Provably Secure Identity-Based Aggregate Signcryption Scheme in Random Oracles

Jayaprakash Kar

Information Security Research Group
Faculty of Computing & Information Technology
Department of Information Systems,
King Abdulaziz University, P.O.Box-80221, Jeddah-21589,
Kingdom of Saudi Arabia
{jayaprakashkar, jpkar.crypto}@yahoo.com

Abstract. This article proposes a provably secure aggregate signcryption scheme in random oracles. Security of the scheme is based on computational infesibility of solving Decisional Bilinear Diffie-Hellman Problem and Discrete Logarithm Problems. Confidentiality and authenticity are two fundamental security requirement of Public key Cryptography. These are achieved by encryption scheme and digital signatures respectively. Signcryption scheme is a cryptographic primitive that performs signature and encryption simultaneously in a single logical steps. An aggregate signcryption scheme can be constructed of the aggregation of individual signcryption. The aggreagtion is done taking n distinct signcryptions on n messages signed by n distinct users.

Keywords: Bilinear Pairing, BDHP, aggregate signature, DDHP, Random Oracle Model

1 Introduction

Signcryption, first proposed by Zheng [4], is a cryptographic primitive that performs signature and encryption simultaneously in a one logical steps at lower computational costs and communication overheads than those required by the traditional sign-then-encrypt approach. Due to its advantages, there have been many signcryption schemes proposed after Zheng's publication. Zheng's original schemes were only proven secure by Baek *et al.* [1] who described a formal security model in a multiuser setting. Confidentiality, integrity, authentication, and non-repudiation are the important security requirements for many cryptographic applications. Applications must often contain at least two cryptographic primitives: signcryption, signature, and encryption, which will definitely increase the corresponding computation and implementation complexity and even will be infeasible in some resources-constrained environments such as embedded systems, sensor networks and ubiquitous computing. Motivated by this, in 2006, Han *et al.* [8]proposed the concept of Generalized signcryption which can implement the separate or joint encryption and signature functions in a single primitive.

Identity-based cryptography(IBC) was introduced by Shamir (1984). The unique property of identity-based cryptography is that a user's public key can be any binary string, such as an email address, IP address, that can identify the user. This removes the need for senders to look up the recipient's public key before sending out an encrypted message. Identity-based cryptography is supposed to provide a more convenient alternative to conventional public key infrastructure(PKI). Several practical identity-based signature schemes have been devised since 1984, but a satisfying identity-based encryption (IBE) scheme only appeared in 2001 (Boneh and Franklin, 2001). It was proposed by Boneh and Franklin uses bilinear maps (Weil or Tate

pairing) over super singular elliptic curves. Subsequently, several ID-based signcryption schemes have been proposed [5] [6] [7]. The most important practical benefit of IBC is in greatly reducing the need for and trust on the public key certificates.

2 Previous Works

In 1997, Zheng [4] introduced the concept of signcryption where both these properties are achieved in a single logical step, but in a more efficient way. The notion of identity based cryptography was introduced by Shamir [9] in 1984. It is a form of public key cryptography in which the users do not obtain certificates for their public keys. Instead, public keys are generated using arbitrary identifiers such as email addresses, telephone numbers and social security numbers that uniquely identifies a user in the system [3]. The private keys corresponding to the public keys are generated by a trusted authority called Private Key Generator (PKG). The first fully practical identity based encryption scheme was proposed by Boneh and Franklin [10] in 2001. Malone-Lee [11] proposed the first identity based signcryption scheme.

Yu *et al.* [16] proposed the first ID based signcryption scheme in the standard model. But it was proved that, are insecure [17] [15]. Also later on the schemes [16] [15] have shown these are insecure.

In 2002, Malone-Lee [11] proposed an efficient IBSC scheme combining the concepts of identity-based cryptography and signcryption. Libert and Quisquater [19] proved that Malone-Lee's scheme is not semantically secure because the signature of the message is visible in the signcrypted message. subsequently, Libert and Quisquater also proposed three different types of IBSC schemes which satisfy either public verifiability or forward security. It became the an open challenge to design an efficient signcryption scheme providing both public verifiability and forward security. Soon, this open problem was solved. Chow *et al.* [20] designed an IBSC scheme that provides both public verifiability and forward security. Boyen [21] presented an IBSC scheme that provides not only public verifiability and forward security but also ciphertext unlinkability and anonymity. Chen and Malone-Lee [22] improved Boyen's scheme in efficiency and Barreto *et al.* [13] constructed the most efficient IBSC scheme to date.

3 Preliminaries

3.1 Notation

Definition 1. Bilinearity *Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of same prime order q. $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ is a multiplicative group. Let e be a computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, which satisfies the following properties:*

- **Bilinear***: $e(aP, bQ) = e(P, Q)^{ab}$, where $P, Q \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, i.e for $P, Q, R \in \mathbb{G}_1, e(P + Q, R) = e(P, R)e(Q, R)$.*
- **Non-degenerate***: If P is a generator of $\mathbb{G}_1$, then $e(P, P)$ is generator of $\mathbb{G}_2$. There exists $P, Q \in \mathbb{G}$ such that $e(P, Q) \neq 1_{\mathbb{G}_2}$*
- **Computability***: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.*

We call such a bilinear map e is an admissible bilinear pairing and Weil or Tate pairing in elliptic curve can give a good implementation of the admissible bilinear pairing.

3.2 Mathematical Assumption

Definition 2. Decision Diffie-Hellman Problem (DDHP)*: For $a, b, c \in \mathbb{Z}_q^*$, given P, aP, bP, cP decide whether $c \equiv ab \bmod q$*

Definition 3. Computational Diffie-Hellman Problem (CDHP)*: For $a, b \in \mathbb{Z}_q^*$, given P, aP, bP compute abP.*

Definition 4. Bilinear Diffie-Hellman Problem*: Let $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$ be a 5-tuple generated by $\mathcal{G}(k)$, and let $a, b, c \in \mathbb{Z}_q^*$. The BDHP in $\mathbb{G}$ is as follows: Given (P, aP, bP, cP) with $a, b, c \in \mathbb{Z}_q^*$, compute $e(P, P)^{abc} \in \mathbb{G}_T$. The (t, ϵ)-BDH assumption holds in $\mathbb{G}$ if there is no algorithm $\mathcal{A}$ running in time at most t such that*

$$\mathbf{Adv}_{\mathbb{G}}^{BDH}(\mathcal{A}) = \ Pr[\mathcal{A}(P, aP, bP, cP) = \ e(P, P)^{abc}] \geq \epsilon$$

where the probability is taken over all possible choices of (a, b, c). Here the probability is measured over random choices of $a, b, c \in \mathbb{Z}_q^*$ and the internal random operation of A. We assume that BDHP is a hard computational problem: letting q have the magnitude $2k$ where k is a security parameter, there is no polynomial time (in k) algorithm which has a non-negligible advantage (again, in terms of k) in solving the BDHP for all sufficiently large k. Following are the two variations of BDHP [3].

Definition 5. Decisional Diffie-Hellman Problem *: Let $(q, \mathbb{G}, \mathbb{G}_T, e, P)$ be a 5-tuple generated by $\mathcal{G}(k)$, and let $a, b, c, r \in \mathbb{Z}_q^*$. The DBDHP in $\mathbb{G}$ is as follows: Given Given (P, aP, bP, cP, r) with some $a, b, c \in \mathbb{Z}_q^*$, Output is* **yes** *if $r = \ e(P, P)^{abc}$ and* **no** *otherwise. The (t, ϵ)-HDDH assumption holds in $\mathcal{G}$ if there is no algorithm $\mathcal{A}$ running in time at most t such that*

$$\mathbf{Adv}_{\mathbb{G}}^{DBDH}(\mathcal{A}) = \ |Pr[\mathcal{A}(P, aP, bP, cP, e(P, P)^{abc})) = \ 1] - Pr[\mathcal{A}(P, aP, bP, cP, r) = \ 1]| \geq \epsilon$$

where the probability is taken over all possible choices of (a, b, c, h).

Definition 6. Hash Decisional Diffie-Hellman Problem *:Let $(q, \mathbb{G}, \mathbb{G}_T, e, g)$ be a 5-tuple generated by $\mathcal{G}(k)$,$\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ be a secure cryptographic hash function, whether l is a security parameter, and let $x, y \in \mathbb{Z}_q^*, h \in \{0,1\}^l$, the HDDH problem in $\mathbb{G}$ is as follows: Given (P, aP, bP, cP, h), decide whether it is a hash Diffie-Hellman tuple $((P, aP, bP, cP\mathcal{H}(e(P, P)^{abc}))$. If it is right, outputs 1; and 0 otherwise. The (t, ϵ)-HDDH assumption holds in $\mathcal{G}$ if there is no algorithm $\mathcal{A}$ running in time at most t such that*

$$\mathbf{Adv}_{\mathbb{G}}^{HDDH}(\mathcal{A}) = \ |Pr[\mathcal{A}(P, aP, bP, cP\mathcal{H}(e(P, P)^{abc})) = \ 1] - Pr[\mathcal{A}(P, aP, bP, cP, h) = \ 1]| \geq \epsilon$$

where the probability is taken over all possible choices of (a, b, h).

4 Framework of Aggregate Signcryption

An ID-based Aggregate Signcryption scheme (IDASC) consists of the following six probabilistic polynomial time (PPT) algorithms:

- **Setup**: $(param, msk) \leftarrow \text{Set}(1^k)$ takes a security parameter $k \in \mathbb{N}$ and generates $param$, the global public parameters and msk, the master secret key.
- **Key Extract**: $(<S_{ID_i}, d_i>, P_{pub}, q_i) \leftarrow Ext(1^k, param, msk, ID_i)$ takes a security parameter k, the global parameters $param$, a master secret key msk and an identity of the sender ID_i to generate a private key $<S_{ID_i}, d_i>$ and public key P_{pub} and q_i corresponding to this identity.

- **Signcrypt**: $\sigma_i \leftarrow Signcrypt(1^k, param, m_i, X_i, d_i, ID_i, ID_B)$ takes a security parameter k, global parameters $param$ and $(m_i, X_i, d_i, S_{ID_i}, ID_i, ID_B)$ to generate signcrypt σ_i. Let $\mathcal{M}$, $\mathcal{W}$ and $\mathcal{R}$ are the message space, signcrypted message space and the space of sender respectively. Any member can be identified as U by its identity ID_U, where $U \in \mathcal{R}$.
  For any message $m_i \in \mathcal{M}, 1 \leq i \leq n, n \in \mathbb{Z}^+$.
- **Aggregate**:$\sigma \leftarrow Aggregate(\{\sigma_i, ID_i\}_{i=\ 1...n})$ The algorithm take the set of all signcryption $\{\sigma_i\}_{i=1...n}$ and the corresponding identity ID_i outputs the final aggregate signcryption σ.
- **UnSigncrypt**: $(\{m_i\}_{i=1...n}, Z_{agg}) \leftarrow UnSigncrypt(1^k, param, \sigma_{agg}, S_{ID_B}, d_B, ID_B)$ takes a security parameter k, the global parameters $param$, aggregate signcryption σ_{agg}, secret key of the receiver S_{ID_B} and d_B to generate the plaintext m_i and signature Z_{agg}.
- **Verify**: $(Valid/\perp) \leftarrow Verify(1^k, param, \{m_i\}_{i=1...n}, Z_{agg}, S_{ID_B}, d_B)$. The algorithm takes a security parameter k, a global parameters $param$, message m, signature Z_{agg} an the private key $<ID_B, d_B>$ outputs $Valid$ or $\perp$ for invalid signature.

5 Security notions

Security of signcryption consists of two different mechanism: one ensuring privacy, and the other authenticity. On a high level, privacy is defined somewhat analogously to the privacy of an ordinary encryption, while authenticity to that of an ordinary digital signature. For example, one can talk about indistinguishability of signcryptexts under chosen ciphertext attack, or existential unforgeability of signcryptexts under chosen message attack, among others. For compactness, we focus on the above two forms of security too, since they are the strongest. However, several new issues come up due to the fact that Signcryption and Unsigncryption take as an extra argument the identity of the sender and recipient.

Definition 7. (Confidentiality) *An identity based signcryption scheme is semantically secure or has indistinguishability against adaptive chosen ciphertext attack (IND-IDASC-CCA2) is no polynomial bounded(PPT) adversary has a non-negligible advantage in the following game.*

1. **Initial**: The challenger $\mathcal{C}$ runs the **Setup** algorithm with the security parameter k as input and sends the system parameters $param$ to the adversary $\mathcal{A}$ and keeps the master private key msk secret.
2. The adversary $\mathcal{A}$ performs polynomial bounded number of queries to the oracles provided to $\mathcal{A}$ by $\mathcal{C}$. The description of the queries in the first phase are listed below:
   - **Extraction oracle** : $<S_{ID_i}, d_i> \leftarrow Ext(mask, ID_i)$. $\mathcal{A}$ submits an identity ID_i to the extraction oracle and receives the private key pairs $<S_{ID_i}, d_i>$ corresponding to ID_i.
   - **Signcryption oracle** : $\mathcal{A}$ submits a message m_i, the signer identity ID_i and the receiver identity ID_r to the challenger $\mathcal{C}$. $\mathcal{C}$ computes private key $<S_{ID_i}, d_i>$ for $ID_i$and runs the algorithm $Signcrypt(m_i, X_i, d_i, ID_i, ID_B)$ to obtain signcryption σi . Finally $\mathcal{C}$ returns σ_i to $\mathcal{A}$.
   - **UnSigncryption oracle**: $\mathcal{A}$ submits the receiver identity $ID_B \notin \{ID_i\}_{i=1...n}$ to $\mathcal{C}$. $\mathcal{C}$ generates the private key pair $<S_{ID_B}, d_B>$ by querying the Key Extraction oracle. $\mathcal{C}$ unsigncrypts using the private key pairs $<S_{ID_B}, d_B>$ and returns the output to $\mathcal{A}$. If σ is an invalid signcrypted ciphertext returns a symbol $\perp$ for rejection from $\{ID_i\}_{i=1...n}$ to ID_B. $\mathcal{A}$ can present its queries adaptively *i.e* every request may depend on the response to the previous queries.
3. $\mathcal{A}$ chooses two messages m_{i0}, m_{i1}, identities $\{ID_i\}_{i=1...n}$ and ID_B of sender and receiver on which $\mathcal{A}$ wishes to be challenged. The challenger $\mathcal{C}$ now chooses a random bit $b \in \{0, 1\}$ and computes the aggregate signcryption σ_{agg} by running $\sigma_i^* = Signcrypt(1^k, param, m_i, X_i, d_i, ID_i, ID_B)$ and aggregate algorithm $Aggregate(\{\sigma_i, ID_i\}_{i=\ 1...n})$ and sends to $\mathcal{A}$.

4. $\mathcal{A}$ performs polynomially bounded number of new queries in the first stage, with the restrictions that $\mathcal{A}$ cannot query the UnSigncryption oracle for the unsigncryption of σ^*_{agg} or the $Keygen$ oracles for the private keys pairs of ID^*_B.
5. At the end of the game $\mathcal{A}$ returns a bit $b^{'}$ and wins the game if $b^{'} = b$. The success probability is defined by:

$$Adv^{(IDASC-IND-CCA2)}(\mathcal{A}) = |Pr[b^{'} = b] - \tfrac{1}{2}|$$

Here Adv is called the advantage for the adversary in the above game.

Definition 8. (Signature Unforgeability) *An identity based aggregate signcryption scheme (IDASC) is said to be existentially signature unforgeable against adaptive chosen-messages attacks (EUF-IDASC-CMA) if no polynomial bounded adversary (PPT) has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm with the security parameter k as input and sends the system parameters $param$ to the adversary $\mathcal{A}$ and keeps the master private key msk secret.
2. $\mathcal{A}$ performs polynomial bounded number of queries to the same oracles described in IDASC-IND-CCA2 game which are simulated by the challenger $\mathcal{C}$. The queries may be adaptive $i.e$ the current query may depend on the previous query responses.
   The adversary $\mathcal{A}$ returns a recipient identity ID_B and a ciphertext σ_i
   $\mathcal{A}$ submits a signcryption ciphertext σ_i and two identity ID^*_B and ID^*_i , $\mathcal{A}$ wins the game if the ciphertext σ_i is decrypted as a signed message (ID_i, m^*_i, V^*_i) having $ID_i \neq ID_B, ID_i \in \{ID_i\}_{i=1...n}$ result of the $UnSingncrypt(\sigma_{agg}, S_{ID_B}, d_B)$, otherwise returns the symbol $\perp$.
   Formally, we can be defined as
   - $(\{m^*_i\}_{i=1...n}, Z^*_{agg}) \leftarrow UnSigncrypt(1^k, param, \sigma^*_{agg}, S_{ID^*_B}, d^*_B, ID^*_B)$
     takes a security parameter k, the global parameters $param$, aggregate signcryption σ_{agg}, secret key of the receiver S_{ID_B} and d_B to generate the plaintext m_i and signature Z_{agg}. $i.e$ $\mathcal{A}$ submit a signcryption ciphertext σ^*_{agg}, global parameters $param$, k and identity ID^*_B returns $\{m^*_i\}_{i=1...n}, Z^*_{agg}$ such that $valid \leftarrow Verify(m^*_i, \sigma^*, \{ID^*_i\}_{i=1...n})$.
   - There will be no signcryption oracle decrypts to (m^*, σ^*) such that $valid \leftarrow Verify(m^*, \sigma^*, \{ID^*_i\}_{i=1...n})$.
   - No extra query was made on $\{ID^*_i\}_{i=1...n}$.

   $\mathcal{A}$'s advantage is defined as

$$Adv^{EUF-IDASC-CMS}_{\mathcal{A}} = Pr[Verify(m^*_i, \sigma^*, \{ID^*_i\}_{i=1...n}) = Valid]$$

Definition 9. (Ciphertext Unforgeability) *An identity based aggregate signcryption scheme (IDASC) is said to be existentially ciphertext unforgeable against adaptive chosen-messages attacks (AUTH-IDASC-CMA) if no polynomial bounded adversary (PPT) has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm with the security parameter k as input and sends the system parameters $param$ to the adversary $\mathcal{A}$ and keeps the master private key msk secret.
2. The adversary $\mathcal{A}$ performs polynomial bounded number of queries to the oracles provided to $\mathcal{A}$ by $\mathcal{C}$. The attack may be conducted adaptively $i.e$ the current query may depend on the previous query responses and allows the same queries described as in the (IND-IDASC-CCA2) game .
3. Forgery. The adversary $\mathcal{A}$ produces a new aggregate signcryption σ_{agg} from a set $\{ID_i\}_{i=1...n}$ of n users on messages $m_i, \forall i = 1 \ldots n$ to a final receiver $ID_B \notin \{ID_i\}_{i=1...n}$, where the private keys of the users in $\{ID_i\}_{i=1...n}$ was not queried in query phase and σ_i is not the output of a previous query to the Signcrypt queries. Outcome. The adversary $\mathcal{A}$ wins the game if $\perp$ is not returned by $UnSigncrypt(1^k, param, \sigma_{agg}, S_{ID_B}, d_B, ID_B)$.

6 Proposed ID-based Aggregate Signcryption Scheme

The scheme comprise five randomized polynomials algorithms.

- **Setup** Given security parameters k, the PKG chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q. A generator P of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and collision resistant hash function $\mathcal{H}_0 : \{0,1\}^* \to \mathbb{F}_q^*$, $\mathcal{H}_1 : \mathbb{G}_2 :\to \{0,1\}^l \times \mathbb{F}_q^*$, $\mathcal{H}_2 : \{0,1\}^l \times \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \times \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{F}_q^*$, $\mathcal{H}_3 : \{0,1\}^l \times \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{F}_q^* \times \mathbb{G}_1 \times \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{F}_q^*$. It chooses a master-key $s \in \mathbb{F}_q^*$ and computes $P_{pub} = sP$. The PKG publishes the system public parameters

$$\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, n, \hat{e}, P, P_{pub}, \mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3)$$

- **Extract** This algorithms follows of the following steps
  - Given an identity $ID_i \in \{0,1\}^*$, PKG computes $Q_{ID_i} = \mathcal{H}_0(ID_i)$ and the partial private key as $S_{ID_i} = s \cdot Q_{ID_i}$.
  - Chooses a random number $x_i \leftarrow_R \mathbb{F}_q^*$ and computes $X_i = x_i \cdot P$.
  - Computes $d_i = (x_i + sq_i) \bmod q$, for all $i = 1 \ldots n$. corresponding public key $q_i = \mathcal{H}_0(ID_i \| X_i)$.
  - The PKG send the corresponding private key $<S_{ID_i}, d_i>$ and public key $<X_i, q_i>$ through a secure channel to the users.
- **Signcrypt**$(m_i, X_i, d_i, ID_i, ID_B)$: The algorithm works as follows
  - Chooses a random $r_i \leftarrow_R \mathbb{F}_q^*$ and computes $W_i = r_i \cdot P, w_i = \hat{e}(P_{pub}, Q_{ID_B})^{r_i}$.
  - Computes $h_{1i} = \mathcal{H}_1(w_i), h_{2i} = \mathcal{H}_2(m_i, ID_i, X_i, w_i, ID_B, X_B)$.
  - Computes $h_{3i} = \mathcal{H}_3(m_i, ID_i, X_i, w_i, ID_B, X_B, h_{2i})$.
  - Computes $v_i = (r_i h_{2i} + h_{3i} d_i) \bmod q$.
  - Computes $C_i = (m_i \| v_i) \oplus h_{2i}, Z_i = v_i \cdot P$.
  - Output $\sigma_i = <C_i, W_i, Z_i, X_i>$ is the signcryption of ID_i on message m_i.
- **Aggregate**$(\{\sigma_i, ID_i\}_{i=1\ldots n})$: On input a set of signcryption $\sigma_i = <C_i, W_i, Z_i, X_i>$, $i = 1 \ldots n$ and the corresponding identity ID_i such that $\forall i = 1 \ldots n, \sigma_i$ are the signcryption of message m_i by ID_i.
  1. $Z_{agg} = \sum_{i=1}^{n} Z_i$, $Z_i = v_i \cdot P$, $i = 1 \ldots n$
  2. Output the final aggregate signcryption $\sigma_{agg} = <\{C_i, W_i, X_i, ID_i\}_{i=1\ldots n}, Z_{agg}>$.

  The aggregate can be computed by the sender or a trusted third party.
- **UnSigncrypt**$(\sigma_{agg}, S_{ID_B}, d_B)$: To decrypt and verify the aggregate signcryption $\sigma_{agg} = <\{C_i, W_i, X_i, ID_i\}_{i=1\ldots n}, Z_{agg}>$, the receiver with identity ID_B use his private key $<S_{ID_B}, d_B>$ and follows the following steps.
  - Computes $C_i \oplus h_{1i} = m_i \| v_i$, where $h_{1i} = \mathcal{H}_1(w_i), w_i = \hat{e}(W_i, S_{ID_B})$.
  - $\forall i = 1 \ldots n$, computes $h_{2i} = \mathcal{H}_2(m_i, ID_i, X_i, w_i, ID_B, X_B)$.
  - Checks whether the following equation holds

$$Z_{agg} = \sum_{i=1}^{n} v_i \cdot P \tag{1}$$

$$Z_{agg} = \sum_{i=1}^{n} (h_{2i} W_i) + \sum_{i=1}^{n} (h_{3i} X_i) + P_{pub} \sum_{i=1}^{n} h_{3i} q_i \tag{2}$$

7 Proof of Correctness

$w_i = \hat{e}(W_i, S_{ID_B}) = \hat{e}(r_i P, S_{ID_B})$
$= \hat{e}(P, S_{ID_B})^{r_i}$
$= \hat{e}(P, sQ_{ID_B})^{r_i} = \hat{e}(sP, Q_{ID_B})^{r_i}$
$= \hat{e}(P_{pub}, Q_{ID_B})^{r_i}.$
$Z_{agg} = \sum_{i=1}^{n}(v_i \cdot P) = \sum_{i=1}^{n}(r_i h_{2i} + h_{3i} d_i) \cdot P$
$= \sum_{i=1}^{n} h_{2i}(r_i \cdot P) + \sum_{i=1}^{n} h_{3i}(d_i \cdot P)$
$= \sum_{i=1}^{n} h_{2i}(r_i \cdot P) + \sum_{i=1}^{n} h_{3i}(x_i + sq_i) \cdot P$
$= \sum_{i=1}^{n} h_{2i} W_i + \sum_{i=1}^{n} h_{3i} X_i + P_{pub} \sum_{i=1}^{n} h_{3i} q_i$
$Z_{agg} = \sum_{i=1}^{n} v_i \cdot P$
$= \sum_{i=1}^{n}(h_{2i} W_i) + \sum_{i=1}^{n}(h_{3i} X_i) + P_{pub} \sum_{i=1}^{n}(h_{3i} q_i).$

7.1 Security Analysis

In this section we have proved that the proposed scheme is secure in random oracle model with respect to security properties IDASC-IDASC-CCC2, AUTH-IDASC-CMA and EUF-IDASC-CMA defined in definition-7, 8 and 9. We prove the security as similar to [2].

Theorem 1 *In the random oracle model, we assume the adversary $\mathcal{A}$ for IND-IDASC-CCA2 able to distinguish two valid ciphertext during the game with a non-negligible advantage and run Keygen queries, Signcrypt queries, and Unsigncrypt queries; then there exists a distinguisher $\mathcal{B}$ that can solve an instances of Decisional Bilinear Diffie-Hellman problem with a non-negligible advantage.*

Proof:

– **Setup**: The distinguisher $\mathcal{B}$ receives a random instance (P, aP, bP, cP, μ) of the Decisional Bilinear Diffie-Hellman problem. His goal is to decide whether $\mu = \hat{e}(P, P)^{abc}$ or not. $\mathcal{B}$ will run $\mathcal{A}$ as a subroutine and act as $\mathcal{A}$s challenger in the IND-IDASC-CCA2 game. $\mathcal{B}$ needs to maintain lists L_0, L_1, L_2 and L_3 that are initial empty and are used to keep track of answers to queries asked by $\mathcal{A}$ to oracles $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ respectively.

Oracle Simulation:

1. $\mathcal{H}_0$**-Oracle**: For $\mathcal{H}_0$-queries, at the beginning of the game, $\mathcal{B}$ gives $\mathcal{A}$ the system parameters with $P_{pub} = cP$ (c is unknown to $\mathcal{B}$ and plays the role of the PKGs master-key). Then $\mathcal{B}$ chooses two distinct random numbers $i, j \in \{1 \ldots q_{\mathcal{H}_0}\}$. $\mathcal{A}$ asks a polynomial bounded number of $\mathcal{H}_0$ requests on identities of his choice. At the i^{th} $\mathcal{H}_0$ request, $\mathcal{B}$ answers by $\mathcal{H}_0(ID_i) = aP$. At the j^{th}, he answers by $\mathcal{H}_0(ID_j) = bP$. Since aP and bP belong to a random instance of the DBDH problem, $\mathcal{A}$s view will not be modified by these changes. Hence, the private keys S_{ID_i} and S_{ID_j} (which are not computable by $\mathcal{B}$) are respectively acP and bcP. Thus the solution $\hat{e}(P, P)^{abc}$ of the BDH problem is given by $\hat{e}(Q_{ID_i}, S_{ID_j}) = \hat{e}(S_{ID_i}, Q_{ID_j})$. For requests $\mathcal{H}_0(ID_k)$ with $k \neq i, j$, $\mathcal{B}$ chooses $b_k \leftarrow_R \mathbb{F}_q^*$, puts the pair (ID_k, b_k) in list L_0 and answers $\mathcal{H}_0(ID_k) = b_k P$.

   Further on input $ID_i \in \{0,1\}^*$, $\mathcal{B}$ first checks the L_0-list $<ID_i, X_i, q_i, x_i>$, if $ID_i = ID_B>$, selects new random $\gamma_i \leftarrow_R \mathbb{F}_q^*$, sets $X_i = b \cdot P, q_i = \gamma_i$, add this tuple $<ID_i, X_i, q_i, *>$ to the L_0-list and returns q_i. Otherwise, $\mathcal{B}$ selects a new random $\gamma_i \leftarrow_R \mathbb{F}_q^*$, $x_i \leftarrow_R \mathbb{F}_q^*$, sets $X_i = x_i \cdot P, q_i = \gamma_i$, add this tuple $<ID_i, X_i, q_i, x_i>$ to the L_0-list and returns q_i.

2. $\mathcal{H}_1$**-Oracle**: When a $(m_i, ID_i, X_i, w_i, ID_B, X_B)$ is submitted in $\mathcal{H}_1$ query for the first time, $\mathcal{B}$ returns checks the L_1-list, whether the tuples $<w_i, h_{1i}>$ in L_1-list, $\mathcal{B}$ returns h_{1i}, otherwise, $\mathcal{B}$ chooses a new random $h_{1i} \leftarrow_R \mathbb{F}_q^*$, includes the tuples $<w_i, h_{1i}>$ to the L_1-list and return h_{1i}.
3. $\mathcal{H}_2$**-Oracle**: On input $(m_i, ID_i, X_i, w_i, ID_B, X_B)$, $\mathcal{B}$ first checks the L_2-List, whether the tuple $<m_i, ID_i, X_i, W_i, ID_B, X_B, h_{2i}>$ in the L_2-List, $\mathcal{B}$ returns h_{2i}, otherwise $\mathcal{B}$ chooses a new random $h_{2i} \leftarrow_R \mathbb{F}_q^*$, includes h_{2i} to the L_2-list and return h_{2i}.
4. $\mathcal{H}_3$**-Oracle**: On input $(m_i, ID_i, X_i, w_i, ID_B, X_B, h_{2i})$, $\mathcal{B}$ first checks the L_3-List, whether the tuple $<m_i, ID_i, X_i, W_i, ID_B, X_B, h_{2i}>$ in the L_3-List, $\mathcal{B}$ returns h_{3i}, otherwise $\mathcal{B}$ chooses a new random $h_{3i} \leftarrow_R \mathbb{F}_q^*$, includes h_{3i} to the L_3-list and return h_{3i}.
5. **Keygen-Oracle**: When $\mathcal{A}$ makes a $Keygen$ query with ID_i as the input, $\mathcal{B}$ checks the L_0-List to verify whether or not there is an entry for ID_i. If the L_0-List does not contain an entry for ID_i, return $\perp$. Otherwise, if $ID_i = ID_B$, $\mathcal{B}$ recovers the tuple $<ID_i, X_i, q_i, x_i>$ from the L_0-List and returns $<X_i, q_i, *, *>$, if $ID_i \neq \{ID_i\}_{i=\ 1\ldots n}$ $\mathcal{B}$ recovers the tuple $<ID_i, X_i, q_i, x_i>$ from the L_0-List and returns $<X_i, q_i, S_{ID_i}, d_i>$, where $S_{ID_i} = x_i(aP) = a(x_iP) = aX_i$ and $d_i \leftarrow_R \mathbb{F}_q^*$ is randomly selected.
6. **Signcryption Oracle**: When $\mathcal{A}$ makes a Signcrypt query with ID_i as the input, $\mathcal{B}$ checks the L_0-List to verify whether or not there is an entry for ID_i. If the L_0-List does not contain an entry for ID_i returns$\perp$. Otherwise, $\mathcal{B}$ executes $Signcrypt(m_i, X_i, d_i, ID_i, ID_B)$ as usual and returns what the signcrypt algorithm returns.
7. **UnSigncryption Oracle**: When $\mathcal{A}$ makes an $Unsigncrypt$ query with $\sigma_{agg} = <\{C_i, W_i, X_i, ID_i\}_{i=\ 1\ldots n}, Z_{agg}>$ and the receiver with identity ID_B, $\mathcal{B}$ first verifies whether or not there are entries for $ID_i, (ID_i \neq ID_B)$ and ID_B in L_0-List and there is an entry of the form $<ID_i, X_i, q_i, \gamma_i>$. If at least one of these conditions is not satisfied, $\mathcal{B}$ returns$\perp$. Otherwise, $\mathcal{B}$ executes $Unsigncrypt(\sigma_{agg}, S_{ID_B}, d_B)$ in the normal way and returns what the $Unsigncrypt$ algorithm returns.

After getting sufficient training, $\mathcal{A}$ submits two equal length of messages m_{i0} and m_{i1}. $\mathcal{A}$ randomly chooses a bit $b^* \leftarrow \{0, 1\}$ and obtains the challenge signcrypted ciphertext by running $Signcrypt(m_{ib^*}, X_i, d_i, ID_i, ID_B)$ and $Aggregate(\{\sigma_i^*, ID_i\}_{i=\ 1\ldots n})$, then returns σ_{agg}^* to $\mathcal{A}$.

Phase 2. This phase is similar to Phase 1. However, in Phase 2, $\mathcal{A}$ cannot ask for $Unsigncrypt$ on the challenge aggregate Signcrypt $\sigma_{agg}^* = <\{C_i, W_i, X_i, ID_i\}_{i=1\ldots n}, Z_{agg}>$ or the $Keygen$ queries for the secret keys ID_B.

Output: After $\mathcal{A}$ has made a sufficient number of queries, $\mathcal{A}$ returns its guess: a bit. If then $\mathcal{B}$ outputs 1 as the answer to the DBDH problem. Otherwise, it outputs 0. Since the adversary is denied access to the Unsigncrypt oracle with the challenge signcryption, for $\mathcal{A}$ to find that m_i is not a valid ciphertext, $\mathcal{A}$ should have queried the $\mathcal{H}_1$ Oracle with $w_i = e(W_i, S_{ID_B})$. Here S_{ID_B} is the private key of the receiver, and it is $aX_B = (bP)a = abP$. Also, $\mathcal{B}$ has set $W_i = cP$. We have $w_i = e(W_i, S_{ID_B}) = e(cP, abP) = e(P, P)^{abc}$.

Theorem 2 *In the random oracle model, the proposed ASC is secure against any probabilistic polynomial time adversary $\mathcal{A}$ for AUTH-IDASC-CMA if the Elliptic Curve Discrete Logarithm Problem is hard in $\mathbb{G}_1$.*

Proof: $\mathcal{B}$ receives a random instance $(P, W_{r_\alpha}) = r_\alpha P$ and $(P, d_\alpha P)$ of ECDLP as a challenge in the AUTH-IDASC-CMA game defined in Definition 2. His goal is to determine r_α and d_α. $\mathcal{B}$ will run $\mathcal{A}$ as a subroutine and act as $\mathcal{A}$s challenger in the AUTH-IDASC-CMA game. $\mathcal{A}$ can compute $d_\alpha P$ as $W_\alpha + (sP)q_\alpha$, $d_\alpha P = (x_\alpha + sq_\alpha) \cdot P = W_\alpha + (sP)q_\alpha$.

- $\mathcal{H}_0$ **Oracle**: For $\mathcal{H}_0$-queries on input $ID_i \in \{0,1\}^*$, $\mathcal{B}$ first checks the L_0-list $<ID_i, X_i, q_i, x_i>$, selects random $\gamma_i \leftarrow_R \mathbb{F}_q^*$, sets $X_i = x_i \cdot P, q_i = \gamma_i$, add this tuple $<ID_i, X_i, q_i, *>$ to the L_0-list and returns q_i.
- **Keygen Oracle**: When $\mathcal{A}$ makes a *Keygen* query with ID_i as the input, $\mathcal{B}$ checks the L_0-List to verify whether or not there is an entry for ID_i. If the L_0-List does not contain an entry for ID_i, return $\perp$. Otherwise, if $ID_i \in \{ID_i\}_{i=\ 1\ldots n}$, $\mathcal{B}$ recovers the tuple $<ID_i, X_i, q_i, x_i>$ from the L_0-List and returns $<X_i, q_i, *, *>$, if $ID_i \neq \{ID_i\}_{i=\ 1\ldots n}$ $\mathcal{B}$ recovers the tuple $<ID_i, X_i, q_i, x_i>$ from the L_0-List and returns $<X_i, q_i, S_{ID_i}, d_i>$, where $S_{ID_i} = x_i(aP) = a(x_iP) = aX_i$ and $d_i \leftarrow_R \mathbb{F}_q^*$ is randomly selected.
- **Forgery**: $\mathcal{A}$ chooses the corresponding senders identities set $\{ID_i\}_{i=\ 1\ldots n}$ and receiver identity ID_B and returns a forged signcryption $\sigma_\alpha^* = <C_\alpha^*, W_\alpha^*, Z_\alpha^*, X_\alpha^*>$ on message m_α^* from $ID_\alpha \in \{ID_i\}_{i=\ 1\ldots n}$ to $\mathcal{B}$. $\mathcal{B}$ retrieves the entry corresponding to ID_B in the L_0-List and uses s_B to execute $Unsigncrypt(\sigma_{agg}, S_{ID_B}, d_B)$. If σ_α^* is a valid signcryption from ID_α to receiver ID_B, that is, a message m_α^* is returned by the *Unsigncrypt* algorithm, then $\mathcal{B}$ applies the oracle replay technique to produce two valid signcryptions $\sigma_\alpha^{'} = <C_\alpha^{'}, W_\alpha^{'}, Z_\alpha^{'}, X_\alpha^{'}>$ and $\sigma_\alpha^{''} = <C_\alpha^{''}, W_\alpha^{''}, Z_\alpha^{''}, X_\alpha^{''}>$ on message m_α from the ID_α to receiver ID_B. $\mathcal{B}$ obtains the signatures as $v_\alpha^{'} = r_\alpha h_{2\alpha}^{'} + h_{3\alpha}^{'} d_\alpha$ and $v_\alpha^{''} = r_\alpha h_{2\alpha}^{''} + h_{3\alpha}^{''} d_\alpha$ with $h_{2\alpha}^{'} \neq h_{2\alpha}^{''}$ and $h_{3\alpha}^{'} \neq h_{3\alpha}^{''}$. The PPT algorithm $\mathcal{B}$ can computes r_α and d_α as

$$r_\alpha = \frac{v_\alpha^{'} h_{3\alpha}^{''} - v_\alpha^{''} h_{3\alpha}^{'}}{h_{2\alpha}^{'} h_{3\alpha}^{''} - h_{2\alpha}^{''} h_{3\alpha}^{'}},\ h_{2\alpha}^{'} h_{3\alpha}^{''} - h_{2\alpha}^{''} h_{3\alpha}^{'} \neq 0.$$

$$d_\alpha = \frac{v_\alpha^{'} h_{2\alpha}^{''} - v_\alpha^{''} h_{2\alpha}^{'}}{h_{3\alpha}^{'} h_{2\alpha}^{''} - h_{3\alpha}^{''} h_{2\alpha}^{'}},\ h_{3\alpha}^{'} h_{2\alpha}^{''} - h_{3\alpha}^{''} h_{2\alpha}^{'} \neq 0.$$

Theorem 3 *In the random oracle model, the proposed ASC is secure against any probabilistic polynomial time adversary $\mathcal{A}$ for EUF-IDASC-CMA if the Decisional Biliner Diffie-Hellman Problem is hard in $\mathbb{G}_1$*

Proof: $\mathcal{B}$ simulates the $\mathcal{A}$'s challenger in the EUF-IDASC-CMA game. $\mathcal{B}$ can performs queries as defined in Definition-9. we describe the process as follows.

Keygen Oracle: When $\mathcal{A}$ makes a *Keygen* query with ID_i as the input, $\mathcal{B}$ checks the L_0-List to verify whether or not there is an entry for ID_i. If the L_0-List does not contain an entry for ID_i, return $\perp$. Otherwise, if $ID_i = ID_\alpha$, $\mathcal{B}$ recovers the tuple $<ID_i, X_i, q_i, x_i>$ from the L_0-List and returns $<X_i, q_i, *, *>$, if $ID_i \neq ID_\alpha$ $\mathcal{B}$ recovers the tuple $<ID_i, X_i, q_i, x_i>$ from the L_0-List and returns $<X_i, q_i, S_{ID_i}, d_i>$, where $S_{ID_i} = x_i(sP)$ and $d_i \leftarrow_R \mathbb{F}_q^*$ is randomly selected.

Eventually, $\mathcal{A}$ returns a forgery, consisting of a ciphertext and a recipient identity ID_B. $\mathcal{B}$ decrypts the ciphertext for ID_B (by invoking its own decryption oracle), which causes the plaintext forgery (ID_i, m_i, V_i) to be revealed. Note that if $\mathcal{B}$ has made the correct guess, that is, $ID_i = ID_\alpha$, then $ID_B \neq ID_\alpha$ and the decryption works.

If σ_i is a valid signcryption from ID_i to receiver ID_B, that is, a message m_i is returned by the *Unsigncrypt* algorithm, then $\mathcal{B}$ applies the oracle replay technique to produce two valid signed messages (ID_i, m_i, V_i) and (ID_i, m_i, V_i) on a message m_i from the ID_i to receiver ID_B. This is achieved by running the truing machine again with the same random tape but with a different hash value. $\mathcal{B}$ obtains the signatures$v_\alpha^{'} = r_\alpha h_{2\alpha}^{'} + h_{3\alpha}^{'} d_\alpha$ and $v_\alpha^{''} = r_\alpha h_{2\alpha}^{''} + h_{3\alpha}^{''} d_\alpha$ with $h_{2\alpha}^{'} \neq h_{2\alpha}^{''}$ and $h_{3\alpha}^{'} \neq h_{3\alpha}^{''}$.

8 Performance

Efficiency of aggregate sincryption scheme can be evaluated with respect to computational cost and ciphertext length [2]. To compute the computational cost, we consider costly operations that

include point multiplications in $\mathbb{G}_1$ ($Mul(\mathbb{G}_1)$), exponentiations in $\mathbb{G}_2$ ($Exp(\mathbb{G}_2)$), and pairing operations ($Pairing$).

Let we symbolize confidentiality (Con), unforgeability (Unf), public verifiability (PuV), forward security (FoS), ciphertext unlinkability (CiU) and ciphertext anonymity (CiA). "$\surd$" and "$\times$" denotes Yes and No respectively.

Table 1. Security Comparison

Scheme	$Conf$	Unf	PuV	Fos	CiU	CiA
Libert and Quisquater-1	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
Libert and Quisquater-2	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
Libert and Quisquater-3	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\times$
Malone-Lee	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
Barreto *et al.*	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
Boyen	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Chow *et al.*	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
IDASC	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

Table 2. Comparison of Computational Cost

	Signcrypt			*UnSigncrypt*		
	$Pairing$	$Mul(\mathbb{G}_1)$	$Exp(\mathbb{G}_2)$	$Pairing$	$Mul(\mathbb{G}_1)$	$Exp(\mathbb{G}_2)$
Libert and Quisquater-1	1(+1)	2	2	4		2
Libert and Quisquater-2	1(+1)	2	2	4		2
Libert and Quisquater-3	1	2	1	2	1	
Malone-Lee	1	3		4		1
Barreto *et al.*		2	1	2	1	1
Boyen	1	3	1	4	2	
Chow *et al.*	2	2		4	1	
IDASC	1	2		1	1	

9 Conclusion

In this article, we proposed a provably secure aggregate signcryption scheme in random oracle model which is more efficient than the scheme proposed by Xun-Yi Ren *e*t al. [2] with respect to the length of Ciphertext and secure than the other schemes summarized in the tables. We prove that the scheme meets the three strong security requirements confidentiality, signature unforgeability and ciphertext unforgeability in the random oracle model under the assumption, Elliptic Curve Discrete Logarithm problem and Bilinear Diffie-Hellman Problem are computationally hard. It can be implemented on low power devices such as PDA, smart card, cell phone, wireless

Table 3. Comparison of Ciphertext size

Scheme	Ciphertext size
S.S.D.Selvi *et al.* and D.Boneh *et al.*	$\mid M \mid + \mid \mathbb{Z}_q^* \mid +3 \mid \mathbb{G}_1 \mid$
Xun-Yi Ren *et al.*	$\mid M \mid + \mid \mathbb{Z}_q^* \mid +4 \mid \mathbb{G}_1 \mid$
IDASC	$\mid M \mid + \mid \mathbb{Z}_q^* \mid +2 \mid \mathbb{G}_1 \mid$

sensor network. Since our scheme is compact, fast and unforgeable, in real time application such as key transport, multi cast electronics commerce, authenticated e-mail, it can be applied.

References

1. J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption,*Journal of Cryptology*, vol. 20, no 2, pp. 203-235,2007.
2. Xun-Yi Ren, Zheng-Hua Qi and Geng. Provably secure Aggregate Signcryption Scheme, *ETRI journal* Vol.34(3), pp-421-428, 2012
3. J.Kar. Provably Secure Identity Based Online/Offline Signature Scheme for Wireless Sensor network *Cryptology ePrint Archive, Report 2012/162*, 2012, http:/eprint.iacr.org.
4. Y. Zheng. Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost(signature) + cost (encryption). *In Advances in Cryptology CRYPTO1997, Lecture Notes in Computer Science*, vol. 1294. Springer: Heidelberg, 1997, 165-179.
5. G. Yu, X.X. Ma, Y. Shen Provable secure identity based generalized signcryption scheme, *Theoretical Computer Science*, vol. 411(40), pp. 3614-3624, 2010.
6. B. Zhang, Q. L. Xu An ID-Based Anonymous Signcryption Scheme for Multiple Receivers Secure in the Standard Model, *in Proceedings of Computer Science and Information Technology*, Chengdu, China, pp. 15-27, 2010.
7. Z. P. Jin, Q. Y. Wen, and H. Z. Du An improved semantically-secure identity-based signcryption scheme, *in the standard model Computers & Electrical Engineering*, Vol. 36, pp. 545-552, 2010.
8. Y. Han and X. Yang Elliptic curve based generalized signcryption scheme,*Cryptology ePrint Archive, Report 2006/126*, 2006, http:/eprint.iacr.org.
9. Adi Shamir. Identity-based cryptosystem and signature schemes. *In CRYPTO 84*, pages 4753, 1984.
10. Dan Boneh and Matthew K. Franklin Identity-based encryption from the weil pairing. *In CRYPTO 01, Lecture Notes in Computer Science*, vol.2139, pp 213229. Springer, 2001.
11. John Malone-lee. Identity-based signcryption. *In Proceedings of Public Key Cryptography - PKC 2005*, LNCS 3386, pp 362379. Springer, 2002.
12. A new traitor tracing, *ICICE Trans.* Vol.E85-A, No-2, pp481-484, 2002.
13. P.S.L.M. Barreto, B. Libert, N. McCullagh, and J. J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. *In Proceedings of Advances in Cryptology* ASIACRYPT 2005, LNCS 3788, Springer-Verlag, pp. 515-32, 2005.
14. S. Even, O. Goldreich, and S. Micali. *On-Line/Off-Line digital signatures. In Proceedings of Advances in Cryptology* CRYPTO 89, LNCS, vol. 435. Springer Berlin, 1990, pp. 263275.
15. Ren Yanli and Gu Dawu. Efficient identity based signature/signcryption scheme in the standard model.*In The First International Symposium on Data, Privacy, and E-Commerce, 2007*, pp 133-137, 2007.
16. Yong Yu, Bo Yang, Ying Sun, and Shenglin Zhu. Identity based signcryption scheme without random oracles. *Computer Standards & Interfaces*, 31(1):56-62, 2009.
17. Xing Wang and Hai feng Qian. Attacks against two identity-based signcryption schemes. *In Second International Conference on Networks Security Wireless Communications and Trusted Computing*, vol 1, pp 24-27, 2010.
18. S.Sharmila Deva Selvi, S.Sree Vivek, J.Shriram, S.Kalaivani, and C.Pandu Rangan. Security analysis of aggregate signature and batch verification signature schemes. *Cryptology ePrint Archive*, Report 2009/290, 2009, http://eprint.iacr.org.

19. B. Libert, and J. J. Quisquater. A new identity based signcryption schemes from pairings. *In Proceedings of IEEE Information Theory Workshop*, Paris, France, pp. 155-158, 2003.
20. S. S. M. Chow, S. M. Yiu, L. C. K. Hui, and K. P. Chow. Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity. *In Proceedings of Information Security and Cryptology-ICISC 2003*, LNCS 2971, Springer-Verlag, pp. 352-69, 2004.
21. X. Boyen. Multipurpose identity-based signcryption. *In Proceedings of Advances in Cryptology-CRYPTO 2003*, LNCS 2729, Springer-Verlag, pp. 383- 99, 2003.
22. L. Chen, and J. Malone-Lee. Improved identity-based signcryption. *In Proceedings of Public Key Cryptography-PKC 2005*, LNCS 3386, Springer-Verlag, pp. 362-79, 2005.
23. Alexandra Boldyreva, Craig Gentry, Adam ONeill, and Dae Hyun Yum. Ordered multi-signatures and identity based sequential aggregate signatures, with applications to secure routing,*ACM Conference on Computer and Communications Security*, pp 276285. ACM, 2007.