Efficient identity based aggregate signcryption scheme using bilinear pairings over elliptic curves

G. Swapna¹ and P. Vasudeva Reddy²

¹Department of Mathematics, Gokaraju Rangaraju Institute of Engineering and Technology, Telangana, INDIA.
²Department of Engineering Mathematics, Andhra University, Visakhapatnam, A.P. INDIA.

ABSTRACT. Signcryption is a cryptographic primitive that performs signature and encryption simultaneously at a cost significantly lower than the traditional signature-then-encryption methodology. An aggregate signature scheme is a digital signature scheme that combines $n$ signatures on $n$ different messages from $n$ different users to a single signature. In this paper, we propose an aggregate signcryption scheme in the identity based setting using bilinear pairings over elliptic curves by combining aggregate signatures with signcryption schemes. We prove that our scheme is secure and satisfies the unforgeability and confidentiality in the random oracle model with the assumptions that the CBDHP and BDHP are hard. Also our scheme is public verifiable and requires constant number of pairings for aggregate verification. Comparing with previous schemes of this kind, our scheme is more efficient in computation and communication.

1. Introduction

Public key cryptography (PKC) achieves confidentiality and authentication through encryption and digital signatures respectively. Now a days, many real-world applications require these different goals to be achieved simultaneously. Inspired by this, Zheng [1] provided a novel cryptographic primitive called “signcryption”. The motivation behind this kind of cryptosystem is to encrypt and sign information in a single operation which requires smaller bandwidth and computational costs than that of by doing both operations in consecutive manner.

In 1984, Shamir [2] first proposed the idea of Public Key Cryptography (ID-PKC) in Identity based setting to streamline key management issue in traditional certificate based Public Key Infrastructure (PKI). In ID-PKC, a user’s public key is directly access from specific features of users identity, such as an IP address having a place with a system or an email address related with a user. Private keys of user’s are generated by a trusted third party called a Private Key Generator (PKG). The direct evaluation of public keys in ID-PKC eliminates the necessity of certificates and the problems associated with them. A rapid development of ID-PKC has taken place, due to the contribution of Boneh et al. [3] in 2001.

After Boneh et al. [3], there are several ID based signature schemes [4-5] and signcryption schemes [6-11] have been proposed using bilinear pairings. Not only signature and signcryption schemes, there
are many extensions such as proxy, blind, multi, aggregate signcryption schemes using the bilinear pairings have been proposed in the literature.

Concept of signatures on variety of different messages produced by different users have lot of practical significance in real world. In this process we come across the applications in which one entity has to verify many signatures simultaneously. For instance in a PKI of \( n \) users where each user is given a chain of \( n \) certificates, the chain contains \( n \) signatures by each of \( n \) Certificate Authorities (CAs) on \( n \) different certificates. S. Kent et al. [13] have proven that each router obtains a list of \( n \) signatures confirming to a certain path of length \( n \) in a network in the Secure Border Gateway Protocol (SBGP). A router signs its own fragment of the route and forwards the subsequent list of \( n+1 \) signature to the next router which makes the number of signatures are linear in routing messages in the length of the path. The method for compressing the list of signatures on different messages generated by different users would benefit both of these applications. This type of compression can be achieved precisely by the aggregate signature scheme.

Aggregation is useful to reduce bandwidth and storage and is especially attractive for mobile devices like sensors, cell phones and PDAs where communications is more power-expensive than computation and contributes significantly to reduce the battery life. In 2003, Boneh et al. [14] proposed the first aggregate signature, using BLS [15] short signature scheme, in which distinct signatures on distinct messages computed by distinct users can be aggregated into a single signature. Then \( n \) signatures are compressed into one, it reduces to a half of signature. Preferably, the length of the aggregate signature (excluding the messages and the public keys of the signers) is not linear to the number of users and should be a constant. According to signature aggregation techniques, there are two types of aggregations in which the aggregation of the signatures can be done by any one of the authenticated users in any order in the first method where as a signer adds his signature to the previous aggregate signature sent by the previous signer in the second method.

The first aggregate signature scheme in ID based setting is introduced by Cheon et al. [16] in 2004. Since then many ID-based aggregate signatures were presented in the literature [17-23]. In 2009, Selvi et al. [24] proposed three identity based aggregate signcryption schemes along with a formal model and a formal security proof. But their scheme is not public verifiable.

In 2012, Xun-Yi Ren et.al [25] proposed an aggregate signcryption scheme, which is based on the aggregate signature scheme [21]. In 2013, J. Kar [26] proposed an aggregate signcryption scheme with provably secure in random oracle model and they claimed that their scheme is public verifiable and also efficient than Xun-Yi Ren et al. [25] scheme. But this scheme is not public verifiable, because any third party (who is unaware of private key of the receiver) cannot verify the authenticity of the aggregate signcryption text without knowing the original message. However, the design of an aggregate signcryption scheme with public verifiability is far from satisfactory.

In this paper, we propose an aggregate signcryption (ID-ASC) scheme with public verifiability. The proposed aggregate signcryption scheme is constructed by taking \( n \) multiple signcryptions on \( n \) distinct messages signed by \( n \) distinct users. Our ID-ASC scheme involves constant number of pairings in aggregate verification, which significantly improves the efficiency of the system. Additionally, our scheme does not include any sort of interaction among the users before the aggregation, which reduces the communication complexity to a large extent. We formally prove that the proposed aggregate signcryption scheme is unforgeable and is CCA-2 secure in the random oracle model.

The rest of the paper is organized as follows: Section 2 gives some mathematical background which will be used in this paper; then the syntax and security requirements of ID-based aggregate signcryption scheme are discussed in Section 3; we present our ID-based aggregate signcryption
scheme in Section 4; proof of correctness, security and efficiency analysis of the proposed scheme are presented in Section 5; Section 6 concludes the paper.

2. Preliminaries

In this section, we briefly review bilinear pairings and some computational problems which are used in this paper. Some notations are used throughout this paper for our convenience and are represented in Table 1.

2.1. Bilinear Pairings

Bilinear pairing is a significant cryptographic primitive and has been extensively used in many applications in cryptography.

A bilinear pairing is a map \( \hat{e}: G_1 \times G_1 \rightarrow G_2 \), where \( G_1 \) additive cyclic group generated by \( P \) and \( G_2 \) is a multiplicative cyclic group of the prime order \( q \) satisfies the following properties.

1. Bilinear: \( \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} \) for all \( P, Q \in G_1 \) and \( a, b \in \mathbb{Z}_q^* \).
2. Non–Degenerate: There exists \( P, Q \in G_1 \) such that \( \hat{e}(P, Q) \neq 1 \).
3. Computability: There exists an efficient algorithm to compute \( \hat{e}(P, Q) \) for all \( P, Q \in G_1 \).

2.2. Computational Problems

For proving the security of our aggregate signcryption scheme in id based setting, we are using the following computational problems.

**Definition 1.** The Computational Diffie Hellman Problem (CDHP) in \( G_1 \) is such that given \( (P, aP, bP) \) with uniformly random choices of \( a, b \in \mathbb{Z}_q^* \), to compute \( abP \).

**Definition 2.** The Computational Bilinear Diffie-Hellman Problem (CBDHP) is such that given \( (P, aP, bP, cP) \) with uniformly random choices of \( a, b, c \in \mathbb{Z}_q^* \), to compute \( \hat{e}(P, P)^{abc} \).

**Definition 3.** The Elliptic Curve Discrete Logarithm Problem (ECDLP) is to find an integer \( n \), such that \( Q = nP \) for given \( P \) and \( Q \) in \( G_1 \).

| Notation | Meaning |
|----------|---------|
| \( k, k_s \) | Security parameter, master secret key of the system generated by PKG. |
| \( params \) | System Parameters. |
| \( Z_q^* \) | Group of elements \( 1, 2, \ldots, q-1 \) under addition modulo \( q \). |
| \( G_1, G_2 \) | Additive, Multiplicative cyclic groups of prime order \( q \). |
| \( H_1, H_2, H_3 \) | Cryptographic one way hash functions. |
| \( ID_A, ID_B \) | User Identity, Receivers Identity |
| \( S_{ID_A}, S_{ID_B} \) | User secret key, Receivers secret key respectively. |
| \( A, C \) | Adversary and Challenger respectively. |
| \( \xi \) | Challenger |
| \( \hat{e}: G_1 \times G_1 \rightarrow G_2 \) | An admissible bilinear map. |
| \( \sigma_{agg} \) | Signcryption and Aggregate signcryption on a message. |
3. Syntax and security model for our ID-ASC scheme

In this segment, we present a model for our public verifiable Identity based Aggregate Signcryption (ID-ASC) Scheme for aggregation of different signcryptions by multiple users on distinct messages. We then provide the security model for our ID-ASC scheme.

3.1. Syntax of the ID-ASC scheme

An ID-ASC scheme consists of the following six algorithms.

Setup($k$): Given a security parameter $k$, the Private Key Generator (PKG) generates the master private key $M_{sk}$ and public parameters $Params$. $Params$ are made public while $M_{sk}$ is kept secret.

Key Extract($ID_i$): Given an identity $ID_i$ of any user, the public key $Q_{ID_i}$ and corresponding private key $S_{ID_i}$ of user computed by PKG, and sends it to the respective user through a secure channel.

Signcryption($m_i$, $ID_i$, $ID_B$, $S_{ID_i}$, $L$): In this algorithm, sender creates a signcryption of a message $m_i$ by taking the parameters the sender’s identity $ID_i$, the message $m_i$, sender’s private key $S_{ID_i}$, the receiver’s the identity $ID_B$ as input. And it generates valid signcryption $\sigma_i$ for the corresponding message $m_i$ from sender $ID_i$ to $ID_B$.

Aggregate Signcryption($\{\sigma_i, ID_i\}_{i=1,2,...,n}$): The algorithm takes the set of all signcryptions $\{\sigma_i\}_{i=1,2,...,n}$ and corresponding identity $ID_i$ as input from each sender with the identity $ID_i$; and outputs the aggregate signcryption $\sigma_{agg}$. This algorithm is executed by any one of the user or by any third party who is not in the senders list.

Aggregate Unsigncryption($\sigma_{agg}$, $ID_i$, $ID_B$, $S_{ID_i}$): By taking aggregate signcryption $\sigma_{agg}$, senders identity $\{ID_i\}_{i=1,2,...,n}$, private key $S_{ID_i}$ of the receiver B with the identity $ID_B$ as input, this algorithm outputs $m_i$ (for $i=1,2,...,n$) if $\sigma_{agg}$ is a “valid” signcryption text on message $m_i$ from sender $ID_i$ to $ID_B$ (for $i=1,2,...,n$) or “invalid”, otherwise.

Public Verify($\sigma_{agg}$, $ID_i$, $ID_B$): In this algorithm any third party can verify the validity legitimacy of the signcryption $\sigma_{agg}$ without any knowledge on the message $m_i$ used for the generation of the signcryption $\sigma_i$. It takes the aggregate signcryption $\sigma_{agg}$, the senders identity $\{ID_i\}_{i=1,2,...,n}$, and the receiver identity $ID_B$ as input and outputs “valid”, if $\sigma_{agg}$ is a valid signcryption or “invalid”, otherwise.

3.2. Security model for ID-ASC scheme

In this section, we present a security model for ID-ASC schemes. The ciphertext indistinguishability and the existential unforgeability security models are used to capture the confidentiality and authenticity requirements, respectively.

Unforgeability: An ID-ASC scheme is existentially unforgeable under adaptive chosen identity and chosen message attack (EUF-ID-ASC-CMA) if no probabilistic polynomial time adversary $A$ has non-negligible advantage in game defined in [24].
4. Proposed scheme

In this section, we present our ID-based Aggregate Signcryption (ID-ASC) scheme which uses bilinear pairings over elliptic curves. As discussed in section 3.1, our ID-ASC scheme consists of the following algorithms.

**Setup**: Given a security parameter $k$, the PKG chooses two cyclic groups $(G_1, +)$ and $(G_2, \cdot)$ of same prime order $q$, a generator $P$ of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_2$, $H_3 : \mathbb{Z}_q^* \rightarrow \mathbb{Z}_q^*$. PKG chooses a master key $s \in \mathbb{Z}_q^*$ and computes $P_{pub} = sP$ as a system public key. Finally, the PKG publishes the system parameters as $params = \{k, G_1, G_2, \hat{e}, P, P_{pub}, H_1, H_2, H_3\}$.

**Key Extract**: Given an identity $ID_i$, PKG computes $Q_{ID_i} = H_i(ID_i)$, and the respective private key $S_{ID_i} = sQ_{ID_i}$. Then PKG provides $S_{ID_i}$ to the corresponding user $ID_i$ over a secure channel.

**Signcryption**: To signcrypt a message $m_i$, each sender $ID_i$ performs the following by using his private key $S_{ID_i}$ and with receiver’s identity $ID_B$.

1. Selects a random $r_i \in \mathbb{Z}_q^*$, and compute $U_i = r_iP \in G_1$
2. Compute $K_i = \hat{e}(P_{pub}, Q_{ID_B})^{r_i} \in G_2$
3. Compute $c_i = H_2(K_i) \oplus (m_i \oplus ID_i)$
4. Compute $h_i = H_3(c_i, U_i, ID_i, Q_{ID_B})$
5. Compute $V_i = h_i S_{ID_i} + r_i P_{pub} \in G_1$.

On message $m_i$, the signcryption text from sender $ID_i$ to receiver $ID_B$ is $\sigma_i = (V_i, c_i, U_i)$.

**Aggregate Signcryption**: Any one of the sender or any third party can aggregate the signcryption. The aggregation is completed as follows after receiving $n$ individual signcryptions $\sigma_i = (V_i, c_i, U_i)$, where $i = 1, 2, \ldots, n$. Compute $V_{agg} = \sum_{i=1}^{n} V_i$ and send $\sigma_{agg} = ((ID_i, c_i, U_i)_{i=1,2\ldots,n}, V_{agg})$ as the aggregate signcryption to the receiver.

**Aggregate Unsigncryption**: Upon receiving the aggregate signcryption text $\sigma_{agg} = ((ID_i, c_i, U_i)_{i=1,2\ldots,n}, V_{agg})$, the receiver $ID_B$ uses his secret key $S_{ID_B}$ and perform the following to unsigncrypt $\sigma_{agg}$. For each sender $i$, the receiver $ID_B$ calculates the following:

1. Computes $K'_i = \hat{e}(U_i, S_{ID_B})$
2. Retrieves the plain text $m'_i \oplus ID_i = H_2(K'_i) \oplus c_i$
3. Computes \( h_i = H_3(c_i, U_i, ID_i, Q_{ID_b}) \)

4. Accept the message \( m'_i \) iff \( \hat{\epsilon}(P, V_{agg}) = \hat{\epsilon}(P_{pub}, \sum_{i=1}^{n}(h_iQ_{ID} + U_i)) \).

Public Verify: The authenticity of an aggregate signcryption \( \sigma_{agg} \) can be verified by any third party \( T \) as follows.

1. \( T \) computes \( h_i = H_3(c_i, U_i, ID_i, Q_{ID_b}) \)

2. Accept the signcryption \( \sigma_{agg} \) iff \( \hat{\epsilon}(P, V_{agg}) = \hat{\epsilon}(P_{pub}, \sum_{i=1}^{n}(h_iQ_{ID} + U_i)) \).

5. Analysis of the proposed scheme

In this section, first we present the proof of correctness and then we discuss the security and efficiency analysis of the proposed ID-ASC scheme.

5.1 Proof of Correctness

The following equations give the correctness of the proposed scheme.

\[
\hat{\epsilon}(P, V_{agg}) = \hat{\epsilon}(P, \sum_{i=1}^{n} V_i) \\
= \hat{\epsilon}(P, \sum_{i=1}^{n}(h_iS_{ID} + r_iP_{pub})) \\
= \hat{\epsilon}(sP, \sum_{i=1}^{n}(h_iQ_{ID} + r_iP)) \\
= \hat{\epsilon}(P_{pub}, \sum_{i=1}^{n}(h_iQ_{ID} + U_i)).
\]

5.2. Security Analysis

In the following we discuss the security requirements, as discussed in Section 3.2, of the proposed ID-ASC scheme.

Unforgeability

Theorem 1: The proposed ID-ASC scheme is secure against any EUF-ID-ASC-CMA adversary \( A \) under adaptive chosen identity and adaptive chosen message attack in the random oracle model if the CBDHP is hard in \( G_1 \).

Proof: Let \( C \) be a challenger, who is challenged with an instance of CBDHP say, \((P, aP, bP) \in \mathbb{G}_1^3\) for unknowns \( a, b \in \mathbb{Z}_q^* \). The aim of \( C \) is to compute \( \hat{\epsilon}(P, P)^{ab} \). Consider an adversary \( A \) who is capable of breaking the EUF-ID-ASC-CMA security of ID-ASC. \( C \) can make use of \( A \) to solve the CBDHP instance with non-negligible advantage in polynomial time as described below.

In the entire game, \( A \) will get the answers to the random oracles \( H_1, H_2, H_3 \) by consulting \( C \) and \( C \) desires to maintain hash lists \( L_1, L_2, L_3 \) and \( S_{list} \) that are initially empty and are used to keep track of answers to queries asked by \( A \) to oracle \( H_1, H_2, H_3 \) and \( S_{list} \). We assume that hash functions \( H_1, H_2 \) and \( H_3 \) were queried before signcryption.

- Setup: Adversary \( A \) receives the master public key \( P_{pub} = aP \) and the system parameters from the challenger \( C \).
Training Phase: Throughout the training phase, the adversary $A$ is permissible to access many oracles provided by $C$. $A$ can get sufficient training before generating the forgery. During the training phase, $C$ provided the following oracles to $A$.

- **Oracle $O_{H_1}(ID_y)$**: We will make a simplifying supposition that in each query, $A$ queries the $O_{H_1}$ oracle with different identities. Due to this supposition there is no loss of generality because, if the same identity is repeated, by definition, the oracle consults the list $L_4$ and gives the same response. Thus, we assume that $A$ asks $q_{H_1}$ distinct queries for distinct identities. Among $q_{H_1}$ identities, a random identity has to be selected as target identity and it is done as follows.
  
  $C$ selects a random index $\gamma$, where $1 \leq \gamma \leq q_{H_1}$. $C$ does not reveal $\gamma$ to $A$, when $A$ generates the $\gamma^{th}$ query as $ID_y$. $C$ fixes $ID_y$ as target identity for the challenge phase and $C$ responds to $A$ as follows.
  
  - If $i = \gamma$, then $C$ chooses random value $x_i \in \mathbb{Z}^*_q$ and responses with $Q_{ID_y} = x_i b P$. Here $C$ does not know $b$ since $C$ uses the $b P$ value given in the instance of CBDHP.
  - If $i \neq \gamma$ (for all other than above queries) then $C$ sets $Q_{ID_y} = x_i P$, returns $Q_{ID_y}$ as the response to the query to $A$, and $C$ stores $(ID_y, Q_{ID_y}, x_i)$ in the list $L_4$.

- **Oracle $O_{H_2}(K_i)$**: To answer $H_2$-queries, $C$ retains a hash list $L_2$ which contains $(K_i, \alpha_i)$. The following is the response given by $C$ when $A$ makes a query with input $K_i$.
  
  - If the query $ID$ already exists in the list $L_2$ then $C$ responds with $\alpha_i = H_2(K_i)$ to $A$.
  - Otherwise, $C$ picks a random number $\alpha_i \in \{0, 1\}^r$, responds $\alpha_i = H_2(K_i)$ to $A$ and add the tuple $(\alpha_i, K_i)$ to the list $L_2$.

- **Oracle $O_{H_3}(c_i, U_i, ID_y, Q_{ID_y})$**: A hash list $L_3$ contains $(c_i, U_i, ID_y, Q_{ID_y})$, which is maintained by $C$ to answer $H_3$-queries, and then $C$ replies as follows.
  
  - If $(c_i, U_i, ID_y, Q_{ID_y}, h_i)$ is available in the list $L_3$ then $C$ retrieves $h_i$ from the list $L_3$ and responds $h_i$ to $A$.
  - Otherwise $C$ choose a new random integer $h_i \in \mathbb{Z}^*_q$ and add the tuple $(c_i, U_i, ID_y, Q_{ID_y}, h_i)$ to the list $L_3$ and responds $h_i$ to $A$.

- **Oracle $O_{Key\ Extract}(ID_y)$**: In this oracle, $A$ makes a query for the private key $S_{ID_y}$ corresponding to an identity $ID_y$ (Note that $A$ should have performed the $O_{H_1}(ID_y)$ query before performing this query). For this query, $C$ replies as follows.
  
  - If $ID_y = ID_y$ then $C$ terminates.
  - Else, $C$ recovers $x_i$ corresponding to $ID_y$ from the list $L_4$, computes $Q_{ID_y} = x_i P_{pub}$ and returns $S_{ID_y}$ to $A$.

- **Oracle $O_{Signcrypt}(m_i, ID_y, ID_B)$**: The following is the response given by $C$, when $A$ makes a query for the signcryption of $m_i$ with the sender $ID_y$ and the receiver $ID_B$. 

Checks the list $L_4$ related to $ID_i$, if $i \neq \gamma$, then $C$ computes $(c_i, U_i, V_i)$ by using original signcryption scheme.

If $i = \gamma$, then $C$ executes the following.

- Randomly selects $r_i, h_i \in Z^*_q$ and sets $U_i = \langle r_i, P - Q_{ID_i} \rangle h_i$.

- Calculate $K_i = \hat{e}(U_i, S_{ID_i})$, and $c_i = H_2(K_i) \oplus (m_i \square ID_i)$ where $H_2(K_i)$ is obtained from the list $L_2$.

- If the tuple $(c_i, U_i, ID_i, Q_{ID_i}, h_i)$ already appears in the list $L_3$, $C$ chooses another $r_i, h_i \in Z^*_q$ and tries again. Otherwise $C$ computes $V_i = h_i r_i P_{pub}$, and stores $(c_i, U_i, ID_i, Q_{ID_i}, h_i)$ in the list $L_3$.

Signcryption of the message $m_i$ is $\sigma_i = (V_i, c_i, U_i)$.

Verification of signcryption:

$$\hat{e}(P_{pub}, h_i Q_{ID_i} + U_i) = \hat{e}(P_{pub}, h_i Q_{ID_i} + (r_i P - Q_{ID_i}) h_i) = \hat{e}(P_{pub}, h_i P) = \hat{e}(P, h_i h_{P_{pub}}) = \hat{e}(P, V_i).$$

$C$ outputs $\sigma_i = (V_i, c_i, U_i)$ and stores $(m_i, \alpha_i, U_i, V_i, ID_B)$ in the list $S_{list}$.

Oracle $O_{Agg unsigncrypt}(V_{agg}, (c_i, U_i, ID_i))_{i=1,2,\ldots,n}$: The following is the response by $C$ when $A$ makes aggregate unsigncation query with the signcryption $\sigma_{agg} = (V_{agg}, (c_i, U_i, ID_i))_{i=1,2,\ldots,n}$ and $ID_B$ as receiver, $C$ responds as follows.

- $C$ checks the list $L_4$ for the identity $ID_B$. If $ID_B \neq ID_r$, $C$ uses the Unsigncryption algorithm, since $C$ knows the secret key of the receiver $ID_B$.

- Otherwise, i.e. if $ID_r = ID_r$, targeted identity, then $C$ adopts the following steps.

  - $C$ obtains as input $\sigma_{agg} = (V_{agg}, (c_i, U_i, ID_i))_{i=1,2,\ldots,n}$. $C$ have the list of secret keys of all the users $ID_i_{i=1,2,\ldots,n}$.

    - From the list $L_2$, $C$ gathers the tuples of the form $(\alpha_i, K_i)$.

    - First $C$ checks whether the $ID_i$ belongs to the users list of $\sigma_{agg}$. Then $C$ finds $c_i \oplus \alpha_i$ and get $m_i \parallel ID_i$.

    - $C$ collects all such $m_i, ID_i$ pair corresponding to $\sigma_{agg}$. If at least $n$ such pairs are not present, then output “Invalid”.

    - $C$ finds the tuple $(c_i, U_i, ID_i, Q_{ID_i}, h_i)$ from the list $L_3$.

    - Checks whether $\hat{e}(P, V_{agg}) = \hat{e}(P_{pub}, \sum_{i=1}^n (h_i Q_{ID_i} + U_i))$, this signcryption is valid or not.

    - If the signcryption is valid, then $C$ outputs $m_i, ID_i$, for all $i=1,2,\ldots,n$, otherwise, output “Invalid”. 

− **Forgery:** A selects \( n \) senders with the identities \( \{ID_i\}_{i=1,2,...,n} \) and an identity \( ID_B \) as the receiver and outputs the aggregate signcryption \( \sigma_{agg}^* = (V_{agg}^*, (c_i, U_i, ID_i))_{i=1,2,...,n} \) to C. If at least one of the chosen identities is the target identity \( ID_B = ID_B \) and A has not requested for a signcryption query of the corresponding \( m_i, ID_i \) pair, then A succeeds the game.

C has \( V_{agg}^* \) and all the secret keys of the senders other than the target identity. Therefore C can compute, \( \hat{V}_{agg}^* \). C has \( \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i} = \sum_{i=1}^{n} \tau_i h_i P_{pub} + \sum_{i=1}^{k} S_{ID_i} \).

C has \( \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \).

C knows the values of \( h_i \) (from the list \( L_3 \)), \( \{U_i\}_{i=1,2,...,n} \), where \( U_i \) is available in the forged aggregate signcryption and \( P_{pub} \) is the master public key. Therefore, C can compute the inverse of the component in the above equation. By multiplying \( \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \) \( = \hat{V}_{agg}^* - \sum_{i=k+1}^{n} S_{ID_i}, P \).

C checks the list \( L_3 \) for the identity \( ID_B \). If \( ID_B = ID_B \), then C terminates the algorithm. Otherwise C performs the following steps.

**Confidentiality**

**Theorem 2:** The proposed ID-ASC scheme is secure against IND-ID-ASC-CCA2 adversary A under adaptive chosen message and adaptive chosen identity attack in the random oracle model if BDHP is hard in \( G_1 \).

**Proof:** Let C be the challenger, who is challenged with an instance of CBDHP say, \( (P, aP, bP, cP) \in G_1 \) for unknowns \( a, b, c \in \mathbb{Z}_q^* \). The aim of C is to compute \( \hat{e}(P, P)^{abc} \). Consider an adversary A who is capable of breaking the IND-IDASC-CCA2 security of ID-ASC. C makes use of A to solve BDHP instance with non negligible advantage in polynomial time as described below.

− **Setup:** This algorithm is run by the challenger C and find the public parameters as follows.

Choose two groups \( G_1 \) and \( G_2 \). C finds \( P_{pub} = aP \) and gives system parameters A.

− **Phase 1:** A can ask various types of queries to the oracles. All the oracles are same as unforgeability game of ID-ASC scheme.

− **Challenge:** After acquiring sufficient training, A submits the different messages \( (m_q, m_y, ID_i)_{i=1,2,...,n} \), with the identity \( ID_B \) as the receiver to C. C checks the list \( L_q \) for the identity \( ID_B \). If \( ID_q = ID_B \), then C terminates the algorithm. Otherwise C performs the following steps.

9
• C Picks a number \( k \in [1, n] \).
  
  • For each \( i \), C checks whether \( i = k \). If not, C picks a random number \( b \in \{0, 1\} \) and using the original signcryption algorithm, signcrypts message \( m_{h_i} \).
  
  • If \( i = k \) then, C
    
    ➢ Sets \( U_i^* = eP \).
    
    ➢ Computes \( c_i^* = H_2(K_i) \oplus (m_{h_i} \square ID_i) \), then C chooses \( V_i^* \).
    
    ➢ Finally, C generates the signcryption \( \sigma_i^* = (U_i^*, V_i^*, c_i^*) \) for A.
  
  • C gives the challenge aggregate signcryption \( \sigma_{agg}^* = (V_{agg}^*, (c_i^*, U_i^*, ID_i))_{i=1,2,...,n} \) to A.

  - **Phase 2:** This is also similar to phase 1, but in this phase, A cannot request for aggregate unsigncryption on the challenge aggregate signcryption \( \sigma_{agg}^* = (V_{agg}^*, (c_i^*, U_i^*, ID_i))_{i=1,2,...,n} \).

  - **Output:**
    
    After A has made adequate number of queries, for each \( i = 1 \) to \( k - 1 \), A yields the guess \( b'_i \).
    
    For the \( k^{th} \) output, if the adversary terminates then A has obtained that it is not an authenticated signcryption of either of the messages. (We suppose that the adversary is efficient for finding the targeted message). If so, C acquires \( (K_i, \alpha_i) \) from the list \( L_2 \) and outputs \( K_i^{U_{x_i}} \) as the solution to BDH problem where \( x_i \) is corresponding to \( ID_B \) in the list \( L_4 \).

    \[ K_i^{U_{x_i}} = e(U_1, S_{ID_B})^{U_{x_i}} = e(cP, x_iabP)^{U_{x_i}} = e(P, P)^{abc} \]

    Suppose that there are \( n \) such \( K_i^{U_{x_i}} \)s exist in the list \( L_2 \). One of them must be the solution to the BDH problem, without that hash value, A would not be capable to unsigncrypt the challenge ciphertext.

5.3 Efficiency Analysis

In this section, we compare the efficiency of our ID-ASC scheme with the related scheme [24, 25, 26]. We denote \( A_{G_1} \) by a point addition in \( G_1 \), \( M_{G_1} \) by a scalar multiplication in \( G_1 \), \( E_{G_2} \) by exponentiation in \( G_2 \) and \( P \) by a computation of pairing. In table-2, we compare the computational complexity of our scheme with the Selvi et al. IBAS scheme [24] in Signcryption and Aggregate Unsigncryption stages. From the Table-2, it is clear that the proposed ID-ASC scheme requires less number of pairings in aggregate unsigncryption but the other schemes requires one more pairing in aggregate unsigncryption. Hence our scheme is effective in computation.

By the aggregation of a part of the signcryption, the size of the signcryption is reduced by half. Hence the communication cost is highly reduced. Also, in our ID-ASC scheme, the verification of various signcryptions can be completed in a single step instead of performing each verification separately. This greatly reduces the cost involved in verification. Our scheme does not require any interactions between the senders before the signcryption generation, which reduces the communication complexity to a large extent.

Also constant number of pairings are required in aggregate verification stage and it requires only two pairings in our scheme but the Selvi et al. IBAS scheme III requires three pairings in aggregate verification. Thus the proposed ID-ASC scheme is more efficient in computational and communicational aspects than the existing ID-ASC schemes.
Table 2. Comparison with the related schemes

| Scheme          | Signcryp | Aggregate | Unsigncryp | Aggregate | Length of Cipher Text | Length of Aggregate signcryp | Length of Public Key | Length of Private Key | Public Verify |
|-----------------|----------|-----------|------------|-----------|-----------------------|-----------------------------|---------------------|----------------------|---------------|
| Selvi IBAS-1    | $nA_g + 2nM_g$ | $(n-1)A_g$ | $(n-1)A_g$ | $2|G| + |M|$ | $(n+1)|G|$ | $|G|$ | $|G|$ | No             |
| Selvi IBAS-3    | $nA_g + 3nM_g$ | $2(n-1)A_g$ | $(2n-2)A_g$ | $2|G| + |M|$ | $(n+1)|G|$ | $|G|$ | $|G|$ | No             |
| Xun-Yi Run J Kar | $2nE_g + nE_g$ | $2(n+1)E_g$ | $(n+1)E_g$ | $3|G| + |M|$ | $(2n+1)|G|$ | $|G| + |Z|$ | $|E| + |E|$ | No             |
| Our Scheme      | $nA_g + 2nM_g$ | $(2n-1)A_g$ | $(n-1)A_g$ | $2|G| + |M|$ | $(n+1)|G|$ | $|G|$ | $|G|$ | Yes            |

6. Conclusion

In this paper, we have proposed an ID-based Aggregate Signcrypion (ID-ASC) scheme with public verifiability, which uses the K.A. Shim’s aggregate signature scheme [22] as the base scheme. The proposed aggregate signcrypion scheme can be done by the aggregation of $n$ different signcryptions on $n$ different messages signed by $n$ multiple users. We have verified that the proposed aggregate signcrypion scheme is unforgeable and is CCA-2 secure in the random oracle model. Our ID-ASC scheme requires constant number of pairings for aggregate verification, which significantly improves the efficiency of the system. Also our scheme does not involve any kind of communication among the signers before the aggregation, which reduces the communication complexity to a large extent.

7. References

[1] Zheng Y 1997 Digital Signcrypion or How to Achieve Cost (Signature and Encryption) Cost (Signature)+ Cost (Encryption) Advances in Cryptology, LNCS 1294 pp 165–179
[2] Shamir A 1984 Identity-Based Cryptosystems and Signature Schemes Advances in Cryptology-Crypto 196 pp 47-53
[3] Boneh D and Franklin M 2001 Identity-Based Encryption from the Weil Pairing avances in Cryptology-Crypto 2139 pp 213-229
[4] Beak J, Newmarch J et.al. 2004 A Survey of Identity-Based Cryptography In Proc. of the 10th Annual Conference for Australian Unix User’s Group pp 95-102
[5] Gorantla M C, Gangi Setti R and Saxena A 2005 A Survey on ID-based Cryptographic Primitives IACR, Cryptology e-print Archive, 094 http://eprint.iacr.org
[6] Malone Lee J 2002 Identity based Signcrypion In Cryptology e-Print Archive Report 2002/098
[7] Libert B and Quisquater J 2003 A New Identity Based Signcrypion Scheme from Pairings In IEEE Information Theory Workshop pp 155-158
[8] Chow S S M et al. 2003 Efficient Forward and Provably Secure ID-based Signcrypion Scheme with Public Verifiability and Public Ciphertext Authenticity In ICISC, 2971 pp 352–369
[9] Chen L and Lee J M 2005 Improved Identity-Based Signcryption In Proc. Of Public Key Cryptography 3386 pp 362–379
[10] Barreto P S L M et al. 2005 Efficient and Provably-Secure Identity-based Signatures and Signcryption from Bilinear maps In Advances in Cryptology-ASIACRYPT 3788 pp 515–532
[11] Selvi S S D, Vivek S S and Rangan C P 2010 Identity based Public Verifiable Signcryption Scheme 6402 pp 244-260
[12] Fagen Li and Khan M K 2011 A Survey of Identity-Based Signcryption IETE Technical Review 28(3) pp 265-272
[13] Kent S, Lynn C and Seo K 2000 Secure Border Gateway Protocol (Secure-BGP) IEEE Journal on Selected Areas in Communications 18 (4) pp 582–592
[14] Boneh D et al. 2003 Aggregate and Verifiably Encrypted Signatures from Bilinear Maps EUROCRYPT 2656 pp 416-432
[15] Boneh D, Lynn B, Shacham H 2002 Short Signatures from the Weil Pairing Advances in Cryptology: Asiacrypt’01 2248 pp 514–532
[16] Cheon J H et al. 2004 A New ID-Based Signature with Batch Verification Available from http://eprint.iacr.org/2004/131
[17] Xu Y, Zhang Z and D. Feng 2005 ID-based Aggregate Signature from Bilinear Pairings Proceedings of the 4th International Conference on Cryptology and Network Security pp 110-119
[18] Herranz J 2006 Deterministic identity-based signatures for partial aggregation Computer Journal 49(3) pp 322-330
[19] Gentry C and Ramzan Z 2006 Identity-based Aggregate Signatures Proceedings of PKC’06 3958 pp 257–273
[20] Wang Z et al. 2008 Practical Identity-based Aggregate Signature Scheme from Bilinear Maps Shanghai Jiao Tong University Press 13(6) pp 684–687
[21] Selvi S S D et al. 2010 Efficient and Provably Secure Identity Based Aggregate Signature Schemes with Partial and Full Aggregation Cryptology e-Print Archive Report 461 http://eprint.iacr.org/2010/461
[22] Shim K A 2010 An ID-based Aggregate Signature Scheme with Constant Pairing Computations The Journal of Systems and Software 83 pp 1873-1880
[23] Yu Y 2011 An Identity based Aggregate Signature from Pairings Journal of Networks 6(4) pp 631-637
[24] Selvi S S D et al. 2009 Identity Based Aggregate Signcryption Schemes, INDOCRYPT 5922 pp 378-397
[25] Xun-Yi Ren, Zheng-Hua Qi and Geng 2012 Provably Secure Aggregate Signcryption Scheme ETRI journal 34(3) pp 421-428
[26] Kar J, Provably Secure Identity based Aggregate Signcryption Scheme in Random Oracle Model International Journal of Network Security 17(5) pp 580-587