Efficient and Provably Secure Identity Based Aggregate Signature Schemes With Partial and Full Aggregation

S. Sharmila Deva Selvi[1], S. Sree Vivek[1,⋆⋆], J.Shriram[2], C. Pandu Rangan[1,⋆]

[1]Department of Computer Science and Engineering,
Indian Institute of Technology Madras.
sharmila@cse.iitm.ac.in,svivek@cse.iitm.ac.in, prangan@iitm.ac.in.
[2]National Institute of Technology Trichy, India

Abstract. An identity based signature allows users to sign their documents using their private keys and the signature can be verified by any user by using the identity of the signer and public parameters of the system. This allows secure communication between the users without any exchange of certificates. An aggregate signature scheme is a digital signature scheme which allows aggregation of different signatures by different users on different messages. An aggregate signature on n messages m_i by n users U_i convinces the verifier that each user U_i has signed the corresponding message m_i. The primary objective of the aggregate signature scheme is to achieve both computational and communication efficiency. Here we discuss two identity based aggregate signature schemes. The first aggregate scheme IBAS-1 uses a variation of light weight Schnorr based signature. IBAS-1 does not involve any pairing operations in signature verification. IBAS-1 is computationally efficient since it avoids the costlier operation in elliptic curve groups(pairings). Also because of the light weight property of IBAS-1, it is much suitable for practice. The second aggregate signature scheme IBAS-2, which also has Schnorr type key construct, achieves full aggregation of signatures without agreeing on common randomness and without having any kind of interaction among all the signers. IBAS-2 achieves communication efficiency. But the computational complexity of IBAS-2 is higher than the IBAS-1 because it involves bilinear pairing.

Keywords: Identity Based Signature, Aggregate Signature, Random Oracle Model, Provable Security.

1 Introduction

The idea of identity based cryptography is to derive the public key of the user using the identity, which uniquely defines the user. This reduces the overhead of storage of certificates that are typical in public key systems. Identity based cryptography was introduced by Shamir [15]. Since then, several identity based signature schemes have been proposed [16] [3] [20] [10]. Different variations such as proxy, group, ring signatures etc. have been proposed in the identity based settings depending on various practical applications.

One such variation of identity-based signature is Aggregate Signature. The major requirements in the recent scenario of technological development are computation and communication efficiency. In any communication network, the bandwidth is the limiting constraint. One must make sure that during the design of aggregate signature scheme the communication efficiency is improved by reducing the amount of data to be communicated. Another limiting constraint is the cost involved in verifying the aggregate signature. Decreasing either the computation or the communication cost or both makes the aggregate signature scheme highly efficient.

Using aggregate signature schemes, signatures from different users on different messages can be aggregated into a single compact signature. The desired property of aggregate signature is that an adversary should not be able to extract a single signature from the aggregated signature. Aggregate signatures are of two kinds. In the first type the signatures can be aggregated in any order and aggregation can be done by any user(signer or a third party) in the system. The second type is called sequential aggregation, where each

⋆ Work supported by Project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation sponsored by Department of Information Technology, Government of India

user aggregates his signature to the previous aggregated signature. Sequential aggregation is a weaker model compared to the former model.

Application of aggregate signature include traffic control, military applications, banking transactions and also for ordinary business use. Certificate chains in hierarchical PKI systems consists of various signatures at different levels in the hierarchy. By using aggregate signature one can combine all these signatures and thus reduce the certificate length. Sequential aggregation is used in communication between the routers in a network where each router receives the data and signature of the previous router. It aggregates its own signature to the previous aggregate signature and routes it to the next router. This aggregated signature can be used to find the path travelled by the data from source to destination by using a single aggregate signature. Aggregate signatures can also be used in wireless network scenarios. Since the major constraint in wireless networks is communication complexity, the use of efficient aggregate signature helps in reducing the amount of data to be communicated.

A number of aggregate signature schemes have been proposed in literature. Some of them achieve partial aggregation and some achieve full aggregation. An aggregate signature scheme is claimed to achieve partial aggregation if a part of the signature is aggregated, namely the part with the secret key component of signers is fully aggregated and the randomness part is propagated without aggregation. If both the parts in the signature are fully aggregated then the scheme is said to achieve full aggregation. We provide a brief survey about the efficiency and weakness various identity based aggregate signature schemes.

Survey of Existing Schemes : Currently, in literature we have number of identity based aggregate signature schemes and batch verification schemes. Though batch verification schemes do not exactly morph the aggregation technique, it has a similar goal of reducing the computational complexity. So we have included batch verification schemes in our discussion.

Shi et al. proposed an efficient identity based signature scheme [5] with batch verification. Though the scheme in [5] achieves efficiency in computation with just two pairing operations and linear exponentiation operations, it is required to pass all the signatures separately and hence increases the communication complexity. Also a universal forgery of the signature of any singer is possible in this scheme as shown in [14].

Wang et al. designed an identity based aggregate signature [16] and it is claimed to be the most efficient scheme. It uses constant pairing operation for signature verification. But the aggregate signature in this scheme [16] is not secure since universal forgery of signature of any user is possible in this scheme. Also, the scheme achieves only partial aggregation. The attack in Wang et al. scheme [16] is shown in [14].

Xiangguo et al. gave a aggregate signature scheme [4] which uses the BLSR scheme [2] as the base signature scheme. In this scheme all the signers have to broadcast their own random values used for singing to all the cosigners so that everyone agrees upon a common randomness before the generation of aggregate signature. This result in quadratic communication complexity which is a big overhead. Mutual interaction between all the signers is not a desirable step in aggregate signatures.

Hyo et al. gave a number of batch verification techniques [19]. In their paper, the type 3 batch verification is the scheme whose properties have close resemblance to that of aggregate signatures. Only partial aggregation is achieved in the scheme [19]. Also, during verification it requires linear number of pairings which also increases the computation complexity considerably.

Yiling et al. proposed an efficient aggregate signature scheme with full aggregation and constant pairing operations in [17]. But the scheme in [17] is not secure since universal forgery of the base signature scheme used in [17] is possible as shown in [14].

Javier Herranz came up with an identity based signature scheme [9] with partial aggregation. But his scheme produces deterministic signature where the signature component on a message will always be the same. This is a major draw back in real world scenarios. It also uses linear number of pairing operations

leading to inefficiency in computation.

Xu et al. in [18] proposed an identity based aggregate signature scheme. This scheme uses Sakai et al.'s signature construct as the base signature scheme. This achieves only partial aggregation and also requires linear number of pairings during signature verification.

Gentry and Ramzan proposed an efficient identity based aggregate signature scheme [8]. This scheme achieves both full aggregation and also constant number of pairing operations during signature verification. But the scheme in [8] has certain weaknesses which makes it unsuitable for real life scenarios. The weaknesses of the scheme are briefly reported in the appendix.

Boldyreva et al. proposed an identity based sequential signature scheme [1]. Hwang et al. in [11] proposed an attack on [1] and claimed that the only existing efficient aggregate signature scheme is of Gentry and Ramzan [8] which involves interaction between all the signers whose signatures are to be aggregated. The design of an efficient identity based aggregate signature scheme without any interaction between the signers was left open by Hwang et al. [11]

Our contribution : In this paper, we propose two aggregate signature schemes. Our first scheme addresses the open problem posed by Hwang et al. [11]. We develop a scheme which does not require any pairing operation during aggregate signature verification. Also, we eliminate the interaction among the signers before signature generation which reduces the communication complexity to a large extent. We also achieve efficiency in computation (i.e we eliminate the costly operation, pairing). However in this scheme we are able to achieve only partial aggregation and not full aggregation. We use the identity based signature construct of Galindo et al. [7]. It is a light weight schnorr based signature construct which can be used in practice. We formally prove the security of our first scheme in the random oracle model. Our second scheme achieves efficiency in communication. This scheme achieves full aggregation with no interaction among the signers before the generation of aggregate signature, but the scheme requires linear number of pairing operations for aggregate signature verification. In the literature, there is no aggregate signature scheme that achieves full aggregation without any interaction among the users. Thus, this scheme is the first in literature to achieve this property. We formally prove the security of the scheme in the random oracle model.

2 Preliminary

2.1 Bilinear Pairing

Let $\mathbb{G}$ be an additive cyclic group generated by P, with prime order q, and $\mathbb{G}_T$ be a multiplicative cyclic group of the same order q. Let $\hat{e}$ be a pairing defined as $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. It satisfies the following properties. For any P, Q, $R \epsilon \mathbb{G}$ and a, $b \epsilon \mathbb{Z}_q^*$

- **Bilinearity :** $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
- **Non Degenerate :** $\hat{e}(P, P) \neq 1$.
- **Easily Computable :** $\hat{e}(P, Q)$ must be easily and efficiently computable.

2.2 Computational Assumptions

In this section, we review the computational assumptions related to bilinear maps that are relevant to the protocol we discuss.

Bilinear Diffie-Hellman Problem (BDHP): Given $(P, aP, bP, cP) \epsilon \mathbb{G}^4$ for unknown $a, b, c \epsilon \mathbb{Z}_q^*$, the BDH problem in $\mathbb{G}$ is to compute $\hat{e}(P, P)^{abc}$. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the BDH problem in $\mathbb{G}$ is defined as

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

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

Computation Diffie-Hellman Problem (CDHP): Given $(P, aP, bP) \epsilon \mathbb{G}^3$ for unknown $a, b \epsilon \mathbb{Z}_q^*$, the $CDHP$ problem in $\mathbb{G}$ is to compute abP. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{G}$ is defined as

$$Adv_{\mathcal{A}}^{CDH} = Pr[\mathcal{A}(P, aP, bP) = abP | a, b \epsilon \mathbb{Z}_q^*]$$

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

Discrete Logarithm Problem (DLP): Let $(\mathbb{G}, *)$ be a multiplicative group of order p, $g \in \mathbb{G}$ be a generator of $\mathbb{G}$ and $h = g^x \in \mathbb{G}$, where $x \in \mathbb{Z}_p$ be unknown. Given g and h, the discrete logarithm problem is to find x.

An algorithm $\mathcal{A}$ has an advantage ϵ in solving DLP_{G_1} if

$$Pr[\mathcal{A}(g, h) = x] \geq \epsilon.$$

3 Generic Model

An identity based aggregate signature scheme (IBAS) consists of following six algorithms.

- **Setup :** The private key generator (PKG) provides the security parameter κ as the input to this algorithm, generates the system parameters $params$ and the master private key Msk. PKG publishes $params$ and keeps Msk secret.
- **KeyGen :** The user U_i provides his identity ID_i to PKG. The PKG runs this algorithm with identity ID_i, $params$ and Msk as the input and obtains the private key D_i. The private key D_i is sent to user U_i through a secure channel.
- **Sign :** For generating a signature on a message m_i, the user U_i provides his identity ID_i, his private key D_i, $params$ and message m_i as input. This algorithm generates a valid signature σ_i on message m_i by user U_i.
- **Verify :** This algorithm on input of a signature σ on message m by user U with identity ID checks whether σ is a valid signature on message m by ID. If true it outputs "$Valid$", else it outputs "$Invalid$".
- **Aggregate :** On receiving the various signatures $(\sigma_i)_{i=1\,to\,n}$ from different users $(U_i)_{i=1\,to\,n}$, any third party or one of the signers can run this algorithm and generate the aggregate signature σ_{agg} for the set of $< message, identity >$ pairs $(m_i, ID_i)_{i=1\,to\,n}$.
- **AggregateVerify :** This algorithm on input of an aggregate signature σ_{agg}, the list for $(m_i, ID_i)_{i=1\,to\,n}$ and the $params$ checks whether σ_{agg} is a valid aggregate signature on m_i by ID_i for all $i = 1$ to n. If true, it outputs "$Valid$", else outputs "$Invalid$".
- **Token Generation :** In some models of identity based systems, the PKG may generate a random value that corresponds to the registered user at the time of registration / key generation. This value will be made public by the user. We refer this value as *'token'* and tokens are not used for any encryption schemes. These tokens will always be send as a part of the signature. This extended version of identity based signatures is not considered as violation because this toke is used only for signing purpose. More on this have been discussed in section 5.

Important Remark: As encryption algorithms only use the publicly known identities of the user alone as public key, tokens are never used in encryption schemes. Since ours is an identity based signature scheme, introduction of token in our cryptosystem is not a violation of the definition of identity based system.

4 Security Model

4.1 Unforgeability

Gentry et al. in [8] proposed a formal model for aggregate signature scheme. Their scheme used a common randomness. We follow the security model proposed by Gentry et al. with slight variations since we do not have a common random value. An IBAS scheme is secure against existential forgery under adaptive-chosen-identity and adaptive-chosen-message attack if no probabilistic polynomial time algorithm $\mathcal{A}$ has non-negligible advantage in the following game.

– **Setup phase :** The challenger $\mathcal{C}$ runs the setup algorithm and generates the $params$ and Msk. Challenger $\mathcal{C}$ gives $params$ to adversary $\mathcal{A}$.
– **Training phase :** After the setup, $\mathcal{A}$ starts interacting with $\mathcal{C}$ by querying the various oracles provided by $\mathcal{C}$ in the following way:
  • **KeyGen oracle :** When $\mathcal{A}$ makes a query with ID_i, $\mathcal{C}$ outputs D_i, the private key of ID_i to $\mathcal{A}$, provided $\mathcal{C}$ knows the private key for the queried identity. Else it aborts.
  • **Signing oracle :** When $\mathcal{A}$ makes a signing query with ID_i, message m_i, $\mathcal{C}$ outputs a valid signature σ_i on m_i by ID_i.
– **Forgery phase :** $\mathcal{A}$ outputs an aggregate signature σ_{Agg} for signatures $(\sigma)_{i=1\,to\,n}$ from the users $(ID_i)_{i=1\,to\,n}$ on messages $(m_i)_{i=1\,to\,n}$ where there exists at least one target identity $ID_T \in \{ID_i\}_{i=1\,to\,n}$, for which private key has not been queried for. The adversary $\mathcal{A}$ wins the game if σ_{agg} is a valid aggregate signature and $\mathcal{A}$ has not queried for the signature from the signing oracle for (ID_T, m_T) pair on which it has generated the forgery.

$$Adv_{\mathcal{A}}^{IBAS} = \{Pr[\mathcal{A}(Verify(\sigma_{agg}) = valid)\}$$

5 Identity Based Aggregate Signature scheme Without Pairings-IBAS-1

Normally, the public key of a user in identity based cryptography is obtained by hashing the user's identity, which uniquely identifies him. In the identity based signature by Galindo et al. [7], we find an interesting and subtle difference between all existing schemes and [7]. In [7], Galindo et al. have used a Schnorr signature which in turn uses a purely random value chosen by the PKG to generate the private key of the user. This random value can be interpreted as a *'token'* which we discussed in section 3 on generic model of identity based aggregate signature scheme. This token along with the identity of the user is hashed together to obtain the public key corresponding the user. It should be noted that this is not a violation of the property of identity based cryptosystem with respect to digital signature schemes because in a digital signature scheme all the components of a signature on an arbitrary message are generated by the signer who is in possession of the private key. Hence, the signer has to send the random value obtained with his private key from the PKG along with each signature he generates. The interesting part is that, if the signer or any potential forger tries to alter the random value obtained from the PKG for the signer, both will fail miserably in generating a valid signature because neither signer nor the forger will be able to generate a valid private key corresponding to the altered random value. We emphasize again that tokens can never be used for encryption schemes and can always be used in signature schemes. In Galindo et al.'s [7] paper, the component g^r is send by the PKG to the user. This component is called as *'token'* in our convention.

Similar kind of key constructs for identity based cryptosystem can be seen in [6] and [12]. In [6], an identity based key agreement protocol was proposed by Dario et al. and in [12] an identity based online/offline signature was proposed. In this section, we describe a new identity based aggregate signature scheme based on the identity based signature scheme by Galindo et al. [7]. This scheme consists of six algorithms which are described below.

– **IBAS-1.Setup :** Let κ be the security parameter of the system. Let $\mathbb{G}$ be a multiplicative group of order q. Choose a random generator g of $\mathbb{G}$. Choose three cryptographic hash functions which are defined as $H_1 : \{0,1\}^* \times \mathbb{G} \longrightarrow \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{Z}_q^*$ and $H_3 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{Z}_q^* \times \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{Z}_q^*$. Let $s \in_R \mathbb{Z}_q^*$ be the master private key and the master public key is set to be g^s. The public parameters are $params = < g, g^s, \mathbb{G}, H_1, H_2, H_3 >$ and the master private key s is kept secret.

– **IBAS-1.KeyGen :** The user U_i provides his identity ID_i to the Private Key Generator(PKG). The PKG runs this algorithm with ID_i, $params$ and master private key s as the input. The algorithm does the following:
  • Choose a random $x_i \in \mathbb{Z}_q$
  • Computes $X_i = g^{x_i}$ and $q_i = H_1(ID_i, X_i) mod(q)$
  • Computes $d_i = (x_i + sq_i) mod(q)$
  • Outputs $< q_i, X_i, d_i >$

The PKG sends $< q_i, X_i, d_i >$ securely to the user U_i. The user U_i keeps the d_i as secret and $< q_i, X_i >$ as public. Here X_i is called the token.

Remark: It is to be noted that the private key d_i is a Schnorr signature on the identity ID_i and thus a user who is capable of producing another private key d_i' for the same identity ID_i or a private key d_i'' for an arbitrary identity ID_i'' can effectively forge the underlying Schnorr signature. As a consequence, the private key generated by the PKG is secure and cannot be generated by any user by altering the token value, unless he knows the master private key s. Therefore, we do not consider token as a separate entity for the formal proof of unforgeability of our schemes.

- **IBAS-1.Sign :** The user U_i who wishes to sign a message m_i gives his ID_i, private key d_i and *params* as input to this algorithm. The algorithm does the following to generate the signature:
  - Chooses a random $r_i \in \mathbb{Z}_q$.
  - Computes $W_i = g^{r_i}$
  - Generates $h_{1i} = H_2(m_i, ID_i, W_i, X_i)$
  - Generates $h_{2i} = H_3(m_i, ID_i, h_{1i}, W_i, X_i)$
  - Computes $V_i = (r_i h_{1i} + h_{2i} d_i) mod(q)$
  - Outputs $\langle V_i, W_i \rangle$ as the signature of ID_i on message m_i.
- **IBAS-1.Verify :** Any user can run this verification algorithm. The user provides $< V_i, W_i >$, ID_i, m_i and *params* as input to this algorithm. The verification is done as follows:
  - Check whether $g^{V_i} \stackrel{?}{=} (W_i)^{h_{1i}} (X_i)^{h_{2i}} (g^s)^{q_i h_{2i}}$, where
    $h_{1i} = H_2(m_i, ID_i, W_i, X_i)$.
    $h_{2i} = H_3(m_i, ID_i, h_{1i}, W_i, X_i)$.
  - Outputs "$Valid$" if the signature passes the verification, else it outputs "$Invalid$".

  Correctness

$$\begin{aligned} g^{V_i} &= g^{r_i h_{1i} + h_{2i} d_i} \\ &= g^{r_i h_{1i}} . g^{h_{2i} d_i} \\ &= (g^{r_i})^{h_{1i}} . g^{h_{2i}(x_i + s q_i)} \\ &= (W_i)^{h_{1i}} . (X_i)^{h_{2i}} . (g^s)^{q_i h_{2i}} \end{aligned}$$

  This shows that the above verification check is valid and consistent. Note that the verification can be done by anyone as it involves only publicly known parameters such as $V_i, W_i, h_{1i}, h_{2i}, g^s, X_i, q_i$
- **IBAS-1.Aggregate :** This algorithm takes as input a set of n signatures $\{V_i, W_i\}_{i=1\,to\,n}$ and the corresponding identity, message pairs $< ID_i, m_i >$, such that $\forall i = 1$ to $n < V_i, W_i >$ is the signature on message m_i by ID_i. The aggregation is done as follows:

$$V_{agg} = \sum_{i=1}^{n} V_i.$$

  The algorithm outputs the final aggregate signature $< V_{agg}, W_1, W_2, ..., W_n >$ and the corresponding message identity pair $\{m_i, < ID_i, X_i >\}_{i=1\,to\,n}$.
- **IBAS-1.AggregateVerify :** This algorithm takes the aggregate signature $< V_{agg}, W_1, W_2, ..., W_n >$ and the corresponding message identity pair $\{m_i, ID_i, X_i\}_{i=1\,to\,n}$ as does the following:
  - For all i=1 to n
    Compute $h_{1i} = H_2(m_i, ID_i, W_i, X_i)$
    Compute $h_{2i} = H_3(m_i, ID_i, h_{1i}, W_i, X_i)$
  - If $(g^{V_{agg}} = \prod_{i=1}^{n} (W_i)^{h_{1i}} . \prod_{i=1}^{n} (X_i)^{h_{2i}} . (g^s)^{\sum_{i=1}^{n} q_i h_{2i}})$ then outputs "$Valid$" else outputs "$Invalid$".

  Correctness

$$\begin{aligned} g^{V_{agg}} &= g^{\sum_{i=1}^{n} r_i h_{1i} + \sum_{i=1}^{n} h_{2i} d_i} \\ &= g^{\sum_{i=1}^{n} r_i h_{1i}} . g^{\sum_{i=1}^{n} h_{2i} d_i} \\ &= \prod_{i=1}^{n} (g^{r_i})^{h_{1i}} . g^{\sum_{i=1}^{n} h_{2i}(x_i + s q_i)} \\ &= \prod_{i=1}^{n} (W_i)^{h_{1i}} . \prod_{i=1}^{n} (X_i)^{h_{2i}} . (g^s)^{\sum_{i=1}^{n} q_i h_{2i}} \end{aligned}$$

  This shows that the aggregate verification test is correct and consistent.

6 Security Proof for IBAS-1

In this section, we prove the security of our identity based aggregate signature scheme(IBAS-1). We show that if a polynomial time bounded adversary exists who can break our scheme with non-negligible probability $\epsilon^{'}$ then we will be able to solve the discrete logarithm problem with non-negligible probability ϵ_0. We prove that our scheme is secure against existential forgery under adaptive chosen message and adaptive chosen identity attack. We also use the oracle replay attack technique and forking lemma [13] to prove the security of our scheme.

Theorem 1. *Our aggregate signature scheme IBAS-1 is secure against existential forgery under adaptively chosen identity and adaptively chosen message attack, if there exists a polynomially bounded $(t, \epsilon^{'})$ adversary $\mathcal{A}$ making q_{H_1}, q_{H_2}, q_{H_3} hash queries, q_S signcryption queries and q_E extraction queries, who can break our scheme with a non-negligible advantage $\epsilon^{'}$, then there exists a DL solver $\mathcal{C}$ with a non-negligible advantage, $\epsilon_0 = \frac{1}{9}.\frac{10(q_S+1)(q_S+q_{H_3}+q_{H_2}).(1-\frac{q_E}{q_{H_1}}).n}{2^{k+1}}.\frac{1}{q_{H_1}}\epsilon$ and in polynomial time t_0.*

The proof of this theorem will be added later.

7 Identity Based Aggregate Signature Scheme-IBAS-2

For any identity based signature the secret key and the signature belongs to the same group(mostly elliptic curve groups) and the verification of the signature will be carried out in a different group(usually a multiplicative group). In IBAS-2 we use a different strategy compared to this traditional approach in identity based signature schemes. IBAS-2 has the private key, signature and verification in three different groups namely $(\mathbb{Z}_q^*, \mathbb{G}_1, \mathbb{G}_2)$ In this section we propose another efficient identity based aggregate signature scheme which achieves full aggregation thus reducing the communication complexity considerably. Currently there is no aggregate signature scheme which achieves full aggregation without any interaction among the users. We achieve full aggregation without any kind of interaction among the signers. This scheme consists of six algorithms which are defined as follows:

- **IBAS-2.Setup :** Let κ be the security parameter of the system. The Private Key Generator(PKG) runs this algorithm with κ as input. This algorithm chooses an additive group $\mathbb{G}_1$ and a multiplicative group $\mathbb{G}_2$, both of order q. It randomly selects P which is generator of group $\mathbb{G}_1$, randomly selects $s_1, s_2 \in_R \mathbb{Z}_q^*$ and sets $P_{pub_1} = s_1P$ and $P_{pub_2} = s_2P$. It chooses two cryptographic hash functions which are defined as $H_1 : \{0,1\}^* \times \mathbb{G}_1 \longrightarrow \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^l \times \mathbb{G}_1 \times \{0,1\}^* \longrightarrow \mathbb{G}_1$. It also chooses a bilinear pairing $\hat{e}$ which is defined as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. Finally it outputs the public parameters *params* as $< P, P_{pub_1}, P_{pub_2}, \hat{e}, \mathbb{G}_1, \mathbb{G}_2, H_1, H_2 >$ and the master private keys s_1, s_2 is kept secret.

- **IBAS-2.KeyGen :** The user U_i submits his identity ID_i to the PKG. The PKG runs this algorithm with ID_i, s_1 and *params* as input. This algorithm does the following:
  - Chooses a random $x_i \in Z_q^*$
  - Computes $X_i = x_iP$ and $q_i = H_1(ID_i, X_i)$
  - Computes $d_i = x_i + s_1q_i \bmod(q)$
  - Outputs $< X_i, q_i, d_i >$

  The PKG returns $< q_i, X_i, d_i >$ securely to the user U_i. The user U_i keeps d_i as secret and makes $< q_i, X_i >$ public.

- **IBAS-2.Sign :** The user U_i who wishes to sign the message m_i gives his identity ID_i, private key d_i, public parameters *params* and message m_i as input to this algorithm. The computations performed are the following:

  - Chooses a random $r_i \in Z_q^*$
  - Computes $W_i = r_iP$
  - Generates $H_i = H_2(m_i, X_i, ID_i)$
  - Computes $V_i = r_iP_{pub_2} + d_iH_i$

- Outputs the signature $\sigma_i =< W_i, V_i >$ on message m_i by user with identity ID_i.

– **IBAS-2.Verify :** Any user can run the verification algorithm. This algorithm takes as inputs the signature $\sigma_i =< W_i, V_i >$ and the message, identity pair $< m_i, ID_i >$. The verification is done as follows:

- Check $\hat{e}(V_i, P) \stackrel{?}{=} \hat{e}(W_i, P_{pub_2})\hat{e}(q_i P_{pub_1} + X_i, H_i)$
  where $H_i = H_2(m_i, X_i, ID_i)$
- If the signature passes the verification test it outputs "$Valid$" else it outputs "$Invalid$".

Correctness :

$$\begin{aligned}\hat{e}(V_i, P) &= \hat{e}(r_i P_{pub_2} + d_i H_i, P) \\ &= \hat{e}(r_i P, P_{pub_2})\hat{e}((x_i + s_1 q_i)H_i, P) \\ &= \hat{e}(r_i P, P_{pub_2})\hat{e}(H_i, x_i P)\hat{e}(H_i, q_i s_1 P) \\ &= \hat{e}(r_i P, P_{pub_2})\hat{e}(H_i, X_i)\hat{e}(H_i, q_i P_{pub_1}) \\ &= \hat{e}(W_i, P_{pub_2})\hat{e}(H_i, X_i + q_i P_{pub_1})\end{aligned}$$

This shows that the above verification is valid and consistent. Note that the verification can be done by anyone as it involves only publicly known parameters such as $V_i, W_i, h_{1i}, h_{2i}, g^s, X_i$.

– **IBAS-2.Aggregate :** This algorithm takes in as input a set of n signatures $\{V_i, W_i\}_{i=1\,to\,n}$ and the corresponding identity message pair ID_i, m_i such that $\forall i = 1$ to $n < V_i, W_i >$ is the signature on message m_i by ID_i. The aggregation is done as follows:

$$V_{agg} = \sum_{i=1}^{n} V_i$$
$$W_{agg} = \sum_{i=1}^{n} W_i$$

The algorithm outputs the aggregated signature $< V_{agg}, W_{agg} >$ and the list of message, identity pairs $\{m_i, ID_i\}_{i=1\,to\,n}$.

– **IBAS-2.AggregateVerify :** Any user can run this aggregate verify algorithm. This algorithm takes as input the aggregate signature $< V_{agg}, W_{agg} >$, $params$ and the list of message,identity pairs $\{m_i, ID_i\}_{i=1\,to\,n}$. This checks whether the following relation holds.

- For all $i = 1$ to n Computes $H_i = H_2(m_i, X_i, ID_i)$
- Check if $\hat{e}(V_{agg}, P) \stackrel{?}{=} \hat{e}(W_{agg}, P_{pub_2}) \prod_{i=1}^{n} \hat{e}(q_i P_{pub_1} + X_i, H_i)$
- If the aggregate signature passes the verification it outputs "$Valid$" else the algorithm outputs "$Invalid$".

Correctness :

$$\begin{aligned}\hat{e}(V_{agg}, P) &= \hat{e}(\textstyle\sum_{i=1}^{n} r_i P_{pub_2} + d_i H_i, P) \\ &= \hat{e}(\textstyle\sum_{i=1}^{n} r_i P, P_{pub_2})\hat{e}(\sum_{i=1}^{n}(x_i + s_1 q_i)H_i, P) \\ &= \hat{e}(\textstyle\sum_{i=1}^{n} W_i, P_{pub_2}) \prod_{i=1}^{n} \hat{e}(H_i, x_i P) \prod_{i=1}^{n} \hat{e}(H_i, q_i s_1 P) \\ &= \hat{e}(W_{agg}, P_{pub_2}) \prod_{i=1}^{n} \hat{e}(H_i, X_i) \prod_{i=1}^{n} \hat{e}(H_i, q_i P_{pub_1}) \\ &= \hat{e}(W_{agg}, P_{pub_2}) \prod_{i=1}^{n} \hat{e}(H_i, X_i + q_i P_{pub_1})\end{aligned}$$

This shows that the above verification is valid and consistent.

8 Security Proof for IBAS-2

In this section we formally prove the security of our scheme. We prove that if there is a polynomial time adversary $\mathcal{A}$ exists with non-negligible advantage ϵ to break our scheme then we will be able to solve an instance of CDH problem with non-negligible probability.

8.1 Unforgeability

Theorem 2. *Our aggregate signature scheme IBAS-2 is secure against existential forgery under adaptively chosen identity and adaptively chosen message attack, if there exists a polynomialy bounded (t, ϵ) adversary $\mathcal{A}$ making q_{H_1}, q_{H_2} hash queries, q_S signcryption queries and q_E extraction queries, who can break our scheme with a non-negligible advantage ϵ, then there exists a CDH solver $\mathcal{C}$ with non-negligible advantage $\epsilon_0 = (1-\mu)^{q_E}(1-\mu^n)\epsilon$ and in polynomial time t_0.*

The proof of this theorem will be added later.

9 Efficiency Comparison :

In this section we compare the efficiency of our schemes with few existing schemes. We also give some remarks on the efficiency and merits of our schemes over others.

Table 1. Comparing various aggregate signature schemes

Scheme	Signing for each signer		Verification		
	Exp	Pt.Mul	Exp	Pt.Mul	Pairing
Gentry et al. [8]	-	3	-	n	3
Jung Cheon et al. [19]	-	3	-	n	n+1
Jing Xu et al. [18]	-	2	-	-	n+2
Javier Herranz [9]	1	-	n	-	n
Cheng et al. [4]	-	3	-	n	2
Our scheme IBAS-1	1	-	2n+2	-	-
Our scheme IBAS-2	-	3	-	n	n+2

Remarks :

- In Gentry Ramzan scheme [8] all the signers have to agree upon a common value ω in order to produce a valid aggregate signature. That will increase the communication complexity. Further weakness of Gentry et al.'s scheme is explained in the appendix.
- Jung et al. [19],Jing Xu et al. [18] achieve only partial aggregation and also requires linear number of pairings.
- Javier Herranz [9] also achieves only partial aggregation and also has the disadvantage that the underlying signature scheme is deterministic. The signature on a message by a user always remains the same.
- In Cheng et al. [4] scheme achieves full aggregation and also seems efficient. But in this scheme all the signers have to broadcast their respective randomness to other signers so that all agree upon a common randomness finally. This broadcast technique increases the communication complexity enormously and also it is rather like threshold signature and not like a pure aggregate signature.
- It has to be taken into account that the signing part is for each signer. So if n signers are signing the complexity in signing part will be multiplied by n.
- Our first scheme IBAS-1 achieves only partial aggregation but does verification without any pairing operation making it the most efficient scheme of all the above. We used a light weight schnorr based signature as proposed by Galindo et al. [7] which is highly efficient and practically implementable.
- Our second scheme IBAS-2, though it has linear number of pairings, achieves full aggregation without any kind of interaction among the signers something which has not been achieved by any of the existing schemes. Our scheme has a trivial weakness similar to that of Gentry et al.'s scheme which will be discussed as the fourth weakness in appendix. To achieve a full aggregation scheme without any interactions and without the specified weakness seems to be a really interesting open problem.

10 Conclusion

In this paper, we have considered an identity based signature in which the private key for a user is a Schnorr signature on his identity. This private key is generated by the PKG. Besides, the PKG sends a random *'token'* to every user along with his private key. This token cannot be altered by the user and the token can never be used in any encryption scheme. Since, for encryption schemes, only identities are used as public keys. The presence of tokens in the scheme is not a violation to the definition of identity based scheme. However, the concept of *'token'* can be cleverly deployed to avoid all pairing based computations in aggregate signature schemes. We have demonstrated that Galindo et al's [7] signature scheme which uses the concept of *'tokens'* can be used to design an aggregate signature scheme without pairing.

We have proposed two identity based aggregate signature schemes IBAS-1 and IBAS-2. IBAS-1 and IBAS-2 uses schnorr signature based private key construct. IBAS-1 employs a variant of schnorr signature, with no pairing operation, which is the first aggregate scheme without pairing and achieves partial aggregation. IBAS-2 is the only identity based aggregate signature scheme which achieves full aggregation. Both IBAS-1 and IBAS-2 eliminates the need for interaction among the signers which is a overhead in existing efficient aggregate signature schemes. We have formally proved the security of both the schemes in the random oracle model. We have also addressed the open problem posed by Hwang et al. in [11]. Presently, there seems to be no scheme which achieves efficiency in both communication and computation front without any kind of interaction among the signers. Achieving this, seems to be an important open problem considering its practical significance.

References

1. Alexandra Boldyreva, Craig Gentry, Adam O'Neill, and Dae Hyun Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 276–285. ACM, 2007.
2. Dan Boneh. Bls short digital signatures. In Henk C. A. van Tilborg, editor, *Encyclopedia of Cryptography and Security*. Springer, 2005.
3. Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap diffie-hellman groups. In Yvo Desmedt, editor, *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30. Springer, 2003.
4. Xiangguo Cheng, Jingmei Liu, and Xinmei Wang. Identity-based aggregate and verifiably encrypted signatures from bilinear pairing. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *ICCSA (4)*, volume 3483 of *Lecture Notes in Computer Science*, pages 1046–1054. Springer, 2005.
5. Shi Cui, Pu Duan, and Choong Wah Chan. An efficient identity-based signature scheme with batch verifications. In Xiaohua Jia, editor, *Infoscale*, volume 152 of *ACM International Conference Proceeding Series*, page 22. ACM, 2006.
6. Dario Fiore and Rosario Gennaro. Making the diffie-hellman protocol identity-based. Cryptology ePrint Archive, Report 2009/174, 2009. http://eprint.iacr.org/ (An extended abstract of this paper appears in the proceedings of CT-RSA 2010).
7. David Galindo and F. D. Garcia. A schnorr-like lightweight identity-based signature scheme. In *In Proceedings of 2nd African International Conference on Cryptology, AfricaCrypt 2009*, Lecture Notes in Computer Science 5580, pages 135–148, 2009.
8. Craig Gentry and Zulfikar Ramzan. Identity-based aggregate signatures. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 257–273. Springer, 2006.
9. Javier Herranz. Deterministic identity-based signatures for partial aggregation. *Comput. J.*, 49(3):322–330, 2006.
10. Florian Hess. Efficient identity based signature schemes based on pairings. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 310–324. Springer, 2002.
11. Jung Yeon Hwang, Dong Hoon Lee, and Moti Yung. Universal forgery of the identity-based sequential aggregate signature scheme. In Wanqing Li, Willy Susilo, Udaya Kiran Tupakula, Reihaneh Safavi-Naini, and Vijay Varadharajan, editors, *ASIACCS*, pages 157–160. ACM, 2009.
12. Joseph K. Liu, Joonsang Baek, Jianying Zhou, Yanjiang Yang, and Jun Wen Wong. Efficient online/offline identity-based signature for wireless sensor network. Cryptology ePrint Archive, Report 2010/003, 2010. http://eprint.iacr.org/.

13. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
14. 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/.
15. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
16. Zhu Wang, Huiyan Chen, Ding feng Ye, and Qian Wu. Practical identity-based aggregate signature scheme from bilinear maps. volume 13(6), pages 684–687. Shangai Jiao Tong University Press, 2008.
17. Yiling Wen and Jianfeng Ma. An aggregate signature scheme with constant pairing operations. In *CSSE (3)*, pages 830–833. IEEE Computer Society, 2008.
18. Jing Xu, Zhenfeng Zhang, and Dengguo Feng. Id-based aggregate signatures from bilinear pairings. In Yvo Desmedt, Huaxiong Wang, Yi Mu, and Yongqing Li, editors, *CANS*, volume 3810 of *Lecture Notes in Computer Science*, pages 110–119. Springer, 2005.
19. HyoJin Yoon, Jung Hee Cheon, and Yongdae Kim. Batch verifications with id-based signatures. In Choonsik Park and Seongtaek Chee, editors, *ICISC*, volume 3506 of *Lecture Notes in Computer Science*, pages 233–248. Springer, 2004.
20. Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. An efficient signature scheme from bilinear pairings and its applications. In Feng Bao, Robert H. Deng, and Jianying Zhou, editors, *Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 277–290. Springer, 2004.

11 *APPENDIX :*

In this section we show the various weakness in the Gentry et al.'s scheme [8]. Though it is claimed to be currently the most efficient scheme it has the following weakness.

1. According to the scheme in [8], the signers have to store all the ω they have used previously in a database in order to avoid the re-use of randomness. Every time the signer before signing needs to check whether the current ω was not previously used for any signature generation on any message. This leads to increase in storage cost and checking cost, becoming a huge overhead.
2. Not only this, the common randomness ω chosen by the first signer should satisfy the constraint that it was not used previously for signing any message by any o the other signers participating in the aggregation process. Even if $n-1$ signers agree and n^{th} signer disagrees then they have to run the protocol again by picking up a new ω value. This accounts to a lot of wastage of time and network bandwidth.
3. Any signer, if he/she reuse a ω value even once with or without his/her knowledge then universal forgery of their signature is possible. The universal forgery of signature is as follows:
   - Let $< S_1, T_1 >$ be a signature on m_1 by ID using the value ω.
   - Let $< S_2, T_2 >$ be a signature on m_2 by ID using the same ω
   - The signature components is of the form

$$S_1 = r_1 P_\omega + D_1 + c_1 D_2 \tag{1}$$
$$T_1 = r_1 P \tag{2}$$
$$S_2 = r_2 P_\omega + D_1 + c_2 D_2 \tag{3}$$
$$T_2 = r_2 P \tag{4}$$

     where r_1, r_2 is unknown random numbers, $P_\omega = H_2(\omega)$, $c_1 = H_3(m_1, ID, \omega)$, $c_2 = H_3(m_2, ID, \omega)$.

   - Subtracting 1 from 3 we get

$$S^* = (r_2 - r_1) P_\omega + (c_2 - c_1) D_2 \tag{5}$$

     Dividing by $(c_2 - c_1)$ we get, $S^{**} = \frac{r_2 - r_1}{c_2 - c_1} P_\omega + D_2$

   - Compute a new hash value $c_3 = H_3(m_3, ID, \omega)$ where m_3 is some random message.

   - Multiply S^{**} by c_3 and we get $S^{'} = \frac{r_2 - r_1}{c_2 - c_1} c_3 P_\omega + c_3 D_2$

– Dividing 1 by c_1 and dividing 3 by c_2 we get the two equations

$$S_1^{'} = \frac{r_1}{c_1}P_\omega + \frac{1}{c_1}D_1 + D_2 \tag{6}$$

$$S_2^{'} = \frac{r_2}{c_2}P_\omega + \frac{1}{c_2}D_1 + D_2 \tag{7}$$

– Subtracting 7 from 6 we get
$S_3^{'} = (\frac{r_1}{c_1} - \frac{r_2}{c_2})P_\omega + (\frac{1}{c_1} - \frac{1}{c_2})D_1$

– Dividing $S_3{'}$ by $(\frac{1}{c_1} - \frac{1}{c_2})$ we get
$S_3^{''} = \frac{(\frac{r_1}{c_1} - \frac{r_2}{c_2})}{(\frac{1}{c_1} - \frac{1}{c_2})}P_\omega + D_1$

$S_3^{''} = \frac{r_1c_2 - r_2c_1}{c_2 - c_1}P_\omega + D_1$

– Adding $S^{''}$ to $S^{'}$ we get $S_3 = (\frac{r_2 - r_1}{c_2 - c_1}c_3 + \frac{r_1c_2 - r_2c_1}{c_2 - c_1})P_\omega + D_1 + c_3D_2$

– $S_3 = r^*P_\omega + D_1 + c_3D_2$ where $r^* = (\frac{r_2 - r_1}{c_2 - c_1}c_3 + \frac{r_1c_2 - r_2c_1}{c_2 - c_1})$

– We can derive $T_3 = r^*P$ also knowing the T_1 and T_2 values as follows:

Multiplying T_1 by $(\frac{c_2 - c_3}{c_2 - c_1})$ and multiplying T_2 by $(\frac{c_3 - c_1}{c_2 - c_1})$ and adding the two we get
$T_3 = (\frac{c_2 - c_3}{c_2 - c_1}r_1 + \frac{c_3 - c_1}{c_2 - c_1}r_2)P$

$= (\frac{r_2 - r_1}{c_2 - c_1}c_3 + \frac{r_1c_2 - r_2c_1}{c_2 - c_1})P$

$T_3 = r^*P$

– Thus S_3 and T_3 is a valid signature on message m_3 since its of the standard signature format of Gentry et al.'s scheme where $c_3 = H_3(m_3, ID, \omega)$.

Thus universal forgery of signature is possible in case of Gentry et al.'s scheme. The the signer has to be very careful that he/she does not reuse the ω value. So the storage and checking of all used ω values becomes essential in their scheme. The extension which the authors have stated for using ω repeatedly is to get different private keys equal to number of times ω is reused. That again is not a viable solution since the user will have no idea as to how many times he will reuse ω if he does.

4. Using a single signature on a message one can generate a different signature which is valid on the same message by the same user. Though this is not a flaw in the scheme it is considered as a weakness in certain scenarios (strong unforgeability is not satisfied).