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
(Email: jayaprakashkar@yahoo.com)
(Received Jan. 26, 2013; revised and accepted Jan. 5 & Feb. 6, 2014)

Abstract

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

Keywords: Aggregate signature, BDHP, bilinear pairing, random oracle model

1 Introduction

In 1997, Zheng [22] introduced signcryption where signature and encryption are performed simultaneously in one logical step 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. Baek *et al.* [1] shows Zheng's original schemes is provably secure in formal security model. Authentication, Confidentiality, non-repudiation and integrity are the strong security goals for many cryptographic applications. Applications must often contain at least two cryptographic primitives: signature, and encryption, which will definitely increase the corresponding computation and implementation complexity and even will be infeasible in some resources-constrained environments. To implement on low processor devices, Han *et al.* [6] introduced generalized signcryption scheme. It is feasible to implement joint encryption and signature functions in a single primitive.

2 Previous Works

Zheng [22] devised the principle of signcryption where both these encryption and signature are gained in a single logical step. Identity based cryptography was introduced by Shamir [15] in 1984 without obtaining the certificates for their public keys. In alternate, public keys are constructed taking user's IP address, telephone no, email addresses, social security numbers that distinctively identifies a user [9]. Trusted Third Party called Certificated Authority (CA) or Private Key Generator (PKG) generates the private key correspond to public key. Identity-based cryptography is supposed to provide a more suitable to traditional Public Key Infrastructure(PKI). Several practical identity-based signature schemes were proposed since 1984 with some vulnerability.

In 2001, Boneh and Franklin [2] first introduced fully practical identity based encryption scheme. Subsequently, many ID-based signcryption schemes have been proposed [7, 8, 10, 20, 21]. Yu *et al.* [18] proposed the first Identity based signcryption scheme in the standard model. But it was proved, that are insecure [16, 17, 19]. Also later on the schemes [18, 19] have proven these are insecure.

In 2002, Malone-Lee [12] proposed an efficient IBSC scheme by joining the function of of identity-based cryptography and signcryption. But this scheme is not semantically secure due to the visibility of the signature in the signcrypted message. This is proven by Libert and Quisquater [11]. Subsequently, Libert and Quisquater also proposed three different types of IBSC schemes which suit either forward security or public verifiability. Therefore to design an efficient signcryption scheme that proves both forward security and public verifiability was a great challenge in research community. To provide both the forward security and public verifiability, Chow *et al.* [5] constructed an Identity based Signcryption scheme. Boyen [3] proposed an IBSC scheme that provides ciphertext unlinkability and anonymity along with public verifiability and forward security. The improved version of this scheme was proposed by Chen and Malone-Lee [4] Barreto

et al. [13] which is provably secure and more efficient.

3 Preliminaries

3.1 Notation

Definition 1 [Bilinearity.] *Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic group under the operation addition and multiplication. Both the groups are of same prime order p. Let e be an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:*

- **Bilinearity***: Let $P, Q \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, $e(aP, bQ) = e(P, Q)^{ab}$, 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 $\mathbb{G}_2$. $\exists\ P, Q \in \mathbb{G}$ such that $e(P, Q) \neq 1_{\mathbb{G}_2}$.*
- **Computability***: $\exists$ algorithm that compute $e(P, Q)$ in efficient way $\forall\ P, Q \in \mathbb{G}_1$.*

3.2 Mathematical Assumptions

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

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

Definition 4 [Bilinear Diffie-Hellman Problem.] *Let the algorithm $\mathcal{G}(k)$ generates 5 tuples $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$. Where $a, b, c \in \mathbb{Z}_q^*$. The problem in the group $\mathbb{G}$ is defined as: 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 $\nexists$ algorithm $\mathcal{A}$ running in time at most t such that*

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

The probability is to find out taking all possible choices of (a, b, c) and is measured over the internal random operation of $\mathcal{A}$ and random choices of $a, b, c \in \mathbb{Z}_q^*$. Let us assume that BDHP is computationally infeasible to solve. Let the magnitude of q is $2k$, where k denotes a security parameter. There does not exist a polynomial time (in k) algorithm which has a non-negligible advantage in solving the BDHP, for all values of sufficiently large k. Following are the two variations of BDHP [9].

Definition 5 [Decisional Diffie-Hellman Problem.] *Let the probability is to find out taking all choices of (a, b, c, h). $\mathcal{G}(k)$ generates 5-tuples $(q, \mathbb{G}, \mathbb{G}_T, e, P)$. The problem is defined in the group $\mathbb{G}$ is given (P, aP, bP, cP, r) with some $a, b, c \in \mathbb{Z}_q^*$, if $r = e(P, P)^{abc}$ return* **yes***, otherwise* **no***. Where $a, b, c, r \in \mathbb{Z}_q^*$. The DBDHP in The (t, ϵ)-HDDH assumption holds in $\mathcal{G}$ if $\nexists$ an algorithm $\mathcal{A}$ with running time at most t such that*

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

Definition 6 [Hash Decisional Diffie-Hellman Problem.] *Let $\mathcal{G}(k)$ generates 5-tuple $(q, \mathbb{G}, \mathbb{G}_T, e, g)$. $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ is a hash function, whether l is a security parameter, and let $a, b \in \mathbb{Z}_q^*, h \in \{0,1\}^l$, HDDH problem in $\mathbb{G}$ is defined as: Given (P, aP, bP, cP, h), decide whether it is a hash Diffie-Hellman tuple $((P, aP, bP, cP\mathcal{H}(e(P, P)^{abc}))$. Return 1, if it is correct, otherwise return 0. The (t, ϵ)-HDDH assumption holds in $\mathcal{G}$ if $\nexists$ 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) comprises following probabilistic polynomial time solvable algorithms:

- **Setup**: $(param, msk) \leftarrow \text{Set}(1^k)$ takes $k \in \mathbb{N}$ the security parameter and generates $mask$ master secret key and $param$ global public parameters.
- **Key Extract**: $(<S_{ID_i}, d_i>, P_{pub}, q_i) \leftarrow Ext(1^k, param, msk, ID_i)$ takes $param$ global parameters, msk master secret key, k security parameter and identity of the sender ID_i to generate a private key $<S_{ID_i}, d_i>$ and public key P_{pub} and q_i.
- **Signcrypt**: $\sigma_i \leftarrow Signcrypt(1^k, param, m_i, X_i, d_i, ID_i, ID_B)$ takes k security parameter, $param$ global parameter 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 space of message, space of signcrypted message 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 k a security parameter, $param$ the global parameters, σ_{agg} aggregate signcryption, S_{ID_B} receiver's secret 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 k a security parameter, $param$ global parameters, m the message, Z_{agg} the signature and the

private key $<ID_B, d_B>$ outputs $Valid$ or $\perp$ for invalid signature.

5 Security Notions

Security of signcryption comprises two distinct techniques: providing authenticity and confidentiality or privacy. The two security goals can be provided by digital signature and encryption respectively. Under chosen ciphertext, we can say the indistinguishability of ciphertext with signature (signcrypt) or under chosen message attack, existential unforgeability of signcrypt. To achieve high level security, we concentrate on the above two forms of security.

Definition 7 [Confidentiality.] *An Identity-based signcryption scheme is said to be semantically secure or has indistinguishability against adaptive chosen ciphertext attack (IND-IDASC-CCA2) is there does not exist an adversary of polynomial bounded (PPT) with non-negligible advantage in the following game.*

1) **Initial**: **Setup** *is run by the challenger $\mathcal{C}$ taking the input of security parameter k. It returns the system parameter $param$ and master secret key msk. Master secret key msk is kept secret and send the system parameter $param$ to the adversary $\mathcal{A}$. The adversary $\mathcal{A}$ submits queries of polynomial bounded number of times to the oracles given to $\mathcal{A}$ by $\mathcal{C}$. In the first phase, execution of the queries are scheduled below:*

   - **Extraction oracle**: $<S_{ID_i}, d_i> \leftarrow Ext(mask, ID_i)$. *$\mathcal{A}$ submits ID_i extraction oracle and corresponding to the identity ID_i, get $<S_{ID_i}, d_i>$ as the private key pairs.*
   - **Signcryption oracle**: *$\mathcal{A}$ submits a message m_i, signer identity ID_i, and 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 $Signcrypt(m_i, d_i, ID_i, ID_B)$ to obtain the signcryption σ_i. 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 $\mathcal{C}$. $\mathcal{C}$ produces pair $<S_{ID_B}, d_B>$ as private key by submitting queries to 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. $\mathcal{A}$ submits adaptively the queries to the oracle.*

2) *Let messages m_{i0}, m_{i1} are chosen by $\mathcal{A}$. Identities $\{ID_i\}_{i=1...n}$ and ID_B of sender and receiver on which $\mathcal{A}$ would like to be challenged. Two random bit $b \in \{0,1\}$ are chosen by the challenger $\mathcal{C}$ 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 $\mathcal{A}$.*

3) *Initially $\mathcal{A}$ performs polynomially bounded number of new queries with the restrictions that $\mathcal{A}$ cannot submit query to UnSigncryption oracle for the unsigncryption of σ_{agg}^* or the Keygen oracles for the private keys pairs of ID_B^*.*

4) *$\mathcal{A}$ returns a bit b' and if $b' = b$, then wins the game at the end of the game. The success probability is:*

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

where Adv denotes advantage for the adversary.

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 has a non-negligible advantage in the following game:*

1) *The algorithm* **Setup** *is run by the challenger $\mathcal{C}$ taking input k as security parameter and sends $param$ the system parameters to the adversary $\mathcal{A}$ and keeps secret $mask$ the master private key.*

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 run in adaptive manner.*

   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 it 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 k security parameter, $param$ the global parameters, σ_{agg} aggregate signcryption, 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

$$Adv_{\mathcal{A}}^{EUF-IDASC-CMS} = Pr[Verify(m_i^*, \sigma^*, \{ID_i^*\}_{i=1...n}) = Valid]$$

Definition 9 [Ciphertext Unforgeability.] *An ID-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* **Setup** *algorithm is run by the challenger $\mathcal{C}$ taking the input k as security parameter and sends param the system parameters to the adversary $\mathcal{A}$ and keeps secret msk the master private key.*

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 in adaptive manner and allows as in queries described in (IND-IDASC-CCA2) game.*

3) *Forgery. The adversary $\mathcal{A}$ produces a new aggregate signcryption σ_{agg} from a set $\{ID_i\}_{i=1\ldots n}$ of n users on messages $m_i, \forall i = 1\ldots n$ to a final receiver $ID_B \notin \{ID_i\}_{i=1\ldots n}$, where the private keys of the users in $\{ID_i\}_{i=1\ldots n}$ 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 ID-based Aggregate Signcryption Scheme

The scheme comprises five randomized polynomials algorithms.

- **Setup.** The algorithm take k the security parameter. Groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q are chosen by **PKG**. 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 $P_{pub} = sP$. System parameters are published by **PKG**.

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

- **Extract.** The algorithm follows
  - 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 sends 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 $r_i \leftarrow_R \mathbb{F}_q^*$ randomly and calculate $W_i = r_i \cdot P, w_i = \hat{e}(P_{pub}, Q_{ID_B})^{r_i}$.
  - 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 $h_{2i} = \mathcal{H}_2(m_i, ID_i, X_i, w_i, ID_B, X_B)$.
  - Verify the validity of the following equation

$$\begin{aligned}
w_i &= \hat{e}(W_i, S_{ID_B}) = \hat{e}(r_i P, S_{ID_B}) \\
&= \hat{e}(P, S_{ID_B})^{r_i} \\
&= \hat{e}(P, sQ_{ID_B})^{r_i} = \hat{e}(sP, Q_{ID_B})^{r_i} \\
&= \hat{e}(P_{pub}, Q_{ID_B})^{r_i}.
\end{aligned}$$

$$\begin{aligned}
Z_{agg} &= \sum_{i=1}^{n}(v_i \cdot P) = \sum_{i=1}^{n}(r_i h_{2i} + h_{3i} d_i) \cdot P \\
&= \sum_{i=1}^{n} h_{2i}(r_i \cdot P) + \sum_{i=1}^{n} h_{3i}(d_i \cdot P) \\
&= \sum_{i=1}^{n} h_{2i}(r_i \cdot P) + \sum_{i=1}^{n} h_{3i}(x_i + sq_i) \cdot P \\
&= \sum_{i=1}^{n}(h_{2i} W_i) + \sum_{i=1}^{n}(h_{3i} X_i) \\
&\qquad + P_{pub} \sum_{i=1}^{n}(h_{3i} q_i).
\end{aligned}$$

7 Proof of Correctness

7.1 Security Analysis

Our scheme is secure in IDASC-IDASC-CCC2, AUTH-IDASC-CMA and EUF-IDASC-CMA defined in Definitions 7, 8 and 9. We prove the following theorem as proved in [14].

Theorem 1 *In random oracle model, we assume the adversary $\mathcal{A}$ for $IND-IDASC-CCA2$ is 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 and decide validity of $\mu = \hat{e}(P,P)^{abc}$. $\mathcal{B}$ executes $\mathcal{A}$ as a subroutine and proceeds $\mathcal{A}$s challenger in the IND-IDASC-CCA2 game. A lists L_0, L_1, L_2 and L_3 are set by $\mathcal{B}$. These are initial empty. $\mathcal{A}$ submits queries to the respective oracles $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ and place the answers in the corresponding list.

- **Oracle Simulation**:

  1) $\mathcal{H}_0$**-Oracle**: At the beginning of the game $\mathcal{B}$ submits the system parameters to $\mathcal{A}$ by computing $P_{pub} = cP$ ($\mathcal{B}$ does not know c and act function of master-key). Then $\mathcal{B}$ chooses two distinct random numbers $i, j \in \{1 \dots q_{\mathcal{H}_0}\}$. $\mathcal{A}$ 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 $ID_i \neq \{ID_i\}_{i=1\dots n}$ $\mathcal{B}$ 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 $S_{ID_i} = x_i(aP) = a(x_i P) = aX_i$ 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 $\sigma_{agg} = <\{C_i, W_i, X_i, ID_i\}_{i=1,\cdots,n}, Z_{agg}>$ 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 return ciphertext of the challenged signcryption running $Signcrypt(m_{ib^*}, X_i, d_i, ID_i, ID_B)$ and $Aggregate(\{\sigma_i^*, ID_i\}_{i=1\dots n})$, then returns σ_{agg}^* to $\mathcal{A}$.

- **Output**: $\mathcal{A}$ returns the presumed bit after submitting adequate number of queries. Then *mathcalB* solve BDH problem and returns '1'. Else, it returns '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 *Assume Elliptic Curve Discrete Logarithm Problem is computationally infeasible to solve in* $\mathbb{G}_1$. *The proposed ASC is secure against any probabilistic polynomial time adversary* $\mathcal{A}$ *for AUTH-IDASC-CMA in the random oracle model.*

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 $\mathcal{A}$ submits 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 no entry for ID_i belongs to the L_0-List, 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 *Assume Decisional Biliner Diffie-Hellman Problem is computationally infeasible to solve in* $\mathbb{G}_1$. *The proposed ASC is secure against any probabilistic polynomial time adversary* $\mathcal{A}$ *for EUF-IDASC-CMA in the random oracle model.*

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

Keygen Oracle: When $\mathcal{A}$ submits 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 no entry for ID_i belongs to the L_0-List, return $\perp$. Otherwise, if $ID_i = ID_\alpha$, $\mathcal{B}$ recovers $<ID_i, X_i, q_i, x_i>$ 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.

Let the valid signcryption σ_i is sent from ID_i to ID_B which is, a message m_i is generated by the *Unsigncrypt* algorithm. $\mathcal{B}$ submits the queries to the oracle by applying replay technique return 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. With the same random tape but with a different hash value, this is provided by running the truing machine again $\mathcal{B}$ 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 Comparison

Let symbolize confidentiality (Con), unforgeability (Unf), public verifiability (PuV), forward security (FoS), ciphertext unlinkability (CiU) and ciphertext anonymity (CiA). "√" and "×" denotes Yes and No respectively. Table 1 shows the security comparison among IDASC and others.

Efficiency of aggregate sincryption scheme can be evaluated with respect to computational cost and ciphertext length [14]. To compute the computational cost, we consider scalar multiplications, exponentiations and pairing

Table 1: Security comparison

Schemes	$Conf$	Unf	PuV	Fos	CiU	CiA
Libert and Quisquater(I)	√	√	√	√	×	×
Libert and Quisquater(II)	√	√	√	√	×	×
Libert and Quisquater(III)	√	√	√	×	√	×
Malone-Lee	×	√	√	√	×	×
Barreto *et al.*	√	√	√	√	×	×
Boyen	√	√	√	√	√	√
Chow *et al.*	√	√	√	√	×	×
IDASC	√	√	√	√	√	√

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(I)	1(+1)	2	2	4		2
Libert and Quisquater(II)	1(+1)	2	2	4		2
Libert and Quisquater(III)	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	

Table 3: Comparison of ciphertext size

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

operation are costly operation. Let scalar multiplication in $\mathbb{G}_1$ is denoted by $(Mul(\mathbb{G}_1))$, exponentiations in $\mathbb{G}_2$ is $(Exp(\mathbb{G}_2))$, and pairing operations $(Pairing)$. Tables 2 and 3 show the comparison among IDASC and others with computational cost and ciphertext size items, respectively.

9 Conclusion

Here we have proposed an efferent and secure aggregate signcryption scheme which is more efficient than the scheme proposed by Xun-Yi Ren *et* al. [14] with respect to the length of Ciphertext and secure than the other schemes summarized in the tables. We prove that the scheme in Random oracle model and proven that the scheme achieve the three strong security goals confidentiality, signature unforgeability and ciphertext unforgeability under the assumption, ECDLP and BDHP are computationally hard. 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] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of Advances in Cryptology (Crypto'01)*, LNCS 2139, pp. 213–229, Springer-Verlag, 2001.

[3] X. Boyen, "Multipurpose identity-based signcryption," in *Proceedings of Advances in Cryptology (Crypto'03)*, LNCS 2729, pp. 383–399, Springer-Verlag, 2003.

[4] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Proceedings of Public Key Cryptography (PKC'05)*, LNCS 3386, pp. 362–379, Springer-Verlag, 2005.

[5] 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 *Security and Cryptology (ICISC'03)*, LNCS 2971, pp. 352–369, Springer-Verlag, 2004.

[6] Y. Han and X. Yang, "Elliptic curve based generalized signcryption scheme," Technical Report IACR Archive ePrint-2006/126, Sep. 2006.

[7] Z. P. Jin, Q. Y. Wen, and H. Z. Du, "An improved semantically-secure identity-based signcryption scheme," *Standard model Computers & Electrical Engineering*, vol. 36, no. 2, pp. 545–552, 2010.

[8] J. Kar. "An efficient signcryption scheme from q-diffe-hellman problems,". Technical Report IACR Archive ePrint-2012/483, Sep. 2012.

[9] J. Kar, "Provably secure identity based online/offline signature scheme for wireless sensor network," *International Journal of Network Security*, vol. 16, no. 1, pp. 29–39, 2014.

[10] F. Li, X. Xin, and Y. Hu, "ID-based signcryption scheme with (t, n) shared unsigncryption," *International Journal of Network Security*, vol. 3, no. 2, pp. 155–159, 2006.

[11] B. Libert and J. J. Quisquater, "A new identity based signcryption schemes from pairings," in *Proceedings of IEEE Information Theory Workshop*, vol. 435, pp. 155–158, Paris, France, June 2003.

[12] J. Malone-lee, "Identity-based signcryption," in *Proceedings of Public Key Cryptography*, LNCS 3386, pp. 362–379, Springer-Verlag, 2005.

[13] N. McCullagh P. S. L. M. Barreto, B. Libert and J. J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Proceedings of Advances in Cryptology (Asiacrypt'05)*, LNCS 3788, pp. 383–399, Springer-Verlag, 2005.

[14] X. Y. Ren, Z. H. Qi, and Geng, "Provably secure aggregate signcryption scheme," *ETRI Journal*, vol. 34, no. 3, pp. 421–428, 2012.

[15] A. Shamir, "Identity-based cryptosystem and signature schemes," in *Proceedings of Advances in Cryptology (Crypto'84)*, LNCS 3386, pp. 47–53, Springer-Verlag, 1984.

[16] M. Toorani and A. A. B. Shirazi, "Cryptanalysis of an elliptic curve-based signcryption scheme," *International Journal of Network Security*, vol. 10, no. 1, pp. 51–56, 2010.

[17] X. Wang and H. F. Qian, "Attacks against two identity-based signcryption schemes," in *Proceeding of Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 1, pp. 24–27, Paris, France, June 2010.

[18] Y. Sun Y. Yu, B. Yang and S. Zhu, "Identity based signcryption scheme without random oracles," *Elsevier-Computer Standards & Interfaces*, vol. 31, no. 1, pp. 56–62, 2009.

[19] R. Yanli and G. Dawu, "Efficient identity based signature/signcryption scheme in the standard model," in *Proceeding of First International Symposium on Data, Privacy, and E-Commerce*, vol. 3386, pp. 133–137, Paris, France, June 2007.

[20] G. Yu, X. X. Ma, and Y. Shen, "Provable secure identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 411, no. 40, pp. 3614–3624, 2010.

[21] B. Zhang and 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*, vol. 1294, pp. 15–27, Chengdu, China, June 2010.

[22] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost(signature) + cost (encryption)," in *Proceeding of Advances in Cryptology (Crypto'97)*, LNCS 1294, pp. 165–179, Springer-Verlag, 1997.

Jayaprakash Kar has received his M.Sc and M.Phil in Mathematics from Sambalpur University, M.Tech and Ph.D in Computer Science (Cryptographic Protocols) from Utkal University, India. Currently he is working as Assistant Professor in the Department of Information Systems, Faculty of Computing and Information Technology. He is actively associated with Information Security Research Group, King Abdulaziz University, Saudi Arabia. His current research interests is on development and design of provably secure cryptographic protocols and primitives using Elliptic Curve and Pairing based Cryptography includes digital signature, Signcryption Scheme, Key management problem of broadcast encryption, Deniable authentication protocols, Proxy Blind Signature scheme. He has 01 monograph, 04 book chapters and more than 32 Journal papers and Conference articles to his credit. Dr. Kar is member of advisory and editorial board of many peer reviewed Journals. He is life member of International Association for Cryptology Research (IACR), Cryptology Research Society of India, International Association of Computer Science & Information Technology (Singapore) and International Association of Engineers (United Sates).