OPEN ACCESS Freely available online

PLOS ONE

Efficient Unrestricted Identity-Based Aggregate Signature Scheme

Yumin Yuan[1]*, Qian Zhan[2], Hua Huang[3]

1 School of Applied Mathematics, Xiamen University of Technology, Xiamen, China, **2** University of Science and Technology Beijing, Beijing, China, **3** University of Xiamen, Xiamen, China

Abstract

An aggregate signature scheme allows anyone to compress multiple individual signatures from various users into a single compact signature. The main objective of such a scheme is to reduce the costs on storage, communication and computation. However, among existing aggregate signature schemes in the identity-based setting, some of them fail to achieve constant-length aggregate signature or require a large amount of pairing operations which grows linearly with the number of signers, while others have some limitations on the aggregated signatures. The main challenge in building efficient aggregate signature scheme is to compress signatures into a compact, constant-length signature without any restriction. To address the above drawbacks, by using the bilinear pairings, we propose an efficient unrestricted identity-based aggregate signature. Our scheme achieves both full aggregation and constant pairing computation. We prove that our scheme has existential unforgeability under the computational Diffie-Hellman assumption.

Citation: Yuan Y, Zhan Q, Huang H (2014) Efficient Unrestricted Identity-Based Aggregate Signature Scheme. PLoS ONE 9(10): e110100. doi:10.1371/journal.pone.0110100

Editor: Francesco Pappalardo, University of Catania, Italy

Received November 11, 2013; **Accepted** September 16, 2014; **Published** October 20, 2014

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* Email: yuanymp@163.com

Introduction

An aggregate signature [1] is a useful primitive that allows anyone to compress n individual signatures, say $\sigma_1, \ldots, \sigma_n$ where σ_i is a signature from user with identity ID_i on message m_i for $1 \le i \le n$, into a single (shorter) signature even if these signatures are on the same message or are produced by the same signer. The main goal in the design of such protocols is to reduce the costs on storage, communication and computation. Informally, the length of the aggregate signature should be constant, independent of the number of messages and signers. The resulting signature can convince a verifier that the user ID_i indeed signed the corresponding message m_i for all i:$1 \le i \le n$. This primitive is useful in many real-world applications (which involve multiple signatures on multiple messages generated by multiple users) especially in environments with low-band-width communication, low-storage and low computability. Typical applications for such schemes are Wireless sensor networks (WSNs) since WSNs are resource constraint: limited power supply, bandwidth for communication, memory space [2]. For example, in an environment monitoring network, the sensors record measurements from the environment, sign its data and send them to a monitoring center. The center aggregates these data and the signatures to save storage [3]. Aggregate signature scheme can also be applied to vehicular communications [4], many-to-one authentication [5], electronic transactions [6] and cloud computing [7] to enhance the efficiency of verification and reduce the communication over-head.

Boneh, Gentry, Lynn and Shacham [1] first defined of aggregate signature and presented a concrete aggregate signature which was constructed under traditional public key cryptography (PKC). In traditional PKC, a digital signature provides the authenticity of a signed message with respect to a public key, while the authenticity of the public key with respect to a signer is contained in a certificate provided by a certificate authority (CA). Whenever a verifier wants to verify a signature, he has first to verify the corresponding certificate. Therefore, aggregate signature working under traditional PKC requires heavy management, communication and computation cost to achieve authenticity of all signers' public keys, making the scheme both space and time inefficient, especially when the number of signers is large. To reduce this burden, Shamir [8] proposed the concept of identity-based public key cryptography (IB-PKC). The IB-PKC requires a trusted third party, typically called a "Private Key Generator" (PKG) which serves a similar role to the CA in a PKC system, to generate system parameters and user's private key. In an identity-based cryptosystem, only the PKG has a traditional public key, and the public key of each user is derived directly from his identity information, such as his email address. The direct derivation of users' public keys in these infrastructures eliminates the need for the certificate and some of the problem associated with them. In an identity-based signature (IBS) scheme, to generate valid signatures of a signer with the identity ID, one needs to know the private key of ID, while verifier can directly use the signer's identity ID and the PKG's public key to verify signatures. This advantage of identity-based aggregate signature (IBAS) becomes more compelling when we consider multiple signers. In this setting, when all signers have their secret keys issued by the same private key generator (PKG), the verifier needs only one traditional public key (of the PKG) to verify multiple identity-based signatures on multiple messages.

To shorten the length of signatures and to avoid the authentication of the public keys, Cheon et al. [9] presented the

first identity-based aggregate signature (IBAS) scheme. To date several IBAS schemes have been proposed [9–20]. However, some of them have additional restrictions conditions on aggregation step. The schemes [10,11] do not support simultaneous aggregation, which only allow each signer to aggregate his signature to a previously aggregated signature in turn. The scheme [12] requires that all signers participating in aggregation have to agree upon a common random string which was never used by any of the signers. Secure use of the scheme [12] is restricted to the aggregation of signatures from distinct signers. The scheme [13] requires interactive communication between signers to generate an aggregate signature, and hence increases the communication complexity.

Among existing unrestricted aggregate signature schemes (which enable any user to freely aggregate multiple signatures) in the identity-based setting [9,14–20], all but one of them [9,14–19] are able to achieve only partial aggregation and not full aggregation, i.e., the length of the resulting aggregate signature grows with the number of aggregated individual signatures, which departs from the main goals of aggregate signatures. Obviously, such schemes are impractical for some wireless network scenarios. Only the scheme in [20] achieves constant-length aggregate signature. But this scheme requires a large number of pairing operations in which the number of pairing operations in the aggregate signature verification algorithm is proportional to the number of aggregated individual signatures.

In this paper, we construct an efficient IBAS scheme without any restriction. The proposed protocol is based on bilinear pairings. The new scheme simultaneously achieves constant-length aggregate signature and constant pairing operations during signature verification, and is shown to be existentially unforgeable against adaptive chosen message attacks under the computational Diffie-Hellman assumption in the random oracle model.

Preliminaries

In this section, we review the basic concept of bilinear pairings and the complexity assumption on which our scheme relies.

2.1 Bilinear pairings

Let G_1 be a cyclic additive group of prime order q and G_2 be a cyclic multiplicative group of the same order. A map $e: G_1 \times G_1 \rightarrow G_2$ is called a bilinear pairing if it satisfies the following properties:

1. Bilinear: $e(aP,bQ)=e(P,Q)^{ab}$ for all $P,Q \in G_1$ and all $a,b \in Z_q$.
2. Non-degeneracy: There exist $P,Q \in G_1$ such that $e(P,Q) \neq 1$.
3. Computable: There is an efficient algorithm to compute $e(P,Q)$ for any $P,Q \in G_1$.

2.2 Related complexity assumption

Definition 1. A function : $N \rightarrow R$ is said to be negligible if, for every positive polynomial *poly* $(\cdot)$ there exists an integer $K > 0$ such that for all $k > K$ it holds.

$$(k) < \frac{1}{poly(k)}$$

Otherwise, we call non-negligible.

Definition 2. Let G be a group of prime order $q \geq 2^k$ where k is a security parameter. Computational Diffie-Hellman (CDH) Problem is that given three elements $P,aP,bP \in G$ for unknown randomly chosen $a,b \in Z_q$, compute abP.

Let $\mathcal{A}$ be a probabilistic polynomial-time algorithm. The advantage of $\mathcal{A}$ in solving the CDH problem in group G is defined to be.

$$Adv_{\mathcal{A}}^{CDH} = \Pr[\mathcal{A}(P,aP,bP) = abP]$$

where the probability is taken over the uniformly and independently chosen instance with a given security parameter k and over the random choices of $\mathcal{A}$.

The CDH assumption states that for every probabilistic polynomial-time algorithm $\mathcal{A}$, $Adv_{\mathcal{A}}^{CDH}$ is negligible.

Definitions and Security Models

We first review the definition and the formal security model for IBS schemes. Then we describe the definition and the formal security model for IBAS schemes.

3.1 Formal model of identity-based signature schemes

3.1.1 Definition of identity-based signature schemes. An identity-based signature (IBS) scheme is a tuple of probabilistic polynomial-time algorithms (**Setup**, **Extract**, **Sign**, **Verify**). The description of each algorithm is as follows.

- **Setup**. This algorithm is run by a private key generator (PKG). It takes a security parameter k as input and outputs a master key *msk* and a list of system parameters *params*. The system parameters will be publicly known while the master key will be known to the PKG only.
- **Extract**. This algorithm takes a user's identity ID_i, a system parameters *params* and a master key *msk* as input, and outputs the user's private key D_i. Usually, this algorithm is run by the PKG. The PKG sends D_i to the user ID_i through a secure channel.
- **Sign**. This algorithm takes a system parameters *params*, a message m_i, an identity ID_i and corresponding private key D_i as input, and outputs an individual signature σ_i on the message m_i for the user with identity ID_i. This algorithm is executed by the user ID_i.
- **Verify**. This algorithm takes a system parameters *params*, an identity ID_i, a message m_i and an individual signature σ_i as input, and outputs 1 or 0 for valid or invalid, respectively.

3.1.2 Security requirements for identity-based signature schemes. We review the usual security model of IBS [17,21] which is an extension of the usual notion of existential unforgeability under chosen-message attacks [22]. The security model mainly captures the following two attacks:

1. Adaptive chosen message attack: It allows an adversary to ask the signer to sign any message of its choice in an adaptive way, it can adapt its queries according to previous answers;
2. Adaptive chosen identity attack: It allows the adversary to forge a signature with respect to an identity chosen by the adversary.

Finally, the adversary could not provide a new message-signature pair with non-negligible advantage. The security for an IBS scheme is defined via the following game.

Game I (Unforgeability of IBS). This game is performed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ with respect to scheme (**Setup**, **Extract**, **Sign**, **Verify**), which captures the attacking scenario where a dishonest user who is allowed to have access to

the signing oracle for any desired messages and identities, but he is not able to obtain victim's private key, and wants to create a new valid signature.

Setup. Taking a security parameter k as input, the challenger $\mathcal{C}$ runs the Setup algorithm to obtain a master secret key *msk* and system parameters *params*. Then $\mathcal{C}$ sends *params* to the adversary $\mathcal{A}$, but keeps *msk* secret.

Queries. $\mathcal{A}$ makes a polynomially bounded number of the following queries in an adaptive manner.

- *Extraction queries*. Given an identity ID_i, the challenger returns the private key D_i corresponding to ID_i.
- *Signature queries*. Given an identity ID_i and a message m_i, $\mathcal{C}$ returns an individual signature σ_i on m_i with respect to ID_i.

Forgery. Eventually, $\mathcal{A}$ outputs an identity-based signature σ^* on a message m^* for an identity ID^*. We say that $\mathcal{A}$ wins Game I, iff.

(1) σ^* is a valid signature on message m^* under identity ID^*.

(2) ID^* has never been queried during the Extraction queries. And (ID^*,m^*) has never been queried during the Signature queries.

The advantage of $\mathcal{A}$ is defined as the probability that it wins in Game I.

Definition 3. An IBS scheme is said to satisfy the property of existential unforgeability against adaptive chosen-message attack and adaptive chosen-identity attack (EUF-IBS-CMA) if there is no probabilistic polynomial-time adversary $\mathcal{A}$ with non-negligible advantage in Game I.

3.2 Formal model of identity-based aggregate signature schemes

3.2.1 Definition of identity-based signature aggregate signature schemes. An IBAS scheme involves a PKG, an aggregating multiset of n users and an aggregate signature generator. It allows the generator to compress any n individual signatures along with a multiset of n message-identity pairs, which include on the same message from the same signer, into a single signature. An IBAS scheme is a tuple (**Setup**, **Extract**, **Sign**, **Verify**, **Agg**, **AggVerify**) based on the IBS scheme (**Setup**, **Extract**, **Sign**, **Verify**) by six polynomial-time algorithms with the following functionality:

- **Setup**, **Extract**, **Sign, Verify**. These algorithms are the same as those in the IBS scheme in Section 3.1.1.
- **Agg**. This algorithm is run by an aggregate signature generator and allows the generator to compress multiple individual signatures into an aggregate signature. It takes a system parameters *params*, n signatures $(\sigma_1,\ldots,\sigma_n)$ with each signature σ_i under an identity ID_i on a message m_i as input, and outputs an aggregate signature σ_{Agg} for the multiset of message-identity pairs $\{(m_1,\mathrm{ID}_1),\ldots,(m_n,\mathrm{ID}_n)\}$.
- **AggVerify**. This algorithm takes an aggregate signature σ_{Agg}, a multiset of n message-identity pairs $\{(m_1,\mathrm{ID}_1),\ldots,(m_n,\mathrm{ID}_n)\}$ as input, and outputs 1 if the aggregate signature is valid, or 0 otherwise.

3.2.2 Security requirements for identity-based aggregate signature schemes. An IBAS scheme should be secure against traditional existential forgery under adaptive chosen-message attack and adaptive chosen-identity attack. An unforgeability of IBAS is defined via the following unforgeability game which is performed between a challenger and an adversary. The adversary's goal is the existential forgery of an aggregate signature. Informally, it should be computationally infeasible for any adversary to produce a forgery. We formalize the security model as follows.

Game II (Unforgeability of IBAS). This game is performed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ with respect to scheme (**Setup**, **Extract**, **Sign, Verify**, **Agg**, **AggVerify**), which captures the attacking scenario where a dishonest user who is allowed to have access to the signing oracle for any desired messages and identities, wants to create a forgery without knowing the private keys of all the signers.

Setup. Taking a security parameter k as input, the challenger $\mathcal{C}$ runs the Setup algorithm to obtain a master secret key *msk* and system parameters *params*. Then $\mathcal{C}$ sends *params* to the adversary $\mathcal{A}$, but keeps *msk* secret.

Queries. $\mathcal{A}$ makes a polynomially bounded number of the following queries in an adaptive manner.

- *Extraction queries*. Given an identity ID_i, the challenger returns the private key D_i corresponding to ID_i.
- *Signature queries*. Given an identity ID_i and a message m_i, $\mathcal{C}$ returns a signature σ_i.

Forgery. Eventually, $\mathcal{A}$ outputs a multiset of n message-identity pairs $\{(m_1^*,\mathrm{ID}_1^*),\ldots,(m_n^*,\mathrm{ID}_n^*)\}$ and an aggregate signature σ_{Agg}^*. We say that $\mathcal{A}$ wins the game, iff.

(1) σ_{Agg}^* is a valid aggregate signature on message-identity pairs $\{(m_1^*,\mathrm{ID}_1^*),\ldots,(m_n^*,\mathrm{ID}_n^*)\}$, i.e., $\mathrm{AggVerify}(params,\sigma_{\mathrm{Agg}}^*, \{(m_1^*,\mathrm{ID}_1^*),\ldots,(m_n^*,\mathrm{ID}_n^*)\})=1$.

(2) At least one of the identities, without loss of generality, say $\mathrm{ID}_i^*\in L_{\mathrm{ID}}^*=(\mathrm{ID}_1^*,\ldots,\mathrm{ID}_n^*)$ has never been queried during the Extraction queries. And (ID^*,m^*) has never been queried during the Signature queries.

The advantage of $\mathcal{A}$ is defined as the probability that it wins in Game II.

Definition 4. An IBAS scheme is said to satisfy the property of existential unforgeability against adaptive chosen-message attack and an adaptive chosen-identity attack (EUF-IBAS-CMA) if there is no probabilistic polynomial-time adversary $\mathcal{A}$ with non-negligible advantage in Game II.

A New Identity-Based Signature Scheme

In this section, we propose a provably secure identity-based signature scheme which can be used to construct an unrestricted IBAS scheme.

4.1 Proposed basic identity-based signature scheme

The proposed IBS scheme consists of the following four concrete algorithms:

- **Setup.** Given a security parameter k, the private key generator (PKG) chooses a prime q, a cyclic additive group G_1 and a cyclic multiplicative group G_2 of prime order q, a random generator P in G_1, an admissible pairing $e: G_1\times G_1\rightarrow G_2$, and two cryptographic hash functions $H_1:\{0,1\}^*\rightarrow G_1$ and $H_2:\{0,1\}^*\rightarrow Z_q^*$. It also randomly chooses $s_1,s_2\in Z_q^*$, sets the master key $msk=(s_1,s_2)$, and computes $P_1=s_1P$ and $P_2=s_2P$. Finally, it broadcasts the system parameters, $params=(q,G_1,G_2,e,P,P_1,P_2,H_1,H_2)$.

- **Extract.** For a given identity ID_i, the PKG computes $Q_i = H_1(ID_i)$ and sets this user's private key D_i to be $(s_1Q_i, s_2Q_i) = (D_{i,1}, D_{i,2})$.
- **Sign.** To sign a message m_i with private key D_i, the signer with ID_i chooses $r_i \in Z_q^*$ and computes $U_i = r_iP$, $h_i = H_2(m_i, ID_i)$, $V_i = h_iD_{i,1} + r_iP_1$ and $W_i = D_{i,2} + r_iP_2$. The signature on m_i is $\sigma_i = (U_i, V_i, W_i)$.
- **Verify.** Upon receipt of an individual signature $\sigma_i = (U_i, V_i, W_i)$, the verifier computes $Q_i = H_1(ID_i)$ and $h_i = H_2(m_i, ID_i)$, and checks $e(V_i, P) = e(h_iQ_i + U_i, P_2)$ and $e(W_i, P) = e(Q_i + U_i, P_2)$. If both the equations hold, then the individual signature $\sigma_i = (U_i, V_i, W_i)$ is valid.

4.2 Security proof of the IBS scheme

The following theorem shows that in the random oracle model, our IBS scheme is existentially unforgeable against adaptive chosen-message attack and adaptive chosen-identity attack under the assumption that CDH problem in G_1 is intractable. Concretely, we show that if a probabilistic polynomial-time bounded adversary exists who can break our IBS scheme with non-negligible probability , we will be able to solve the computational Diffie-Hellman problem with non-negligible probability ′, which contradicts the CDH assumption.

Theorem 1. In the random oracle model, if there exists a polynomial-time adversary $\mathcal{A}$ who has an advantage in forging a signature of our IBS scheme in an attack modeled by Game I of Section 3.12 within a time at most t, after asking at most q_{H_i} times H_i ($i = 1, 2$) queries, q_E times Extraction queries and q_S times Signature queries, then the CDH problem in G_1 can be solved within time.

$$t' < 2(t + t_M(q_{H_1} + 2q_E + 4q_S)) + t_M + 2t_I$$

and with probability

$$\epsilon' \geq \frac{1}{(e(q_E+1))^2 q_{H_2}}(\epsilon - \frac{1}{q})^2 - \frac{1}{q}$$

where e is the base of the natural logarithm, t_M is the time of computing a scalar multiplication in G_1, and t_I is the time of computing an inversion in Z_q^*.

Proof. Using a similar proof technique in [17,23,24], we are going to construct a probabilistic polynomial-time algorithm $\mathcal{C}$ to solve the CDH problem by using the adversary $\mathcal{A}$ who can break our IBS scheme. Suppose that $\mathcal{C}$ is given an random instance of the CDH problem $(P, aP, bP) \in G_1^3$ for some unknown $a, b \in Z_q^*$. The task of $\mathcal{C}$ is to compute abP. $\mathcal{C}$ plays the role of $\mathcal{A}$'s challenger in Game I and interacts with $\mathcal{A}$ as follows:

Setup. $\mathcal{C}$ simulates the Setup algorithm as follows:

1. Choose a random value $s \in Z_q^*$ and sets $P_1 = aP$, $P_2 = sP$, where $a \in Z_q$ is unknown to $\mathcal{C}$.
2. Choose a cyclic group G_2 of prime order q, a bilinear map $e : G_1 \times G_1 \to G_2$.
3. Choose two hash functions H_1 and H_2 as random oracle.
4. Send the system parameters $params = (q, G_1, G_2, e, P, P_1, P_2, H_1, H_2)$ to $\mathcal{A}$.

Query. Proceeding adaptively, $\mathcal{A}$ is allowed to query the random oracles H_1, H_2, Extraction oracle and Signature oracle in a polynomial number of times. $\mathcal{C}$ simulates these oracles for $\mathcal{A}$ as follows:

***H_1* queries.** At any time, $\mathcal{A}$ can issue an H_1 query on an identity. To avoid collision and consistently respond to H_1 queries, $\mathcal{C}$ maintains a list L_{H_1} of tuples (ID, t, c, Q) which stores his responses to such queries. This list is initially empty. When querying the oracle H_1 on ID, $\mathcal{C}$ responds as follows:

1. If the query ID already appears on L_{H_1} in a tuple (ID, t, c, Q), $\mathcal{C}$ responds to $\mathcal{A}$ with $H_1(ID) = Q$.
2. Otherwise, $\mathcal{C}$ picks a random coin $c \in \{0, 1\}$ with $\Pr[c = 0] = \delta$.
   - If $c = 0$, then $\mathcal{C}$ randomly chooses $t \in Z_q^*$ and computes $Q = t(bP)$.
   - If $c = 1$, then $\mathcal{C}$ randomly chooses $t \in Z_q^*$ and computes $Q = tP$.

$\mathcal{C}$ adds the tuple (ID, t, c, Q) to the L_{H_1} and responds to $\mathcal{A}$ with $H_1(ID) = Q$.

***H_2* queries.** To respond to H_2 queries, $\mathcal{C}$ maintains a list L_{H_2} of tuples (ID, m, h), which is initially empty. When querying the oracle H_2 on (ID, m), $\mathcal{C}$ responds as follows:

1. If the query (ID, m) already appears on L_{H_2} in a tuple (ID, m, h), $\mathcal{C}$ responds to $\mathcal{A}$ with $H_2(m, ID) = h$.
2. Otherwise, $\mathcal{C}$ randomly chooses $h \in Z_q^*$, adds the tuple (ID, m, h) to L_{H_2} and responds to $\mathcal{A}$ with $H_2(m, ID) = h$.

Extraction queries. When $\mathcal{A}$ queries the private key corresponding to ID, $\mathcal{C}$ first finds the corresponding tuple (ID, t, c, Q) from the L_{H_1}.

1. If $c = 0$, $\mathcal{C}$ fails and aborts the simulation.
2. Otherwise, $\mathcal{C}$ computes $D_1 = tP_1$ and $D_2 = tP_2$, and responds to $\mathcal{A}$ with $D = (D_1, D_2)$.

Signature queries. When $\mathcal{A}$ makes a Signature query on m for ID, $\mathcal{C}$ randomly chooses $r \in Z_q^*$ and computes $U = (rP - hQ)$, $V = rP_1$. Then, $\mathcal{C}$ computes $W = s(Q + U)$ and responds to $\mathcal{A}$ with signature $\sigma = (U, V, W)$.

Forgery. Eventually, $\mathcal{A}$ outputs a forged signature $\sigma^* = (U^*, V^*, W^*)$ on a message m^* for an identity ID^*. $\mathcal{C}$ finds the corresponding tuple (ID^*, t^*, c^*, Q^*) from the L_{H_1}. If $c^* \neq 0$, $\mathcal{C}$ fails and aborts. Otherwise, by applying the forking lemma [25], after replaying $\mathcal{A}$ with the same random tape but different choices of oracle H_2, $\mathcal{C}$ can get two valid signatures $(m^*, ID^*, h^*, \sigma^* = (U^*, V^*, W^*))$ and $(m^*, ID^*, h'^*, \sigma'^* = (U^*, V'^*, W'^*))$ such that $h^* \neq h'^*$. Now, since both forgeries are valid, we have

$$e(V^*, P) = e(h^*Q^* + U^*, P_1)$$

$$e(V'^*, P) = e(h'^*Q^* + U^*, P_1)$$

Combining the above two equations, we have

$$e(V^* - V'^*, P) = e((h^* - h'^*)Q^*, P_1)$$

Note that $P_1 = aP$ and $H_1(ID^*) = Q^* = t^*(bP)$ since $c^* = 0$. We have

$$e(V^* - V'^*, P) = e((h^* - h'^*) \cdot t^* bP, aP)$$

$$= e((h^* - h'^*) t^* abP, P)$$

which implies

$$V^* - V'^* = (h^* - h'^*) t^* abP$$

Consequently, $\mathcal{C}$ could solve the CDH by computing

$$abP = (h^* - h'^*)^{-1} t^{*} - 1(V^* - V'^*)$$

Probability analysis. It remains to evaluate the probability $'$ that $\mathcal{C}$ solves the given instance of CDH. First, we analyze the events needed for $\mathcal{C}$ to succeed before the rewinding.

- E_1: $\mathcal{C}$ does not abort as a result of any of $\mathcal{A}$'s Extraction query.
- E_2: $\mathcal{A}$ generates a valid and nontrivial aggregate signature forgery σ^*_{Agg} for $\{(m_1^*, \mathrm{ID}_1^*), \cdots, (m_n^*, \mathrm{ID}_n^*)\}$.
- E_3: Event E_2 occurs and $c_1^* = 0$, $c_j^* = 1$ for $2 \leq j \leq n$, where for each i, c_i^* is the c-component of the tuple containing ID_i on the L_{H_1}.

$\mathcal{C}$ succeeds before the rewinding if all of these events occur. The probability $\epsilon = \Pr[E_1 \wedge E_2 \wedge E_3]$ is decomposed as

$$\Pr[E_1 \wedge E_2 \wedge E_3] = \Pr[E_1] \Pr[E_2|E_1] \Pr[E_3|E_1 \wedge E_2]$$

The following claims give a lower bound for each of these terms.

Claim 1. The probability that algorithm $\mathcal{C}$ does not abort as a result of $\mathcal{A}$'s Extraction query is at least $(1-\delta)^{q_E}$. Hence we have $\Pr[E_1] = (1-\delta)^{q_E}$.

Proof. Since $\mathcal{A}$ makes at most q_E queries to the Extraction oracle and $\Pr[c=1] = (1-\delta)$, the probability that algorithm $\mathcal{C}$ does not abort as a result of $\mathcal{A}$'s Extraction queries is at least $(1-\delta)^{q_E}$.

Claim 2. If $\mathcal{C}$ does not abort as a result of $\mathcal{A}$'s Extraction query, then $\mathcal{A}$'s view is identical to its view in the real attack. Hence, $\Pr[E_2|E_1] \geq -1/q$.

Proof. Since the probability that $\mathcal{A}$ generates a valid and nontrivial signature for (m^*, ID^*) without asking H_2 oracle in advance is less than $1/q$, the probability that $\mathcal{A}$ outputs a valid forgery σ^* after querying $H_2(m^*, \mathrm{ID}^*)$ is at least $-1/q$.

Claim 3. The probability that $\mathcal{C}$ does not abort after $\mathcal{A}$ outputs a valid and nontrivial forgery is at least δ. Hence, $\Pr[E_3|E_1 \wedge E_2] \geq \delta$.

Proof. After $\mathcal{A}$ outputs a valid and nontrivial forgery, algorithm $\mathcal{C}$ does not abort if and only if $c^* = 0$. Since $\Pr[c^* = 0] = \delta$, the probability that $\mathcal{C}$ does not abort is at least δ.

Combining all of the above results, the probability $\epsilon = \Pr[E_1 \wedge E_2 \wedge E_3]$ is at least

$$(1-\delta)^{q_E} \delta \cdot (-\frac{1}{q})$$

Therefore, in the first run of $\mathcal{A}$, $\mathcal{C}$ does not abort with probability.

$$\epsilon \geq (1-\delta)^{q_E} \delta \cdot (-\frac{1}{q})$$

According to the general forking lemma [25], the probability that $\mathcal{C}$ obtains two successful forgeries of $\mathcal{A}$ and does not abort is

$$\epsilon' > \varepsilon(\frac{\varepsilon}{q_{H_2}} - \frac{1}{q})$$

where $\epsilon = \Pr[E_1 \wedge E_2 \wedge E_3]$. When $\delta = 1/(q_E + 1)$, $(1-\delta)^{q_E} \delta$ is maximized at

$$(1 - \frac{1}{q_E + 1})^{q_E} \frac{1}{q_E + 1}$$

$$\geq \frac{1}{e} \cdot \frac{1}{q_E + 1}.$$

Therefore, the probability of solving the CDH problem is

$$\epsilon' \geq \frac{1}{(e(q_E+1))^2 q_{H_2}} (\epsilon - \frac{1}{q})^2 - \frac{1}{q}$$

which is non-negligible if is non-negligible.

Algorithm $\mathcal{C}$'s running time is roughly the same as $\mathcal{A}$'s running time plus the time it takes to respond to hash queries, Extraction queries and Signature queries, and the time to transform $\mathcal{A}$'s final forgery into the CDH solution. The H_1 query requires a scalar multiplication. The Extraction query requires two scalar multiplications. The Signature query requires 4 scalar multiplications and the output phase requires a scalar multiplication and two inversions. Hence, the total running time is at most $2(t + t_M(q_{H_1} + 2q_E + 4q_S)) + t_M + 2t_I$.

A New Identity-Based Aggregate Signature Scheme

5.1 Proposed identity-based aggregate signature scheme

Now, we construct an IBAS scheme using our basic IBS scheme constructed in the previous section.

- **Setup, Extract, Sign, Verify.** These algorithms are the same as those in our proposed IBS scheme.
- **Agg.** Begin with n signatures $(\sigma_1, \ldots, \sigma_n)$ along with n message-identity pairs $\{(m_1, \mathrm{ID}_1), \cdots, (m_n, \mathrm{ID}_n)\}$ where $\sigma_i = (U_i, V_i, W_i)$ is the individual signature on message m_i for identity $\mathrm{ID}_i, i = 1, \ldots, n$. The aggregate signature generator computes $U = \sum_{i=1}^n U_i$, $V = \sum_{i=1}^n V_i$ and $W = \sum_{i=1}^n W_i$, and outputs $\sigma_{Agg} = (U, V, W)$ as an aggregate signature for message-identity pairs $\{(m_1, \mathrm{ID}_1), \cdots, (m_n, \mathrm{ID}_n)\}$.
- **AggVerify.** To verify the validity of an aggregate signature $\sigma_{Agg} = (U, V, W)$ for message-identity pairs $\{(m_1, \mathrm{ID}_1), \cdots, (m_n, \mathrm{ID}_n)\}$, the verifier computes $Q_i = H_1(\mathrm{ID}_i)$, $h_i = H_2(m_i, \mathrm{ID}_i)$, for $i = 1, \ldots, n$, and checks.

$$e(V,P)=e(\sum_{i=1}^{n} h_i Q_i + U, P_1)$$

and

$$e(W,P)=e(\sum_{i=1}^{n} Q_i + U, P_2)$$

If both the equations hold, then the aggregate signature σ_{Agg} is valid.

5. 2 Security proof of the IBAS scheme

In this subsection, we are going to prove the security of our identity based aggregate signature scheme. The proof outline is as follows.

We assume on the contrary that our IBAS scheme is not EUF-IBAS-CMA secure. That is, assume there exists a polynomial time bounded adversary $\mathcal{A}$ who can forge a signature in IBAS under the adaptive chosen message and chosen identity attacks. The proof's goal is to show that under this assumption, our IBS scheme is not EUF-IBS-CMA secure.

Theorem 2. If there exists an adversary $\mathcal{A}$ who has an advantage in forging an aggregate signature of our IBAS scheme in the chosen aggregate modeled by Game II within a time at most t, after asking at most q_{H_i} times H_i ($i = 1, 2$) queries, q_E times Extraction queries, q_S times Signature queries and at most N signers, then there exists an algorithm which in forging a signature of our IBS scheme in an attack modeled wins Game I within time.

$$t' < t + t_M(q_{H_1} + 2q_E + 4q_S + 2N)$$

and with advantage

$$\epsilon' > \frac{\epsilon}{(q_E+N)e}$$

where e and t_M denote the same quantities as in Theorem 1.

Proof. Here we follow the idea from [17,26,27]. Suppose that $\mathcal{A}$ is a forger who breaks the IBAS scheme. By using $\mathcal{A}$, we will construct an algorithm $\mathcal{C}$ which outputs a forgery of our IBS scheme. Algorithm $\mathcal{C}$ performs the following simulation by interacting with the adversary $\mathcal{A}$.

Setup. It is the same as that described in the proof of Theorem 1.

***H_1* queries.** To respond to H_1 queries, $\mathcal{C}$ maintains a list L_{H_1} of tuples (ID, t, c, Q), which is initially empty. When $\mathcal{A}$ queries the oracle H_1 on ID, $\mathcal{C}$ responds as follows:

1. If the query ID already appears on the L_{H_1} in a tuple (ID, t, c, Q), $\mathcal{C}$ responds with $H_1(\mathrm{ID}) = Q$.
2. Otherwise, $\mathcal{C}$ picks a random coin $c \in \{0,1\}$ with $\Pr[c = 0] = \delta$.

- If $c = 0$ then $\mathcal{C}$ chooses $t \in Z_q^*$ and computes $Q = t(bP)$.
- If $c = 1$ then $\mathcal{C}$ chooses $t \in Z_q^*$ and computes $Q = tP$.

$\mathcal{C}$ adds the tuple (ID, t, c, Q) to the L_{H_1} and responds to $\mathcal{A}$ with $H_1(\mathrm{ID}) = Q$.

***H_2* queries, Extraction queries, Signature queries.** When $\mathcal{A}$ make H_2 queries, Extraction queries, Signature queries, $\mathcal{C}$ responds as those defined in the proof of Theorem 1.

Forgery. Eventually, $\mathcal{A}$ outputs an aggregate signature $\sigma^*_{Agg} = (U^*, V^*, W^*)$ together with $\{(m_1^*, \mathrm{ID}_1^*), \cdots, (m_n^*, \mathrm{ID}_n^*)\}$.

$\mathcal{C}$ recovers the corresponding tuples $(\mathrm{ID}_i^*, t_i^*, c_i^*, Q_i^*)$ from the L_{H_1} and the corresponding tuples $(\mathrm{ID}_i^*, m_i^*, h_i^*)$ from the L_{H_2} for all i, $1 \le i \le n$.

It requires that there exists $k \in \{1, \ldots, n\}$ such that $c_j^* = 1$ for $j = 1, \ldots, n, j \neq k$, $c_k^* = 0$ (without loss of generality, we let $k = 1$), $\mathcal{A}$ has not made a query Signature oracle on (ID_1^*, m_1^*) and $\mathrm{AggVerify}(params, \sigma^*_{\mathrm{Agg}}, \{(m_1^*, \mathrm{ID}_1^*), \ldots, (m_n^*, \mathrm{ID}_n^*)\}) = 1$. Therefore, the aggregate signature $\sigma^*_{Agg} = (U^*, V^*, W^*)$ should satisfy the aggregate verification equations.

$$e(V^*,P)=e(\sum_{i=1}^{n} h_i^* Q_i^* + U^*, P_1)$$

$$e(W^*,P)=e(\sum_{i=1}^{n} Q_i^* + U^*, P_2)$$

$\mathcal{C}$ sets $V_j^* = t_j^* h_j^* P_1$ and $W_j^* = sQ_j^*$. Obviously (V_j^*, W_j^*) satisfy the equations $e(V_j^*, P) = e(h_j^* Q_j^*, P_1)$ and $e(W_j^*, P) = e(Q_j^*, P_2)$ for $2 \le j \le n$. Then, $\mathcal{C}$ constructs as $(V^* - \sum_{j=2}^{n} V_j^*)$ and W'^* as $W^* - \sum_{j=2}^{n} W_j^*$. $\sigma'^* = (U^*, V'^*, W'^*)$ is a valid individual signature on m_1^* for ID_1^* since it satisfies the verification equations as follows:

$$e(V'^*,P)=e(h_1^* Q_1^* + U^*, P_1)$$

$$e(W'^*,P)=e(Q_1^* + U^*, P_2)$$

Finally, $\mathcal{C}$ outputs σ'^* as a forgery of the IBS scheme.

Probability analysis. Similar to the analysis in **Theorem 1**, we analyze three events needed for $\mathcal{C}$ to succeed.

- E_1: $\mathcal{C}$ does not abort as a result of any of $\mathcal{A}$'s Extraction query.
- E_2: $\mathcal{A}$ generates a valid and nontrivial aggregate signature forgery σ^*_{Agg} for $\{(m_1^*, \mathrm{ID}_1^*), \cdots, (m_n^*, \mathrm{ID}_n^*)\}$.
- E_3: Event E_2 occurs and $c_1^* = 0$, $c_j^* = 1$ for $2 \le j \le n$, where for each i, c_i^* is the c-component of the tuple containing ID_i on the L_{H_1}.

$\mathcal{C}$ succeeds if all of these events happen. The probability $\Pr[E_1 \wedge E_2 \wedge E_3]$ is the same as in Theorem 1

Claim 1. The probability that $\mathcal{C}$ does not abort as a result of $\mathcal{A}$'s Extraction query is at least $(1-\delta)^{q_E}$. Hence, $\Pr[E_1] \ge (1-\delta)^{q_E}$.

Claim 2. If $\mathcal{C}$ does not abort as a result of $\mathcal{A}$'s Extraction query and Signature queries, then $\mathcal{A}$'s view is identical to its view in the real attack. Hence, $' > \frac{}{(q_E+N)e}$.

Claim 3. The probability that $\mathcal{C}$ does not abort after $\mathcal{A}$ outputs a valid and nontrivial forgery is at least $(1-\delta)^{N-1} \cdot \delta$.

Proof. Algorithm $\mathcal{C}$ will abort unless $\mathcal{A}$ generates a forgery such that $c_1^* = 0$ and $c_j^* = 1$ for $2 \le j \le n$. Thus, $c_1^* = 0$ occurs with

probability δ. And the probability that $c_j^* = 1$, for $2 \leq j \leq n$, is at least $(1-\delta)^{N-1}$. Therefore

$$\Pr[E_3|E_1 \wedge E_2] \geq (1-\delta)^{N-1} \cdot \delta.$$

.

Combining all of the above results, the advantage $'$ that $\mathcal{C}$ produces the correct answer is at least $\delta(1-\delta)^{q_E}(1-\delta)^{N-1} = \delta(1-\delta)^{q_E+N-1}$ which is maximized at $\delta = 1/(q_E+N)$. Therefore, the advantage $'$ is

$$' \geq (1-\frac{1}{q_E+N})^{q_E+N-1}\frac{1}{q_E+N}$$

$$' \geq (1-\frac{1}{q_E+N})^{q_E+N-1}\frac{1}{q_E+N}$$

as required.

With Theorems 1 and 2, we can get the conclusion that the proposed IBAS scheme is secure against adaptively chosen-message and chosen-identity attacks under the hardness assumption of CDH problem in the random oracle model.

5.3 Performance analysis

Computation cost and aggregate signature size are two important parameters affecting the efficiency of an IBAS scheme. In this section, we compare our scheme with the existing unrestricted identity-based aggregate signature schemes [9,14–17,19,20] from the aspects of aggregate signature size and computation cost in signature phase and aggregate signature verify phase, respectively. Detailed comparisons are summarized in Table 1. Here we only consider the costly operations (i.e., pairing operation, MapToPoint hash operation and multiplication operation in G_1) and omit the computational efforts which can be pre-computed. We use notations as follows:

- pair: the time for performing a pairing operation.
- mul_{G_1}: the time for performing a scalar multiplication in group G_1.
- mtp: the time for performing a map-to-point hash operation.
- $|G_1|$: the length of element in group G_1.
- $|m|$: the length of the message m.
- $|\text{ID}|$: the length of the identity ID.
- t: the number of distinct signers.
- n: the number of aggregated signatures.

From Table 1, we can see that the aggregate signature length of both of the scheme in [20] and our scheme is the same as that of a single individual signature regardless of the number n of signatures while that of the other schemes is directly proportional to either the number n of signatures or the number t of signers.

We also can observe that although the aggregate signature size overhead of Hohenberger et al.'s scheme [20] is better than that of ours (which is the shortest among the protocols under comparison), their scheme is less efficient in signing and aggregate verifying, which requires $O(|m|)$ pairing operations to generate a signature and $O(n(|m|+|\text{ID}|))$ pairing operations to verify an aggregate signature. Our IBAS scheme requires no pairing operations for the signer and only four pairing operations for the verifier. As the pairing computation is the most time consuming in pairing-based cryptosystems [28], the computation overhead in our scheme is much faster than that in the scheme [20]. Therefore, the proposed scheme is more practical.

Conclusions

In this paper, we proposed a new identity-based signature scheme that is provably secure in the random oracle model under the CDH assumption. We constructed an identity-based aggregate signature scheme using our IBS as the base signature scheme. The proposed IBAS enjoys significant advantages: aggregation is very general in that it allows for the aggregation of any multiple signatures from various users on various messages into a single compact signature; the aggregation operation does not require any restricted; AS meets the merit of signatures in ID-PKC which is free from the public key certificate management burden. The most important point is the compared with previous unrestricted IBAS schemes, our proposed scheme is the first IBAS scheme which satisfies both constant length aggregate signature and constant pairing operations. The security analysis has been provided and shown that the proposed schemes are secure against adaptive chosen-message attack and chosen-identity attack in the random oracle model. These features render our IBAS scheme an efficient solution to reduce bandwidth and storage, and are especially attractive for mobile devices like sensors, cell phones and PDAs where communication is more power-expensive than computation and contributes significantly to reducing battery life. Moreover, our scheme can adaptively work as a multi-signature scheme or a proxy signature scheme or a sequential aggregate scheme without any modifications.

Table 1. Comparisons of computation cost and aggregate signature size.

Scheme	Sign Time	AggVerify Time	Aggregate Signature Size
Cheon et al. [9]	$1\,\text{mul}_{G_1}$	$(t+1)\,\text{pair} + t\,\text{mul}_{G_1}$	$(n+1)\lvert G_1\rvert$
Xu et al. [14]	$1\,\text{mul}_{G_1} + 1\,\text{mtp}$	$(n+2)\,\text{pair} + n\text{mtp}$	$(n+1)\lvert G_1\rvert$
Herranz [15]	$1\,\text{mul}_{G_1} + 1\,\text{mtp}$	$(t+1)\,\text{pair} + t\,\text{mul}_{G_1} + n\text{mtp}$	$(t+1)\lvert G_1\rvert$
Kar [16]	$1\,\text{mul}_{G_1}$	$2\,\text{pair} + t\,\text{mul}_{G_1}$	$(2n+1)\lvert G_1\rvert$
Shim [17]	$1\,\text{mul}_{G_1}$	$2\,\text{pair} + n\,\text{mul}_{G_1}$	$(n+1)\lvert G_1\rvert$
Kang [19]	$1\,\text{mul}_{G_1}$	$3\,\text{pair} + t\,\text{mul}_{G_1}$	$(n+1)\lvert G_1\rvert$
Hohenberger et al. [20]	$\lvert m\rvert$ pair	$(n(\lvert m\rvert + \lvert\text{ID}\rvert - 1) + 1)\,\text{pair}$	$1\,\lvert G_1\rvert$
Our scheme	$1\,\text{mul}_{G_1}$	$4\,\text{pair} + t\,\text{mul}_{G_1}$	$3\lvert G_1\rvert$

doi:10.1371/journal.pone.0110100.t001

Acknowledgments

The authors would like to thank anonymous reviewers for their constructive suggestions.

Author Contributions

Contributed reagents/materials/analysis tools: YY QZ. Conceived and designed the experiments: YY QZ. Analyzed the data: YY QZ HH. Wrote the paper: YY QZ HH. Proved the security of the scheme: YY QZ.

References

1. Boneh D, Gentry C, Shacham H, Lynn B (2003) Aggregate and verifiably encrypted signatures from bilinear maps. In: Proc. Advances in Cryptology – EUROCRYPT 2003, Springer LNCS 2656, 416–432.
2. Niu S, Wang C, Yu Z, Cao S (2013) Lossy data aggregation integrity scheme in wireless sensor networks. Computers and Electrical Engineering 39(6):1726–1735.
3. Bellare M, Namprempre C, Neven G (2007) Unrestricted Aggregate Signatures. In: Proc. 34th International Colloquium on Automata, Languages and Programming (ICALP 2007), Springer LNCS 4596, 411–422.
4. Liu JK, Yuen TH, Au MH, Susilo W (2014) Improvements on an authentication scheme for vehicular sensor networks. Expert Systems with Applications 41(5): 2559–2564.
5. Zhang L, Qin B, Wu Q, Zhang F (2010) Efficient many-to-one authentication with certificateless aggregate signatures. Computer Networks 54(14): 2482–2491.
6. Shao Z (2008) Fair exchange protocol of signatures based on aggregate signatures. Computer Communications 31: 1961–1969.
7. Wei L, Zhu H, Cao Z, Dong X, Jia W, et al. (2014) Security and privacy for storage and computation in cloud computing. Information Sciences 258: 371–386.
8. Shamir A (1984) Identity-based cryptosystem and signature scheme. In: Proc. Proceedings of CRYPTO '84 on Advances in Cryptology, Springer LNCS 196, 47–53.
9. Cheon JH, Kim Y, Yoon HJ (2004) A new ID-based signature with batch verification. Cryptology ePrint Archive. Available: http://eprint.iacr.org/2004/131.pdf.
10. Dou B, Chen CH, Zhang H, Xu C (2012) Identity-based sequential aggregate signature scheme based on RSA. International journal of innovative computing information and control 8(9): 6401–6413.
11. Tsai JL, Lo NW, Wu TC (2013) New Identity-Based Sequential Aggregate Signature Scheme from RSA. In: Proc. 2013 International Symposium on Biometrics and Security Technologies. 136–140.
12. Gentry C, Ramzan Z (2006) Identity-based aggregate signature. In: Proc. 9th International Conference on Theory and Practice of Public-Key Cryptography (PKC 2006), Springer LNCS 3958, 257–273.
13. Bagherzandi A, Jarecki S (2010) Identity-based aggregate and multisignature schemes based on RSA. In: Proc. 13th International Conference on Practice and Theory in Public Key Cryptography (PKC 2010), Springer LNCS 6056, 480–498.
14. Xu J, Zhang Z, Feng D (2005) ID-based aggregate signatures from bilinear pairings. In: Proc. 4th International Conference on Cryptology and Network Security (CANS 2005), Springer LNCS 3810,110–119.
15. Herranz J (2006) Deterministic identity-based signatures for partial aggregation. The Computer Journal 49(3): 322–330.
16. Kar J (2012) Provably secure identity-based aggregate signature scheme. In: Proc. 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover. 137–142.
17. Shim KA (2010) An ID-based aggregate signature scheme with constant pairing computations. The Journal of Systems and Software 83(10): 1873–1880.
18. Selvi SSD, Vivek SS, Shriram J, Rangan CP (2010) Efficient and provably secure identity based aggregate signature schemes with partial and full aggregation. Cryptology ePrint Archive, Available: http://eprint. iacr.org/2010/461.pdf.
19. Kang B (2012) On the security of some aggregate signature schemes. Journal of Applied Mathematics 2012: Article ID 416137.
20. Hohenberger S, Sahai A, Waters B (2013) Full Domain Hash from (Leveled) Multilinear Maps and Identity-Based Aggregate Signatures. In: Proc. 33rd Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 2013), Springer LNCS 8042, 494–512.
21. Cha JC, Cheon JH (2003) An identity-based signature from gap Diffie-Hellman groups. In: Proc. 6th International Workshop on Theory and Practice in Public Key Cryptography (PKC 2003), Springer LNCS 2567, 18–30.
22. Goldwasser S, Micali S, Riverst R (1988) A digital signature scheme secure against adaptive chosen message attacks. SIAM Journal of Computing 17(2): 281–308.
23. Tian H, Chen X, Zhang F, Wei B, Jiang Z, et al. (2013) A non-delegatable strong designated verifier signature in ID-based setting for mobile environment. Mathematical and Computer Modelling 58 (5–6): 1289–1300.
24. He D, Chen Y, Chen J (2013) An efficient certificateless proxy signature scheme without pairing. Mathematical and Computer Modelling 57: 2510–2518.
25. Bellare M, Neven G (2006) Multi-signatures in the plain public-key model and a general forking lemma. In: Proc. Proceedings of the 13th ACM conference on Computer and communications security. 390–399.
26. Xiong H, Guan Z, Chen Z, Li F (2013) An efficient certificateless aggregate signature with constant pairing computations. Information Sciences 219: 225–235.
27. Tu H, He D, Huang B (2014) Reattack of a certificateless aggregate signature scheme with constant pairing computations. The Scientific World Journal 2014: Article ID 343715.
28. He D, Chen J, Hu J (2011) An ID-based proxy signature scheme without bilinear pairings. Annalas of Telecomunications 66: 657–662.