attacks (EUF-IRSC-CMA) if no polynomially bounded adversary has a non-negligible advantage in the following game:*

1. ***Setup Phase:*** *The challenger runs the Setup algorithm with a security parameter κ and gives the system parameters to the adversary $\mathcal{A}$.*

2. ***Training Phase:*** *$\mathcal{A}$ performs a polynomially bounded number of queries as described in First Phase of definition 3. The queries may be adaptive, i.e. the current query may depend on the previous query responses.*
3. ***Forgery:*** *Finally, $\mathcal{A}$ produces a new triple $(\mathcal{U}, ID_{\mathbb{R}}, C)$ (i.e. a triple that was not produced by the signcryption oracle), where the private keys of the users in the group $\mathcal{U}$ and the receiver (whose identity is $ID_{\mathbb{R}}$) were not queried in the training phase. $\mathcal{A}$ wins the game if the result of the Unsigncryption $(\mathcal{U}, ID_{\mathbb{R}}, C)$ is not $\perp$ symbol, in other words C is a valid signcrypt of some message $m \in \mathcal{M}$. If $\mathcal{A}$ is able to produce a valid signcryption, $\mathcal{C}$ is capable of solving the hard problem instance given to it.*

4 Attacks on Various Ring Signcryption Schemes

This section gives an overview of several schemes and the attacks corresponding to them. First we consider Yu et al.'s [9] anonymous signcryption scheme, followed by Fagen Li et al.'s [4] authenticatable anonymous signcryption scheme, next we take up Lijun et al.'s [14] ring signcryption scheme and conclude this section with the review and attack on Zhu et al.'s [13] scheme.

4.1 Overview of Anonymous Signcryption (ASC) Scheme of Yu et al.

Yu et al.'s ASC scheme [9] consists of four algorithms namely: *Setup*, *KeyGen*, *Signcryption* and *Unsigncryption*, which we describe below.

1. ***Setup***(κ, l)***:*** Here, κ and l are the security parameters.
   (a) The PKG selects $\mathbb{G}_1$, $\mathbb{G}_2$ of same order q and a random generator P of $\mathbb{G}_1$.
   (b) Selects the master private key $s \in_R \mathbb{Z}_q^*$.
   (c) The master public key is set to be $P_{pub} = sP$.
   (d) Selects three strong public one-way hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^l$, $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
   (e) Selects an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
   (f) The public parameters of the scheme are set to be *params*=($\mathbb{G}_1$, $\mathbb{G}_2$, $\hat{e}$, P, P_{pub}, H_1, H_2, H_3,q).

2. ***KeyGen***(ID_i)***:*** Here, ID_i is the identity of the user $\mathcal{U}_i$, The PKG computes the following.
   (a) The user public key is computed as $Q_i = H_1(ID_i)$
   (b) The corresponding private key $D_i = sQ_i$.
   (c) The PKG sends D_i to the user $\mathcal{U}_i$ via a secure authenticated channel.

3. ***Signcryption***$(\mathcal{U}, m, ID_{\mathbb{R}}, ID_{\mathbb{S}}, D_{\mathbb{S}})$***:*** Inorder to signcrypt the message m the sender does the following:
   (a) Chooses $r \in_R \mathbb{Z}_q^*$ and computes $R = rP$, $R' = \hat{e}(P_{pub}, Q_{\mathbb{R}})^r$, $t = H_2(R')$, $c = m \oplus t$.
   (b) For all $i = 1$ to n and $i \neq \mathbb{S}$, chooses $U_i \in_R \mathbb{G}_1$, computes $h_i = H_3(m, t, \mathcal{U}, U_i)$.
   (c) For $i = \mathbb{S}$ chooses $r'_{\mathbb{S}} \in_R \mathbb{Z}_q^*$, computes $U_{\mathbb{S}} = r'_{\mathbb{S}} Q_{\mathbb{S}} - \Sigma_{i=1, i \neq \mathbb{S}}^{n} (U_i + h_i Q_i)$, $h_{\mathbb{S}} = H_3(m, t, \mathcal{U}, U_{\mathbb{S}})$ and $V = (h_{\mathbb{S}} + r'_{\mathbb{S}}) D_{\mathbb{S}}$.

   Finally the sender outputs the ciphertext as $C = (\mathcal{U}, c, R,h_1, \ldots, h_n,U_1, \ldots, U_n, V)$.

4. ***Unsigncrypt***$(C = (\mathcal{U}, c, R, h_1, \ldots, h_n, U_1, \ldots, U_n, V), D_{\mathbb{R}})$***:*** Inorder to unsigncrypt a ciphertext C the receiver does the following:
   (a) Computes $t' = H_2(\hat{e}(R, D_{\mathbb{R}}))$ and $m' = c \oplus t'$.
   (b) For $i = 1$ to n, checks whether $h'_i \overset{?}{=} H_3(m', t', \mathcal{U}, U_i)$.
   (c) Checks whether $\hat{e}(P_{pub}, \Sigma_{i=1}^{n} (U_i + h'_i Q_i)) \overset{?}{=} \hat{e}(P, V)$.

   If the above checks are true for all values of i, then accept m' as the message, otherwise reject the ciphertext.

Attack on ASC Scheme of Yu et al.: During the challenge phase of the confidentiality proof, the challenger $\mathcal{C}$ receives two messages m_0 and m_1 from the adversary $\mathcal{A}$. The challenger chooses $b \in_R \{0,1\}$ and produces the challenge ciphertext C^* using the message m_b and delivers it to $\mathcal{A}$. Upon receipt of $C^* = (\mathcal{U}, c^*, R^*, h_1^*, \ldots, h_n^*, U_1^*, \ldots, U_n^*, V^*)$, $\mathcal{A}$ does the following to distinguish C^*, whether it is a signcryption of m_0 or m_1.

- Since $\mathcal{A}$ knows both messages m_0 and m_1, $\mathcal{A}$ can perform the following computations.
- Computes $t^* = c^* \oplus m_0$ and checks whether $h_i \overset{?}{=} H_3(m_0, t^*, \mathcal{U}, U_i^*)$, for $i = 1$ to n. If the check holds for all values of i, then C^* is the ciphertext corresponding to m_0.
- If the above check does not hold $\mathcal{A}$ computes $t^* = c^* \oplus m_1$, checks whether $h_i \overset{?}{=} H_3(m_1, t^*, \mathcal{U}, U_i^*)$, for $i = 1$ to n. If it holds then C^* is a valid ciphertext for message m_1.
- Atleast one of the checks should hold *true*, else C^* is an invalid ciphertext.

Thus, $\mathcal{A}$ distinguishes the ciphertext with out solving any hard problem. It is not required for $\mathcal{A}$ to interact with the challenger $\mathcal{C}$ after receiving the challenge ciphertext C^* or even ask for any encryption or decryption queries. Thus, our attack is indeed an attack against the CPA security of the ASC scheme by Yu et al. reported in [9].

Remark: Informally, $\mathcal{A}$ is able to distinguish the ciphertext because, the key component required to evaluate the hash value h_i is t' and it is available in $c = m_b \oplus t'$. $\mathcal{A}$ knows that m_b is either m_0 or m_1 because they were produced to $\mathcal{C}$ during the challenge phase by $\mathcal{A}$. Hence, $\mathcal{A}$ can find t' without having access to the private key of the receiver and this led to the proposed attack.

4.2 Overview of Authenticatable Anonymous Signcryption Scheme(AASC) of Fagen Li et al.

The AASC scheme of Fagen Li et al. [4] consists of the five algorithms. A secure symmetric key encryption scheme (E, D) is employed in this scheme where, E and D are the secure symmetric key encryption and decryption algorithms respectively.

1. ***Setup***(κ)***:*** Here, κ is the security parameter.
   (a) The PKG selects $\mathbb{G}_1$, $\mathbb{G}_2$ of same order q and a random generator P of $\mathbb{G}_1$.
   (b) Selects the master private key $s \in_R \mathbb{Z}_q^*$.
   (c) The master public key is set to be $P_{pub} = sP$.
   (d) Selects three strong public one-way hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_1 : \mathbb{G}_2 \rightarrow \{0,1\}^l$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
   (e) Selects an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a secure symmetric cipher (E, D).
   (f) The public parameters of the scheme are set to be *params*=($\mathbb{G}_1$, $\mathbb{G}_2$, $\hat{e}$, P, P_{pub}, H_0, H_1, H_2).
2. ***Extract***(ID_i)***:*** Here, ID_i is the identity of the user $\mathcal{U}_i$, The PKG computes the following.
   (a) The user public key is computed as $Q_i = H_0(ID_i)$
   (b) The corresponding private key $D_i = sQ_i$.
   (c) The PKG sends D_i to the user $\mathcal{U}_i$ via a secure authenticated channel.
3. ***Signcrypt***$(\mathcal{U}, m, ID_{\mathbb{R}}, ID_{\mathbb{S}}, D_{\mathbb{S}})$***:*** Inorder to signcrypt the message m the sender does the following:
   (a) Chooses $r \in_R \mathbb{Z}_q^*$ and computes $R = rP$, $k = H_1(\hat{e}(P_{pub}, Q_{\mathbb{R}})^r)$, $c = E_k(m)$.
   (b) For $i = 1$ to n, $i \neq \mathbb{S}$, chooses $a_i \in_R \mathbb{Z}_q^*$, computes $U_i = a_i P$ and $h_i = H_2(c, \mathcal{U}, U_i)$.
   (c) For $i = \mathbb{S}$, chooses $a_{\mathbb{S}} \in_R \mathbb{Z}_q^*$, computes $U_{\mathbb{S}} = a_{\mathbb{S}} Q_{\mathbb{S}} - \Sigma_{i=1 i \neq \mathbb{S}}^{n} (U_i + h_i Q_i)$.
   (d) Now he computes $h_{\mathbb{S}} = H_2(c, \mathcal{U}, U_{\mathbb{S}})$ and $\sigma = (h_{\mathbb{S}} + a_{\mathbb{S}}) D_{\mathbb{S}}$.
   Finally the sender outputs the ciphertext as $C = (\mathcal{U}, c, R, U_1, \ldots, U_n, \sigma)$.
4. ***Unsigncrypt***$(C = (\mathcal{U}, c, R, U_1, \ldots, U_n, \sigma), D_{\mathbb{R}})$***:*** To unsigncrypt C the receiver does the following.
   (a) Computes $k' = H_1(\hat{e}(R, D_{\mathbb{R}}))$ and recover $m' = D_{k'}(c)$.
   (b) For $i = 1$ to n, computes $h_i' = H_2(c, \mathcal{U}, U_i)$.
   (c) Accepts C and the message m' if and only if $\hat{e}(P_{pub}, \Sigma_{i=1}^{n} (U_i + h_i' Q_i)) \overset{?}{=} \hat{e}(P, \sigma)$, else reject C.
5. ***Authenticate(C):*** The actual sender $ID_{\mathbb{S}}$ can prove that the message m was indeed signcrypted by him by performing the following interaction with the receiver.
   (a) The sender chooses $x \in_R \mathbb{Z}_q^*$, computes $\mu = \hat{e}(P, \sigma)^x$ and sends μ to the verifier.
   (b) The verifier chooses $y \in_R \mathbb{Z}_q^*$ and sends it to the sender.
   (c) The sender computes $v = (x + y)(h_{\mathbb{S}} + a_{\mathbb{S}})$ and returns v to the verifier.
   (d) The verifier checks whether $\hat{e}(P_{pub}, Q_{\mathbb{S}})^v \overset{?}{=} \mu.\hat{e}(P, \sigma)^y$ and accepts if the check holds.

Attack on AASC Scheme of Fagen Li et al.: The attack on AASC scheme is quite tricky one and it follows that the model considered by the authors did not explain explicitly the scenario of the attack we propose. On receiving the challenge ciphertext $C^* = (\mathcal{U}^*, c^*, R^*, U_1^*, \ldots, U_n^*, \sigma^*)$, during the challenge phase in the confidentiality game, $\mathcal{A}$ can access the private keys of users who are not members of $\mathcal{U}^*$ (here, $\mathcal{U}^*$ is the group of ad-hoc members in the challenge ciphertext C^*). Let us consider $\mathcal{U}_E^{'}$ with identity string ID_E for which $\mathcal{A}$ knows the private key D_E. $\mathcal{A}$ performs the following steps to distinguish C^* as, whether it is a singcrypt of m_0 or m_1, during the second phase of oracle queries.

- $\mathcal{A}$ forms a new group with η users who are totally different from $\mathcal{U}^*$, say $\mathcal{U}^{'} = \{\mathcal{U}_1^{'}, \ldots, \mathcal{U}_\eta^{'}\}$ where $\mathcal{U}_E^{'} \in \mathcal{U}^{'}$.
- For $i = 1$ to η, $i \neq E$, $\mathcal{A}$ chooses $a_i \in_R \mathbb{Z}_q^*$, computes $U_i^{'} = a_i P$ and $h_i^{'} = H_2(c^*, \mathcal{U}^{'}, U_i^{'})$.
- For $i = E$, $\mathcal{A}$ chooses $a_E \in_R \mathbb{Z}_q^*$, computes $U_E^{'} = a_E Q_E - \Sigma_{i=1, i \neq E}^{\eta}(U_i^{'} + h_i^{'} Q_i)$.
- $\mathcal{A}$ computes $h_E^{'} = H_2(c^*, \mathcal{U}^{'}, U_E^{'})$ and $\sigma^{'} = (h_E^{'} + a_E) D_E$.
- Now, $C^{'} = (\mathcal{U}^{'}, c^*, R^*, U_1^{'}, \ldots, U_\eta^{'}, \sigma^{'})$ is a valid ring signcryption on m_b, which was chosen by $\mathcal{C}$ to generate C^* and is entirely different from C^*. Thus, $\mathcal{A}$ can legally send C' in the second phase to $\mathcal{C}$ and obtain corresponding m_b.
- $\mathcal{A}$ gets the decryption to $C^{'}$ and retrieves the message and from this concludes correctly whether C^* is a signcryption of m_0 or m_1.

Distinguishing the ciphertext after the start of the second phase of interaction and a decryption query leads to a break in CCA2 security of the system. Thus, we claim that the AASC scheme by Fagen Li et al. [4] is not adaptive chosen ciphertext secure.

Remark: In this scheme, ring signcryption is achieved by using the *Encrypt-then-Sign* paradigm, where the signature part is a ring signature algorithm. This scheme lacks the binding between the encryption and signature; namely, the output of the encryption (in this scheme c and k) is not used as input in the hash of message, which is used for generating the ring signature. Since the aforementioned binding is absent, $\mathcal{A}$ who is in possession of the private key of the users other than the targeted users can generate a valid ciphertext with a new set of users. This newly formed ciphertext has the encryption part same as the challenge ciphertext (i.e. the same message) but with a different signature. Now, $\mathcal{A}$ queries the decryption oracle for the decryption of the newly formed ciphertext, which is not forbidden in the model.

4.3 Overview of Identity Based Ring Signcryption (IRSC) Scheme of Lijun et al.

The IRSC scheme of Lijun et al. [14] consists of the following four algorithms.

1. ***Setup***(κ)***:*** Here, κ is the security parameters.
   (a) The PKG selects $\mathbb{G}_1$, $\mathbb{G}_2$ of same prime order - q and a random generator P of $\mathbb{G}_1$.
   (b) Selects the master private key $s \in_R \mathbb{Z}_q^*$.
   (c) The master public key is set to be $P_{pub} = sP$.
   (d) Selects three cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
   (e) Selects an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
   (f) The public parameters of the scheme are set to be *params*=($\mathbb{G}_1$, $\mathbb{G}_2$, $\hat{e}$, P, P_{pub}, H_1, H_2, H_3,q).
2. ***KeyGen***(ID_i)***:*** For each user $\mathcal{U}_i$, the PKG computes the public key and private key as follows.
   (a) The user public key is computed as $Q_i = H_1(ID_i)$
   (b) The corresponding private key $D_i = sQ_i$.
   (c) The PKG sends D_i to the user $\mathcal{U}_i$ via a secure and authenticated channel.
3. ***Signcrypt***$(\mathcal{U}, m, ID_{\mathbb{R}}, ID_{\mathbb{S}}, D_{\mathbb{S}})$***:*** Inorder to signcrypt the message m the sender does the following:
   (a) Chooses $r_0 \in_R \mathbb{Z}_q^*$ and computes $R_0 = r_0 P$, $W = r_0 P_{pub}$.
   (b) For $i = 1$ to n, $i \neq \mathbb{S}$, chooses $r_i \in_R \mathbb{Z}_q^*$, computes $R_i = r_i P$ $h_i = H_2(m \| \mathcal{U} \| R_i \| R_0)$.
   (c) For $i = \mathbb{S}$, chooses $r_{\mathbb{S}} \in_R \mathbb{Z}_q^*$, computes $R_{\mathbb{S}} = r_{\mathbb{S}} P - \Sigma_{i=1, i \neq \mathbb{S}}^{n}(h_i Q_i)$, $h_{\mathbb{S}} = H_2(m \| \mathcal{U} \| R_{\mathbb{S}} \| R_0)$ and $V = h_{\mathbb{S}} D_{\mathbb{S}} + \Sigma_{i=1}^{n} r_i P_{pub}$.
   (d) Computes $y = \hat{e}(W, Q_{\mathbb{R}})$, $t = H_3(y)$, $c = m \oplus t$.
   Finally the sender outputs the ciphertext as $C = (\mathcal{U}, c, V, R_0, R_1, \ldots, R_n)$.

4. ***Unsigncrypt(****$C = (\mathcal{U}, c, V, R_0, R_1, \ldots, R_n), D_{\mathbb{R}}$****).** In-order to unsigncrypt C the receiver does the following.
   (a) Computes $t' = H_3(\hat{e}(D_{\mathbb{R}}, R_0))$ and recovers $m' = c \oplus t'$.
   (b) For $i = 1$ to n, computes $h'_i = H_2(m \| \mathcal{U} \| R_i \| R_0)$.
   (c) Checks whether $\hat{e}(P_{pub}, \Sigma_{i=1}^{n}(R_i + h'_i Q_i)) \overset{?}{=} \hat{e}(P, V)$.
   If the above check holds then the receiver accepts m' as a valid message. Otherwise, rejects the cipher text C.

Attack on IRSC Scheme of Lijun et al.: During the challenge phase of the confidentiality proof, the challenger $\mathcal{C}$ receives two messages m_0 and m_1 from the adversary $\mathcal{A}$. The challenger chooses $b \in_R \{0, 1\}$ and produces the challenge ciphertext C^* using the message m_b and delivers it to $\mathcal{A}$. Upon receipt of $C^* = (\mathcal{U}, c^*, V^*, R_0^*, R_1^*, \ldots, R_n^*)$, $\mathcal{A}$ does the following to distinguish C^* as, whether it is a signcryption of m_0 or m_1.

- Since $\mathcal{A}$ knows both messages m_0 and m_1, it can perform the following computations.
- $\mathcal{A}$ can compute $h_i = H_2(m_0 \| \mathcal{U} \| R_i^* \| R_0^*)$ for $i = 1$ to n. (since R_i^*, R_0^* are known from the ciphertext).
- Check whether $\hat{e}(P_{pub}, \Sigma_{i=1}^{n}(R_i^* + h_i Q_i)) \overset{?}{=} \hat{e}(P, V^*)$. If this check holds then C^* is a valid signcryption of m_0.
- If the above check does not hold, perform the previous two steps with m_0 replaced by m_1. If the ciphertext was formed with one of the two messages m_0 or m_1, any one check will hold good else the ciphertext C^* is an invalid one.

Thus, the challenge ciphertext C^* is distinguished by $\mathcal{A}$ without solving any hard problem.

Remark: The intuition behind the attack is, the ring signcryption proposed by Lijun et al. [14] can be verified if the message and the corresponding ciphertext is known. During the confidentiality game the adversary $\mathcal{A}$ knows the message, which is either m_0 or m_1, with these information $\mathcal{A}$ concludes whether C^* is a ring signcryption of m_0 or m_1.

4.4 Overview of IRSC Scheme of Zhu et al.

The IRSC scheme of Zhu et al. [10] consists of the following four algorithms.

1. ***Setup***(κ, l)***:*** Here, κ and l are the security parameters.
   (a) The PKG selects $\mathbb{G}_1$, $\mathbb{G}_2$ of same order q and a random generator P of $\mathbb{G}_1$.
   (b) Selects the master private key $s \in_R \mathbb{Z}_q^*$ and sets the master public key to be $P_{pub} = sP$.
   (c) Selects four cryptographic hash functions $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_1^* \rightarrow \{0, 1\}^l$, $H_3 : \{0, 1\}^l \times \mathbb{G}_2 \rightarrow \{0, 1\}^l$, $H_4 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$.
   (d) Selects an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
   (e) The public parameters of the scheme are set to be $params$=($\mathbb{G}_1$, $\mathbb{G}_2$, $\hat{e}$, P, P_{pub}, H_1, H_2, H_3, H_4, q).
2. ***Keygen***(ID_i)***:*** For each user $\mathcal{U}_i$, the PKG computes the private key and public key as follows.
   (a) The user public key is computed as $Q_i = H_0(ID_i)$
   (b) The corresponding private key $D_i = sQ_i$.
   (c) The PKG sends D_i to the user $\mathcal{U}_i$ via a secure authenticated channel.
3. ***Signcrypt***$(\mathcal{U}, m, ID_{\mathbb{R}}, ID_{\mathbb{S}}, D_{\mathbb{S}})$***:*** Inorder to signcrypt the message m the sender does the following:
   (a) Chooses $r \in_R \mathbb{Z}_q^*$, $m^* \in_R \mathcal{M}$ and computes $R_0 = rP$, $R' = \hat{e}(rP_{pub}, Q_{\mathbb{R}})$, $k = H_2(R')$, $c_1 = m^* \oplus k$ and $c_2 = m \oplus H_3(m^* || R_0)$.
   (b) For $i = 1$ to n, $i \neq \mathbb{S}$, chooses $U_i \in_R \mathbb{G}_1^*$, computes $h_i = H_4(c_2 || U_i)$.
   (c) For $i = \mathbb{S}$, chooses $r' \in_R \mathbb{Z}_q^*$, computes $U_{\mathbb{S}} = r'Q_{\mathbb{S}} - \Sigma_{i=1, i \neq \mathbb{S}}^{n}(U_i + h_i Q_i)$, $h_{\mathbb{S}} = H_4(c_2 || U_{\mathbb{S}})$ and $V = (h_{\mathbb{S}} + r')D_{\mathbb{S}}$.
   Finally the sender outputs the ciphertext as $C = (\mathcal{U}, R_0, c_1, c_2, U_1, \ldots, U_n, V)$.
4. ***Unsigncrypt(****$C = (\mathcal{U}, R_0, c_1, c_2, U_1, \ldots, U_n, V)$, $D_{\mathbb{R}}$****):*** To unsigncrypt C, the receiver does the following.
   (a) For $i = 1$ to n, computes $h'_i = H_4(c_2 || U_i)$.
   (b) Checks whether $\hat{e}(P_{pub}, \Sigma_{i=1}^{n}(U_i + h'_i Q_i)) \overset{?}{=} \hat{e}(P, V)$, if so, computes $k' = H_2(\hat{e}(R_0, D_{\mathbb{R}}))$ and recovers $m^* = c_1 \oplus k'$, $m' = c_2 \oplus H_3(m^* || R_0)$ and accept m' as a valid message.
   Note: The actual scheme in [10] had typos in setup as well as signcryption algorithms. Instead of H_2 it was written H_1 and instead of $U_{\mathbb{S}} = r'Q_{\mathbb{S}} - \Sigma_{i=1, i \neq \mathbb{S}}^{n}(U_i + h_i Q_i)$, it was written $U_{\mathbb{S}} = r'Q_{\mathbb{S}} - \Sigma_{i=1, i \neq \mathbb{S}}^{n}(U_i + h_i Q_{\mathbb{S}})$. We have corrected them in our review, inorder to maintain the consistency of the scheme.

Attack on IRSC Scheme of Zhu et al.: During the confidentiality game, the adversary $\mathcal{A}$ has access to private keys of users who are not members of the group $\mathcal{U}^*$, which is the group of members in the challenge ciphertext. Let us consider $\mathcal{U}_E$ with identity string ID_E as one such user, so $\mathcal{A}$ is allowed to query the private key D_E of $\mathcal{U}_E$. On receiving the challenge ciphertext, $C^* = (\mathcal{U}^*, R_0^*, c_1^*, c_2^*, U_1^*, \ldots, U_n^*, V^*)$ during the challenge phase, $\mathcal{A}$ performs the following steps to distinguish C^* as, whether it is a singcrypt of m_0 or m_1.

- $\mathcal{A}$ forms a new group with η members who are totally different from $\mathcal{U}^*$, say $\mathcal{U}^{'} = \{\mathcal{U}_1^{'}, \ldots, \mathcal{U}_\eta^{'}\}$ where $\mathcal{U}_E^{'} \in \mathcal{U}^{'}$.
- Chooses a message m' and computes $c_2^{'} = c_2^* \oplus m'$.
- For all $i = 1$ to η and $i \neq E$, chooses $U_i^{'} \in_R \mathbb{G}_1^*$ and computes $h_i^{'} = H_4(c_2^{'}||U_i)$.
- For $i = E$, chooses $r^{'} \in_R \mathbb{Z}_q^*$ and computes $U_E^{'} = r^{'}Q_A - \Sigma_{i=1}^{\eta}(U_i^{'} + h_i^{'}Q_i)$.
- Computes $h_E^{'} = H_4(c_2^{'}||U_E^{'})$ and $V^{'} = (r^{'} + h_E^{'})D_E$
- Now, $C^{'} = (\mathcal{U}^{'}, R_0^*, c_1^*, c_2^{'}, U_1^{'}, \ldots, U_n^{'}, V^{'})$ is a valid ciphertext on message $m_b \oplus m'$.

Now, during the second phase of training, $\mathcal{A}$ requests the unsigncryption of C' to $\mathcal{C}$. Note that it is legal for $\mathcal{A}$ to ask for unsigncryption of C' because it is derived from C^* and not exactly the challenge ciphertext C^*. $\mathcal{C}$ responds with $M = m_b \oplus m'$ as the output for the query. $\mathcal{A}$ now obtains $m_b = M \oplus m'$ and thus identifies the message in the challenge ciphertext C^*.

Remark: This attack is possible due to the same reason as described in the remark for the attack stated in section 4.2.

5 New Ring Signcryption Scheme (New-IBRSC)

In this section, we present a new improved identity based ring signcryption scheme (New-IBRSC), taking into account the attacks carried out in the previous section. New-IBRSC consists of the following four algorithms:

1. ***Setup***(κ)***:*** This algorithm is executed by the PKG to initialize the system by taking a security parameter κ as input.
   - It selects $\mathbb{G}_1$ an additive group and $\mathbb{G}_2$ a multiplicative group, both cyclic with same prime order - q and a random generator P of the group $\mathbb{G}_1$.
   - It selects $s \in_R \mathbb{Z}_q^*$ as the master private key and sets $P_{pub} = sP$ as the master public key.
   - Selects three cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^{|\mathcal{M}|} \times \mathbb{G}_1, H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.
   - It also selects a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the appropriate properties as specified in section 2.
   - The public parameters of the scheme are set to be *params*=($\mathbb{G}_1$, $\mathbb{G}_2$, $\hat{e}$, P, P_{pub}, H_1, H_2, H_3, q).
2. ***Keygen***(ID_i)***:*** This algorithm takes ID_i, the identity of an 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:
   - The public key is computed as $Q_i = H_1(ID_i)$
   - The corresponding private key $D_i = sQ_i$.
   - The PKG sends D_i to the user $\mathcal{U}_i$ via a secure and authenticated channel.
3. ***Signcrypt***$(\mathcal{U}, m, ID_{\mathbb{R}}, Q_{\mathbb{R}}, ID_{\mathbb{S}}, D_{\mathbb{S}})$***:*** Signcrypting a message m is done by the sender $\mathcal{U}_{\mathbb{S}}$ with private key $D_{\mathbb{S}}$ and public key $Q_{\mathbb{S}}$, to the receiver $\mathcal{U}_{\mathbb{R}}$ with public key $Q_{\mathbb{R}}$ as explained below:
   - $\mathcal{U}_{\mathbb{S}}$ selects n potential senders and forms an ad-hoc group $\mathcal{U}$, including its own identity $ID_{\mathbb{S}}$ also.
   - Chooses $r \in_R \mathbb{Z}_q^*$, computes $U = rP$ and $\alpha = \hat{e}(P_{pub}, Q_{\mathbb{R}})^r$.
   - For $i = 1$ to n, $i \neq \mathbb{S}$, chooses $U_i \in_R \mathbb{G}_1$ and computes $h_i = H_3(m, U_i, \alpha, \mathcal{U}, Q_{\mathbb{R}})$.
   - For $i = \mathbb{S}$, chooses $r_{\mathbb{S}} \in_R \mathbb{Z}_q^*$, computes $U_{\mathbb{S}} = r_{\mathbb{S}}Q_{\mathbb{S}} - \Sigma_{i=1, i\neq\mathbb{S}}^{n}(U_i + h_iQ_i)$, $h_{\mathbb{S}} = H_3(m, U_{\mathbb{S}}, \alpha, \mathcal{U}, Q_{\mathbb{R}})$ and $V = (h_{\mathbb{S}} + r_{\mathbb{S}})D_{\mathbb{S}}$.
   - Computes $y = (m\|V) \oplus H_2(\alpha)$.

   Finally the sender outputs the ciphertext as $C = (y, \mathcal{U}, U, U_1, \ldots, U_n)$.
4. ***Unsigncrypt(***$C = (y, \mathcal{U}, U, U_1, \ldots, U_n), D_{\mathbb{R}}$***):*** To unsigncrypt C the receiver $\mathcal{U}_{\mathbb{R}}$ with identity $ID_{\mathbb{R}}$ does the following:

- Computes $\alpha' = \hat{e}(U, D_{\mathbb{R}})$ and retrieves the message m' and signature V' as $(m' \| V') = y \oplus H_2(\alpha')$.
- For $i = 1$ to n, computes $h'_i = H_3(m', U_i, \alpha', \mathcal{U}, Q_{\mathbb{R}})$ and checks whether $\hat{e}(P_{pub}, \Sigma_{i=1}^{n}(U_i + h'_i Q_i)) \overset{?}{=} \hat{e}(P, V')$.

If the above check holds, then the receiver $\mathcal{U}_{\mathbb{R}}$ accepts C as a valid ciphertext and the message m' as a valid message.

6 Security Results for New-IBRSC:

The anonymity proof for the new identity based ring signcryption scheme (New-IBRSC) follows from the underlying identity based signature. The composition of encryption and ring signature scheme to form the ring signcryption scheme (New-IBRSC) does not induce a weakness in the anonymity property because the encryption does not include any components that are related to the ring signature. The binding between the encryption and the ring signature is obtained with the help of the session key that is used for encrypting the message. Even though, the session key is an input to the message hash in the ring signature it does not containg any value related to the identity of the sender or that reveals the sender's identity and hence forth we concentrate only on the security against adaptive chosen ciphertext attack (CCA2) and security against chosen message attack (CMA). We formally prove the security of the new identity based ring signcryption scheme (New-IBRSC), indistinguishable under chosen ciphertext attack (IND-New-IBRSC-CCA2) and existentially unforgeable under chosen message and identity attack (EUF-New-IBRSC-CMA) in the random oracle model. We consider the security model given in section 3 to prove the security of our New-IBRSC.

6.1 Confidentiality Proof of New-IBRSC (IND-New-IBRSC-CCA2):

Theorem 1. *The new identity based ring signcryption scheme is secure against any IND-New-IBRSC-CCA2 adversary $\mathcal{A}$ under the random oracle model if DBDHP is hard in $\mathbb{G}_1$.*

Proof: The challenger $\mathcal{C}$ is challenged to solve an instance of the DBDHP, and $\mathcal{C}$ in-turn makes use of an adversary $\mathcal{A}$ which is capable of breaking with non-negligible advantage the IND-New-IBRSC-CCA2 security of our new scheme New-IBRSC, to solve the instance. In other words, $\mathcal{A}$ guesses the bit b' during the *Guess* phase (specified in **Definition 3.**) with non-negligible advantage. On receiving the instance $\langle P, aP, bP, cP, \beta \rangle \in \mathbb{G}_1^4 \times \mathbb{G}_2$ of the DBDHP as input, $\mathcal{C}$ aims to decide whether $\beta \overset{?}{=} \hat{e}(P, P)^{abc} \in \mathbb{G}_2$. $\mathcal{C}$ simulates the system with the various oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{Signcryption}$, $\mathcal{O}_{Unsigncryption}$ and allows $\mathcal{A}$ to adaptively ask polynomially bounded queries to these oracles. The queries are handled by $\mathcal{C}$ as described in the *First Phase* below.

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

- It takes $\mathbb{G}_1$ and $\mathbb{G}_2$, the two groups as well as the generator $P \in \mathbb{G}_1$ as given in DBDHP instance.
- Sets the master public key $P_{pub} = aP$.
- Models the three hash functions as random oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$ and $\mathcal{O}_{H_3}$.
- Selects a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

First Phase: $\mathcal{A}$ adaptively (means that, the input to the current query may depend on the outputs obtained for the previous queries) performs polynomially bounded number of queries to the various oracles in the first stage which are handled by $\mathcal{C}$ as given below.

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

Extract Query: This query is answered by $\mathcal{C}$ by just choosing an integer $b_i \in_R \mathbb{Z}_q^*$ and setting the public key of ID_i as $Q_i = b_i P$, adds the tuple $\langle ID_i, b_i, Q_i \rangle$ to the list L_1, if ID_i is not the target identity. If ID_i is the target identity, $\mathcal{C}$ sets $Q_i = bP$. (*Note:* All b_i's are distinct and $b_i \neq b$ for any i also, bP is obtained from the CDHP instance.)

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

$\mathcal{C}$ proceeds with the calling of signcryption as specified in the algorithm when:

- Both sender identity $ID_{\mathbb{S}}$ and receiver identity $ID_{\mathbb{R}}$ are not the target identity or,
- The sender identity $ID_{\mathbb{S}}$ is not a target identity and the receiver identity $ID_{\mathbb{R}}$ is the target identity.

In these cases it is possible for $\mathcal{C}$ to produce the ciphertext because it knows the private key of the sender $D_{\mathbb{S}}$, which is not the target identity.

If the sender's identity $ID_{\mathbb{S}}$ is the target identity (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 $r \in_R \mathbb{Z}_q^*$ and sets $U = rP$ and $\alpha = \hat{e}(P_{pub}, Q_{\mathbb{R}})^r$.
- For $i = 1$ to n, $i \neq \mathbb{S}$, chooses $U_i \in_R \mathbb{G}_1$ and queries the oracle $\mathcal{O}_{H_3}$ and obtains the value $h_i^{(3)} = \mathcal{O}_{H_3}(m, U_i, \alpha, \mathcal{U}, Q_{\mathbb{R}})$.
- For $i = \mathbb{S}$, chooses $r_{\mathbb{S}}, h_{\mathbb{S}}^{(3)} \in_R \mathbb{Z}_q^*$, computes $U_{\mathbb{S}} = r_{\mathbb{S}}P - h_{\mathbb{S}}^{(3)}Q_{\mathbb{S}} - \Sigma_{i=1, i\neq\mathbb{S}}^{n}(U_i + h_i^{(3)}Q_i)$, adds the tuple $\langle m, U_{\mathbb{S}}, \alpha, \mathcal{U}, Q_{\mathbb{R}}, h_{\mathbb{S}}^{(3)}\rangle$ to the list L_3 and computes $V = r_{\mathbb{S}}P_{pub}$ (***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 $h_{\mathbb{S}}^{(3)} = \mathcal{O}_{H_3}(m, U_{\mathbb{S}}, \alpha, \mathcal{U}, Q_{\mathbb{R}})$. This is possible because the random oracles are manipulated by $\mathcal{C}$).
- Queries $h^{(2)} = \mathcal{O}_{H_2}(\alpha)$ and computes $y = (m\|V) \oplus h^{(2)}$

Finally $\mathcal{C}$ outputs the ciphertext $C = (y, \mathcal{U}, U, U_1, \ldots, U_n)$ to $\mathcal{A}$ as the signcryption of m.

$\mathcal{O}_{Unsigncryption}$ *Query:* Upon receiving an unsigncryption query on a ciphertext $C = (y, \mathcal{U}, U, U_1, \ldots, U_n)$ and a receiver identity $ID_{\mathbb{R}}$ both chosen by $\mathcal{A}$, $\mathcal{C}$ proceeds as follows:

$\mathcal{C}$ proceeds with the calling of unsigncryption as specified in the algorithm when:

- Both sender identity $ID_{\mathbb{S}}$ and receiver identity $ID_{\mathbb{R}}$ are not the target identity or,
- The receiver identity $ID_{\mathbb{R}}$ is not a target identity and the sender identity $ID_{\mathbb{S}}$ is the target identity.

It is possible for $\mathcal{C}$ to use the unsigncryption algorithm directly because, $\mathcal{C}$ knows the private key of the receiver $D_{\mathbb{R}}$, which is not the target identity.

If the receiver identity $ID_{\mathbb{R}}$ is the target identity (i.e. When $\mathcal{C}$ does not know the private key corresponding to $ID_{\mathbb{R}}$), $\mathcal{C}$ cooks up the response as explained below:

- $\mathcal{C}$ does a tuple wise check in the list L_3 for each value of i (where $i = 1$ to n) to check whether a tuple of the form $\langle ., U_i, ., \mathcal{U}, Q_{\mathbb{R}}\rangle$ exists (This occurs with a very high probability). $\mathcal{C}$ retrieves the corresponding α value from the tuple.
- Retrieves the message m and the signature V as $m\|V = y \oplus H_2(\alpha)$.
- For $i = 1$ to n, $\mathcal{C}$ queries the oracle $\mathcal{O}_{H_3}$ and obtains the value $h_i^{(3)'} = \mathcal{O}_{H_3}(m, U_i, \alpha, \mathcal{U}, Q_{\mathbb{R}})$ and checks whether $\hat{e}(P_{pub}, \Sigma_{i=1}^{n}(U_i + h_i^{(3)'}Q_i)) \stackrel{?}{=} \hat{e}(P, V)$. If it holds then $\mathcal{C}$ outputs m else outputs $\perp$.

Challenge Phase: Finally, $\mathcal{A}$ chooses two plaintexts m_0, $m_1 \in \mathcal{M}$, a set of users $\mathcal{U}^*$ and an identity $ID_{\mathbb{R}}$ on which he wants to be challenged and produces them to the challenger $\mathcal{C}$. $\mathcal{A}$ should not have queried the private key corresponding to any user in the group $\mathcal{U}^*$ or even $ID_{\mathbb{R}}$ in the first phase. $\mathcal{C}$ now chooses a bit $b \in_R \{0, 1\}$ and computes the challenge ciphertext C^* of m_b as follows and after that sends C^* to $\mathcal{A}$.

- $\mathcal{C}$ selects a sender identity $ID_{\mathbb{S}}$ from $\mathcal{U}^*$, sets $U^* = cP$ and $\alpha = \beta$. Recall that cP and β are the part of the DBDHP instance $\mathcal{C}$ has received.
- For $i = 1$ to n, $i \neq \mathbb{S}$, chooses $U_i^* \in_R \mathbb{G}_1$ and queries the oracle $\mathcal{O}_{H_3}$ and obtains the value $h_i^{(3)} = \mathcal{O}_{H_3}(m, U_i, \alpha, \mathcal{U}, Q_{\mathbb{R}})$.

- For $i = \mathbb{S}$, chooses $r_{\mathbb{S}} \in_R \mathbb{Z}_q^*$, computes $U_{\mathbb{S}}^* = r_{\mathbb{S}} Q_{\mathbb{S}} - \Sigma_{i=1, i \neq \mathbb{S}}^{n} (U_i + h_i Q_i)$, $h_{\mathbb{S}} = H_3(m, U_{\mathbb{S}}^*, \alpha, \mathcal{U}, Q_{\mathbb{R}})$ and $V^* = (h_{\mathbb{S}} + r_{\mathbb{S}}) D_{\mathbb{S}}$.
- Queries the oracle $\mathcal{O}_{H_2}$ and obtains the value $h^{(2)} = \mathcal{O}_{H_2}(\alpha)$ and computes $y^* = (m \| V) \oplus h^{(2)}$

Now, $C^* = (y^*, \mathcal{U}^*, U^*, U_1^*, \ldots, U_n^*)$ is a valid ciphertext on m, which is passed on to $\mathcal{A}$.

Second Phase: $\mathcal{A}$ performs a polynomially bounded number of requests again just like in the first phase. This time, it is not given access to the secret key of any user in the ad-hoc group $\mathcal{U}^*$ nor $ID_{\mathbb{R}}$ and it is prevented from querying the decryption oracle for the ciphertext C^*. Moreover, $\mathcal{A}$ performs the queries in adaptive fashion.

Guess: At the end of the second phase, $\mathcal{A}$ produces a bit b' to $\mathcal{C}$. If $b' = b$ then the tuple $\langle m_b', U_i, \alpha, \mathcal{U}, Q_{\mathbb{R}}, h_i^{(3)} \rangle$ in the list L_3 contains the value $\hat{e}(P, P)^{abc}$ as α in it. This is because during the signcryption of the challenge message m_b, the challenger set the value of α to be β and while unsigncrypting C^*, $\mathcal{A}$ should have computed α and queried the random oracle $\mathcal{O}_{H_3}()$ with α as a parameter. Thus, $\alpha = \hat{e}(P_{pub}, Q_{\mathbb{R}})^r = \hat{e}(aP, bP)^c$ was queried during unsigncryption. So, if $b = b'$, $\mathcal{C}$ outputs the answer to DBDHP as $true$ else returns $false$ and thus solves the DBDHP instance given to it. The probability that $\mathcal{C}$'s answer to the DBDHP is correct, is same as the probability that $b' = b$ and this implies that C can solve DBDHP with non-negligible advantage and this is a contradiction. □

6.2 Unforgeability Proof of New-IBRSC (EUF-New-IBRSC-CMA):

Theorem 2. *Our new identity based ring signcryption scheme is secure against any EUF-New-IBRSC-CMA adversary $\mathcal{A}$ under the random oracle model if CDHP is hard in $\mathbb{G}_1$.*

Proof: The challenger $\mathcal{C}$ is challenged to solve an instance of the CDHP and $\mathcal{C}$ in-turn interacts with an adversary $\mathcal{A}$ which is capable of breaking the EUF-New-IBRSC-CMA security of our new scheme New-IBRSC, to solve the instance. On receiving the instance $\langle P, aP, bP \rangle \in \mathbb{G}_1^3$ of the CDHP as input, $\mathcal{C}$ aims to compute the value $abP \in \mathbb{G}_1$. $\mathcal{C}$ simulates the system with the various oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$, $\mathcal{O}_{H_3}$, $\mathcal{O}_{Signcryption}$, $\mathcal{O}_{Unsigncryption}$ 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.

- It takes $\mathbb{G}_1$ and $\mathbb{G}_2$, the two groups as well as the generator $P \in \mathbb{G}_1$ as given in CDHP instance.
- Sets the master public key $P_{pub} = aP$.
- Models the three hash functions as random oracles $\mathcal{O}_{H_1}$, $\mathcal{O}_{H_2}$ and $\mathcal{O}_{H_3}$.
- Selects a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

Training Phase: $\mathcal{A}$ adaptively performs polynomially bounded number of queries, namely the *Hash Query*, *Extract Query*, $\mathcal{O}_{Signcryption}$*Query* and $\mathcal{O}_{Unsigncryption}$*Query*, to the various oracles in the training phase which are handled by $\mathcal{C}$ as in the confidentiality game for New-IBRSC.

Forgery: Finally, $\mathcal{A}$ produces a forged signcryption $C^* = (y^*, \mathcal{U}^*, U^*, U_1^*, \ldots, U_n^*)$ 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 $\mathcal{U}^*$ and the receiver $ID_{\mathbb{R}}$ were not queried in the training phase. $\mathcal{C}$ can very well designcrypt and verify the validity of the forged ring signcryption C^* because $\mathcal{C}$ knows the secret key of the receiver $ID_{\mathbb{R}}$.

If the forged ring signcryption passes the signature verification then $\mathcal{C}$ will be able to generate two valid ring signcryptions $C^* = (y^*, \mathcal{U}^*, U^*, U_1^*, \ldots, U_n^*)$ and $C' = (y', \mathcal{U}^*, U^*, U_1^*, \ldots, U_n^*)$ using the oracle replay technique and applying extended version of forking lemma [2] applicable for ring signatures. This is achieved by running the same turing machine with the same random tape but with the different hash oracle and allowing $\mathcal{A}$ to interact. Obviously, $\mathcal{A}$ who is capable of generating a valid ring signcryption will be able to generate new valid ring signcryption again with the same randomness. On getting two valid ring signcryptions on m^*, $\mathcal{C}$ will be able to retrieve $D_{\mathbb{S}} = abP$ as explained below:

- Computes $\alpha = \hat{e}(U, D_{\mathbb{R}})$
- Consecutively, V^* and V' are retrieved as $(m^* \| V^*) = y^* \oplus H_2(\alpha)$ and $(m^* \| V') = y' \oplus H_2(\alpha)$.

- Here, $V^* = (h^*_{\mathbb{S}} + r_{\mathbb{S}})D_{\mathbb{S}}$ and $V' = (h'_{\mathbb{S}} + r_{\mathbb{S}})D_{\mathbb{S}}$ (Since they have the same randomness).
- Thus, $V^* - V' = h^*_{\mathbb{S}}D_{\mathbb{S}} - h'_{\mathbb{S}}D_{\mathbb{S}}{=}(h^*_{\mathbb{S}} - h'_{\mathbb{S}})D_{\mathbb{S}}$.

Since $\mathcal{C}$ knows both the hash values $h^*_{\mathbb{S}}$ and $h'_{\mathbb{S}}$, $\mathcal{C}$ can compute $D_{\mathbb{S}}$ as $D_{\mathbb{S}} = (h^*_{\mathbb{S}} - h'_{\mathbb{S}})^{-1}(V^* - V')$. This means, $\mathcal{C}$ can compute abP because $D_{\mathbb{S}} = abP$. In other words, $\mathcal{C}$ is capable of solving CDHP and this is a contradiction. □

7 Conclusion

As a concluding remark we summarize the work in this paper. Ring signcryption is a primitive which enables an user to transmit authenticated messages anonymously and confidentially. To the best of our knowledge there were seven ring signcryption schemes in the identity based setting. Already it was shown in [7] that [5] was not CCA2 secure and in [4] it was shown by Fagen Li et al. that, [11] was not CCA2 secure. So, five out of seven identity based ring signcryption schemes were believed to be secure till date. We have shown that [9] and [14] does not even provide security against chosen plaintext attack (CPA); [4] and [13] does not provide security against adaptive chosen ciphertext attack (CCA2), by demonstrating attacks on confidentiality of these schemes. This leaves Huang et al.'s [3] scheme as the only secure identity based ring signcryption scheme. We have proposed a new identity based ring signcryption scheme for which we proved the security against chosen ciphertext attack and existential unforgeability in the random oracle model. Also we have compared our scheme with Huang et al.'s scheme below. In the comparison table, n represents the number of members in the ring.

Scheme	Pairing Required		Ciphertext Size
	Signcryption	*Unsigncryption*	
Our New-IBRSC	1	3	$n + 2$
Huang et al. [3]	$n + 2$	3	$2n + 3$

Table 1: Efficiency Comparison with [3]

Thus, our new identity based ring signcryption scheme (New-IBRSC) is a significant improvement over the scheme proposed by Huang et al. [3]

References

1. Joonsang Baek, Ron Steinfeld, and Yuliang Zheng. 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, 2002.
2. Javier Herranz and Germán Sáez. Forking lemmas for ring signature schemes. In *INDOCRYPT*, volume 2904 of *Lecture Notes in Computer Science*, pages 266–279. Springer, 2003.
3. Xinyi Huang, Willy Susilo, Yi Mu, and Futai Zhang. 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, 2005.
4. Fagen Li, Masaaki Shirase, and Tsuyoshi Takagi. Analysis and improvement of authenticatable ring signcryption scheme. In *International Conference ProvSec-08, Paper appears in Journal of Shanghai Jiaotong University (Science)*, volume 13, pages 679–683, December 2008.
5. Fagen Li, Hu Xiong, and Yong Yu. An efficient id-based ring signcryption scheme. In *International Conference on Communications, Circuits and Systems, 2008. ICCCAS 2008.*, pages 483–487, May 2008.
6. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.
7. Sree Vivek S, Sharmila Deva Selvi S, and Pandu Rangan C. Cryptanalysis of ring signature and ring signcryption schemes. Cryptology ePrint Archive, Report 2009/052, 2009.
8. Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84: Proceedings of the 4th Annual International Cryptology Conference*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.
9. Yong Yu, Fagen Li, Chunxiang Xu, and Ying Sun. An efficient identity-based anonymous signcryption scheme. *Wuhan University Journal of Natural Sciences*, Volume: 13, Number: 6, December, 2008:670–674, 2008.

10. Tzer Shyong Chen Yu Fang Chung, Zhen Yu Wu. Ring signature scheme for ecc-based anonymous signcryption. In *Computer Standards & Interfaces Journal*, 2008.
11. Mingwu Zhang, Bo Yang, Shenglin Zhu, and Wenzheng Zhang. 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, 2008.
12. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *CRYPTO-97*, pages 165–179, 1997.
13. ZhenChao Zhu, Yuqing Zhang, and Fengjiao Wang. An efficient and provable secure identity based ring signcryption scheme. In *Computer Standards & Interfaces*, Pages 649-654, http://dx.doi.org/10.1016/j.csi.2008.09.023, 2008.
14. Lijun Zhun and Futai Zhang. Efficient idbased ring signature and ring signcryption schemes. In *International Conference on Computational Intelligence and Security, 2008. CIS '08.*, volume 2, pages 303–307, December 2008.