Practical Certificateless Aggregate Signatures From Bilinear Maps*

Zheng Gong[1], Yu Long[2], Xuan Hong[2] and Kefei Chen[2]

[1] Distributed and Embedded Security Group, Faculty of EEMCS,
University of Twente, The Netherlands.
[2] Department of Computer Science and Engineering,
Shanghai Jiaotong University, Shanghai, 200240
z.gong@utwente.nl,{longyu,xuanhong,kfchen}@sjtu.edu.cn

Abstract. In some restrictive environments, such as sensor networks, each sensor submits the newest information to the server, every message must be authenticated to immune forgery and replay attacks. But the regular signatures need to be saved and verified individually, which will heavily add the costs of the computation, storage and communication than the plain text mode in the constraint devices. Aggregate signature (AS) makes towards solving the above problem because anyone can aggregate n individual signatures on n distinct messages which are signed by n distinct signers, into a single compact signature σ.

In this paper, two practical certificateless aggregate signature schemes, which are the first aggregate signature schemes in the CL-PKC, are proposed from bilinear maps. The first scheme $\mathcal{CAS}$-1 reduces the costs of communication and signer-side computation but loses on the storage, while $\mathcal{CAS}$-2 minimizes the storage but sacrifices the communication. One can choose either of the above schemes by the consideration of the implementation requirement. Our schemes do not need the public key certificate anymore and achieve the trust level 3, the same level with traditional PKI. Both of the schemes are proven secure in the random oracle model (ROM) by assuming the intractability of the computational Diffie-Hellman (CDH) problem over the groups with bilinear maps.

Keywords Digital signature, Aggregate signature, Certificateless, Authentication.

1 Introduction

In nowadays network applications, digital signatures are widely used to provide authentication and integrity properties on the messages. A well-designed

* A preliminary version of this paper appears in SNPD 2007, IEEE Computer Society Proceedings [10]. This is the full version. The first author acknowledges the financial support of SenterNovem for the ALwEN project, grant PNE07007. The authors are partially supported by NSFC (NO.60703030,60803146) and the National Laboratory for Modern Communications Science Foundation of China (NO.51436040405JW0304).

signature must be as short as possible to save the communication bandwidth and storage. In some special applications, such as sensor nets which are used to detect the status in danger environments, each sensor submits the newest information to the server, we need the information is signed by the sensor to immune forgery and replay attacks which may trigger a false alarm or conceal a real danger. But this promotion will heavily add the costs of the computation and communication in the system. From the statistics in [2, 11], each bit transmitted consumes as much power as executing 800-1000 arithmetic instructions. In these cases, an efficient signature scheme is required that the signatures sizes and verifications costs are not linearly increase with the number of the signing messages. Aggregate signature is a reasonable technique towards solving this problem.

Aggregate Signature. Aggregate signature (AS) scheme, which is first introduced by Boneh et.al in [4], is a digital signature scheme with the additional property that anyone can aggregate n individual signatures (a sequence $\sigma_1, \sigma_2, \cdots, \sigma_n$) on n distinct messages $(m_1, m_2, \cdots, m_n)$ which are signed by n distinct signers, into a single compact signature σ. For all $i = 1, 2, \cdots, n$, each of individual signatures (m_i, σ_i) can check its correctness by the corresponding public key pk_i. There is an aggregate verification algorithm that takes input $\{\sigma, (pk_i, m_i)|i = 1, 2, \cdots, n\}$ then returns valid or not. Aggregation property is useful to reduce computation, communication and storage. Consider some special environments, such as PDAs, cell phones and sensors, the limitation of battery life is more constraint than the processor speed. An aggregate signature will be applicable with these communication limited scenarios. We notice that a digital signature with *batch verification* [6] goes similar but it does not have the property that anyone can compact n distinct individual signatures into a single verifiable signature.

Certificateless Signature. To solve key escrow problem while maintain the advantages of identity-based public key cryptosystem (ID-PKC) [14, 6], Al-Riyami and Paterson provided a new certificateless public key cryptosystem (CL-PKC) in [1]. In contrast to directory-based public key system (DB-PKC), the user's public key does not need any certificate to authenticate its validity since it is self-certificated. In ID-PKC, there is a trusted third party called the Private Key Generator (PKG) must be completely believed, because PKG is so powerful that it has the knowledge of all users' secret keys.

In CL-PKC, there exists a less powerful trusted third party, which is called Key Generate Center (KGC). KGC has the master-key to generate a user's *partial private key* D_i, which is computed from the user's identity ID_i. The partial

private key should be securely sent to the user. Afterward, the user adds his private secret information into the received partial private key, then derives his full private key S_i. Correspondingly, the user combines his secret information with the KGC's public key to generate his public key. In this sense, KGC knows nothing about the user's private key, which means key escrow problem does not exist any more.

Table 1. The comparison of the public key cyrptosystems

Cryptosystem	DB-PKC	ID-PKC	CL-PKC
Trust Level	Level 3	Level 1	Level 3
Key Distribution Channel	Authentic	Authentic and Private	Authentic
Retrieve Public Key	Directory	Communication	Communication
Public Key includes ID	No	Yes	Yes

After the intuitive work [1], many certificateless public key signature (CL-PKS) schemes are proposed, such as [17, 12]. The advantages of CL-PKC make them more competitive and feasible in many practical applications.

Our Contribution. In this paper, we propose two certificateless aggregate signature (CAS) schemes, which are the first general aggregate signature schemes in the CL-PKC. The first scheme $\mathcal{CAS}$-1 reduces the costs of communication and signer-side computation but loses on the storage, while $\mathcal{CAS}$-2 minimizes the storage but sacrifices the communication. We can choose one of the above schemes by the consideration of which advantage is the most important in the implementation. Compare with the traditional PKI-based scheme [4], our schemes do not need the public key certificate anymore. According to the CL-PKC, our schemes achieve the trust level 3 [9], the same level with the traditional PKI. In formal, both of the schemes are proven secure in the random oracle model (ROM) by assuming the intractability of the computational Diffie-Hellman (CDH) problem over the groups with bilinear maps, without using the forking lemma technique [13].

Related Work. Boneh et. al first introduced an aggregate signature from bilinear maps in [3], and then in a survey paper [5], Boneh et. al also presented a modification version based on [3]. The schemes is very simple, but it has a disadvantage that the verification costs will increase linearly ($O(n)$) with the number of messages (n) in the aggregated signature. Subsequent to these initial work, many improved schemes were proposed, such as [7, 15, 16], some of them are based on ID-PKC. Compares with the certificate-based scheme, identity-based

scheme does not need certificates' storage and public key verification anymore. Recently, an efficient identity-based aggregate signature was proposed by Gentry and Ramzan in [8]. Their scheme takes advantage that the aggregate verification requires only three times pairing computations, regardless of the number of messages in the aggregated signature.

2 Preliminaries

2.1 Bilinear Maps

Our schemes use a bilinear map, which is often called a "pairing". Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two additive cyclic groups with the same prime order q. Let $\hat{e}$ be a bilinear map such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \leftarrow \mathbb{G}_2$. A map $\hat{e}$ has the following properties:

1. Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, for all $P, Q \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$.
2. Non-degeneracy: $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
3. Symmetric: $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in \mathbb{G}_1$.
4. Admissible: $\hat{e}(., .)$ is efficiently computable.

2.2 Computational Assumption

The security of our schemes is based on the assuming intractability of the computational Diffie-Hellman (CDH) problem.

Definition 1. ***(CDH Problem).*** *Given $P, aP, bP \in \mathbb{G}_1$, an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, compute abP(for unknown randomly chosen $a, b \in \mathbb{Z}_q$).*

We say that the CDH problem is (t, ϵ)-hard if there is no algorithm solved it in polynomial time at most t with probability no more than ϵ.

2.3 Certificateless Aggregate Signature

Here we give the notion of CAS scheme. The scheme is defined by eight polynomial time bound algorithms: Setup, Set-Secret-Value, Set-Public-Key, Partial-Private-Key-Extract, Set-Full-Private-Key, Sign, Agg and Ver. Different from the standard certificateless signature scheme, we use the public key binding technique in Partial-Private-Key-Extract [1]. Through the binding, the scheme can achieve the trust level 3 [9], the same level with traditional PKI. Thus in our schemes, the forgery of one's public key by a malicious KGC will be detectable since one public key is binding to one partial private key. Moreover, the private channel between KGC and user is unnecessary (just need an authentic channel). The details of the algorithms describe as follows.

1. Setup: This algorithm inputs security parameter k, then returns the system parameters params and KGC's secret value master-key.
2. Set-Secret-Value: This algorithm inputs params and ID_A, outputs A's secret value x_A.
3. Set-Public-Key: This algorithm inputs params and x_A, outputs A's public key P_A.
4. Partial-Private-Key-Extract: This algorithm inputs params, master-key, public key P_A and an identifier ID_A for entity A, returns a partial private key D_A.
5. Set-Full-Private-Key: This algorithm inputs params, D_A and x_A, outputs A's full private key S_A.
6. Sign: This algorithm inputs an accepted message m, params, a user's identifier ID_i and the full private key S_i, outputs a signature σ_i.
7. Agg: For $i = 1, 2, \cdots, n$, inputs n distinct users' public keys and identifiers $\{(P_i, ID_i)|i = 1, 2, \cdots, n\}$, n distinct messages $\{m_i|i = 1, 2, \cdots, n\}$ and n individual signatures $\{\sigma_i|i = 1, 2, \cdots, n\}$, outputs a condensed signature σ.
8. Ver: For $i = 1, 2, \cdots, n$, inputs n distinct messages $\{m_i|i = 1, 2, \cdots, n\}$, a condensed signature σ, params, n distinct users' public keys and identifiers $\{(P_i, ID_i)|i = 1, 2, \cdots, n\}$, outputs true if the signature σ is valid, otherwise returns false.

Due to the notion of aggregate signature [4], an aggregate signature σ is declared valid only if the aggregator who created σ was given all valid individual signatures $\{\sigma_i|i = 1, 2, \cdots, n\}$. Thus, an aggregate signature provides non-repudiation at once on many different messages by many users.

2.4 Security Model of Certificateless Aggregate Signature Schemes

In CL-PKC, there are two types of adversaries with different capabilities [1]. In our security analysis, These adversaries are also imported to simulate the adaptive chosen-message attack. A CAS scheme should be secure against the existential forgery under these adaptive adversaries.

TYPE-I Adversary : This type of adversary $\mathcal{A}_I$ can not access the KGC's master-key, but has the ability to replace the public key of any entity, because there are no certificates involved in CL-PKC.

TYPE-II Adversary : This type of adversary $\mathcal{A}_{II}$ can access the KGC's master-key, but he has no ability to replace the public key of any entity.

We notice that $\mathcal{A}_I$ act as a common adaptive forger, while $\mathcal{A}_{II}$ is designed to model the security against a malicious KGC or adversaries who compromised master-key.

According to the different types of the adversary, we define the following games between an adversary $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}\}$ and a challenger $\mathcal{C}$.

1. **Setup:** $\mathcal{C}$ takes a security parameter 1^k and runs the Setup algorithm, publishes the resulting system parameters params, and then
   - **Type-I**: $\mathcal{C}$ keeps master-key to itself. For any user ID, $\mathcal{A}_I$ can request a partial private key of the identifier ID, $\mathcal{C}$ responses D_{ID}. $\mathcal{A}_I$ can select a new secret value x' and compute the corresponding public key $(X'_{\mathsf{ID}}, Y'_{\mathsf{ID}})$. $\mathcal{C}$ will record these replacements $(x', X'_{\mathsf{ID}}, Y'_{\mathsf{ID}})$ as valid.
   - **Type-II**: $\mathcal{C}$ gives master-key to $\mathcal{A}$.
2. **Queries:** For any user ID, $\mathcal{A}$ can submit a query to $\mathcal{C}$ on an arbitrary message m_i. $\mathcal{C}$ returns the signature σ_i which is valid under the user's public key, and records the signed message m_i in the tape M.
3. **Response:** After the above experiments, $\mathcal{A}$ outputs a valid aggregate signature σ', which satisfies $\mathsf{Ver}((pk_1, m_1), \cdots, (pk_n, m_n), \sigma') = true$ and there is at least one message $m_i, i \in \{1, \cdots, n\}$ that is not recorded on $\mathcal{C}$'s tape M.

We define the advantage of an adversary $\mathcal{A}$ wins the above game as

$$\mathbf{Adv}_{CAS}^{Agg-CMA}(\mathcal{A}) = \Pr[\mathsf{Ver}((pk_1, m_1), \cdots, (pk_n, m_n), \sigma') = 1 | \exists m_i \notin M, i \in (1, \cdots, n)].$$

Definition 2. *An adversary is $(t, \epsilon, n, q_H, q_E, q_S)$-breaks an CAS scheme if: there are n individual users; $\mathcal{A}$ runs in time at most t; $\mathcal{A}$ makes at most q_H times hash queries, q_E times partial private key extractions and q_S times signing queries; and $\mathbf{Adv}_{CAS}^{Agg-CMA}(\mathcal{A})$ is at least ϵ.*

Definition 3. *A CAS scheme is $(t, \epsilon, n, q_H, q_E, q_S)$-secure against existential forgery if there is no adversary $(t, \epsilon, n, q_H, q_E, q_S)$-breaks it.*

3 Two Certificateless Aggregate Signatures from Bilinear Maps

Here we propose two certificateless aggregate signature schemes, which are denoted by $\mathcal{CAS}$-1 and $\mathcal{CAS}$-2, respectively. The first scheme $\mathcal{CAS}$-1 reduces the costs of communication and signer-side computation but loses on the storage, while $\mathcal{CAS}$-2 minimizes the storage but sacrifices the communication. Before the detailed description, we give some basic definitions and notions which

will be used in both of the schemes. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two additive cyclic groups with the same prime order q. Let $\hat{e}$ be a bilinear map such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \leftarrow \mathbb{G}_2$. Let n be the maximum number of the users in the schemes, $i \in \{1, 2, \cdots, n\}$.

3.1 $\mathcal{CAS}$-1 Scheme

Setup: KGC generates system parameters params and secret value master-key as follows:

1. randomly selects a generator $P \in \mathbb{G}_1$;
2. chooses a random value $s \in \mathbb{Z}_q^*$ as the master-key, and then computes $Q = sP$;
3. chooses two cryptographic hash functions $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$;
4. publishes the system parameters params $= \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, Q, H_1, H_2\}$.

Set Secret Value: The i-th user chooses a secret random value $x_i \in \mathbb{G}_1$ and save x_i securely.

Set Public Key: The i-th user computes the user's public key $P_i = (X_i, Y_i)$ where $X_i = x_i P$ and $Y_i = x_i Q$. Anyone can check if P_i is valid by the equation

$$\hat{e}(X_i, Q) = \hat{e}(Y_i, P). \tag{1}$$

Partial Private Key Extract: The i-th user sends his identifier $ID_i \in \{0,1\}^*$ and the public key P_i to KGC, KGC constructs the partial private key $D_i = sH_1(ID_i||P_i)$.

Set Full Private Key: When received his partial private key D_i from the KGC, the i-th user computes $S_i = x_i D_i$ as full private key.

Sign: Given an arbitrary message $m_i \in \{0,1\}^*$, the i-th user processes the signing algorithm as follows:

1. selects $r_i \in_R \mathbb{Z}_q^*$, computes $U_i = r_i P$;
2. computes $T_i = H_2(ID_i||m_i||U_i)$;
3. computes $V_i = r_i T_i + S_i$;
4. outputs $\sigma_i = (U_i, V_i)$ as the signature on m_i.

Aggregate: For n individual signatures given by n distinct users, where $n = 1, 2, \cdots$, the aggregation goes:

1. parses σ_i into U_i, V_i;
2. computes $\overline{V} = \sum_{i=1}^{n} V_i$;
3. outputs aggregated signature $\overline{\sigma} = (U_1, U_2, \cdots, U_n, \overline{V})$.

Verify: Given a signature $\overline{\sigma}$, anyone checks if the equation

$$\hat{e}(P, \overline{V}) = \prod_{i=1}^{n} \hat{e}(Y_i, H_1(ID_i||P_i)) \cdot \prod_{i=1}^{n} \hat{e}(U_i, T_i) \tag{2}$$

holds, where $P_i = (X_i, Y_i)$ and $T_i = H_2(ID_i||m_i||U_i)$, and then returns valid or not. The correctness:

$$\begin{aligned} \hat{e}(P, \overline{V}) &= \hat{e}(P, \sum_{i=1}^{n} r_i T_i) \cdot \hat{e}(P, \sum_{i=1}^{n} S_i) \\ &= \hat{e}(Y_i, \sum_{i=1}^{n} H_1(ID_i||P_i)) \cdot \hat{e}(P, \sum_{i=1}^{n} r_i T_i) \\ &= \prod_{i=1}^{n} \hat{e}(Y_i, H_1(ID_i||P_i)) \cdot \prod_{i=1}^{n} \hat{e}(U_i, T_i). \end{aligned} \tag{3}$$

3.2 Security Analysis

In the random oracle model, while assuming the intractability of CDH problem in the groups with bilinear maps, we will prove $\mathcal{CAS}$-1 is existentially unforgeable in the security model of certificatelss aggregate signatures.

Theorem 1. *If there exists an adversary $\mathcal{A}$ can $(t, \epsilon, n, q_{H_1}, q_{H_2}, q_E, q_S)$-breaks $\mathcal{CAS}$-1, then we can construct an algorithm $\mathcal{B}$ can solve CDH problem in the polynomial time bound with a non-negligible probability.*

Proof. Assume that $\mathcal{B}$ is given an instance (q, P, aP, bP) of the CDH problem, and will interact with a Type-I adversary $\mathcal{A}$ as follows to computes abP.

Setup: $\mathcal{B}$ sets the KGC's $\mathsf{params} = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, Q, H_1, H_2\}$, public key $Q = aP$. H_1, H_2 are two random oracles controlled by $\mathcal{B}$.

Hash Queries: $\mathcal{A}$ can make hash query at any time. $\mathcal{B}$ maintaining a list to each random oracle.

For H_1-query on (ID_i, P_i):

1. If (ID_i, P_i) is queried previously, $\mathcal{B}$ retrieves $(k_{i,1}, l_{i,1})$ from H_1-list.
2. Else, with the probability $1 - \frac{1}{q_{H_1}}$, $\mathcal{B}$ generates $k_{i,1} \in_R \mathbb{Z}_q^*$, $l_{i,1} = 0$ and H_1-$coin_i$ = 0; with the probability $\frac{1}{q_{H_1}}$, $\mathcal{B}$ generates $k_{i,1}, l_{i,1} \in_R \mathbb{Z}_q^*$ and H_1-$coin_i$ = 1. $\mathcal{B}$ logs $(ID_i, P_i, H_1\text{-}coin_i, k_{i,1}, l_{i,1})$ in the H_1-list.
3. $\mathcal{B}$ responds with $H_1(ID_i||P_i) = k_{i,1}P + bl_{i,1}P$.

For H_2-query on (ID_i, m_i, U_i):

1. If (ID_i, m_i, U_i) is queried previously, $\mathcal{B}$ retrieves $k_{i,2}$ from H_2-list.
2. Else, $\mathcal{B}$ chooses $k_{i,2} \in_R \mathbb{Z}_q$, then he logs $(ID_i, m_i, U_i, k_{i,2})$ in the H_2-list.
3. $\mathcal{B}$ responds with $H_2(ID_i||m_i||U_i) = k_{i,2}P$.

Partial Private Key Extraction: For $\mathcal{A}$ asks the partial private key for (ID_i, P_i):

1. If (ID_i, P_i) is queried previously, $\mathcal{B}$ retrieves $(H_1\text{-}coin_i, k_{i,1})$ from H_1-list.
2. Else, $\mathcal{B}$ makes H_1-query on (ID_i, P_i).
3. If $H_1\text{-}coin_i = 0$, $\mathcal{B}$ responses $D_i = k_{i,1}Q$. If $H_1\text{-}coin_i = 1$, $\mathcal{B}$ aborts.

Signing Query: While $\mathcal{A}$ requests a signature on (ID_i, P_i, m_i), $\mathcal{B}$ retrieves $H_1\text{-}coin_i$ from H_1-list. If $H_1\text{-}coin_i = 0$, $\mathcal{B}$ processes as below:

1. selects $r_i \in_R \mathbb{Z}_q$, computes $U_i = r_iP$;
2. computes $T_i = H_2(ID_i||m_i||U_i) = k_{i,2}P$;
3. computes $V_i = r_iT_i + S_i'$, $S_i' = x_ik_{i,1}Q = k_{i,1}Y_i$;
4. outputs $\sigma_i = (U_i, V_i)$ as the signature on m_i.

If $H_1\text{-}coin_i = 1$, $\mathcal{B}$ aborts. This is the point that $\mathcal{B}$ uses the forgeability of the adversary $\mathcal{A}$ to solve the CDH problem.

Takes $\mathcal{B}$'s answers to the verifying equation (2), it is easily to prove that the simulation is perfect. If $\mathcal{B}$ does not abort during the interaction, the algorithm is indistinguishable from a legal signer.

Output: After adaptive training, $\mathcal{A}$ forges a valid signature $\sigma_j = (U_j, V_j)$ on the message m_j and identifier ID_i, while m_j never showed in the signature query phase.

If it is not the case that $H_1\text{-}coin_i = 0$, then $\mathcal{B}$ returns failure. Since Type-I adversary can select a new secret value x' and compute the corresponding public

key $(X'_{\mathsf{ID}}, Y'_{\mathsf{ID}})$ for the user ID, we can derive its CDH problem answer bQ from the following equation.

$$\begin{aligned} V_j &= r_j T_j + S_i \\ &= r_j k_{j,2} P + x_i s(k_{i,1} P + b l_{i,1} P) \\ &= r_j k_{j,2} P + k_{i,1} Y_i + x_i l_{i,1} bQ. \end{aligned} \tag{4}$$

It is easily to analyze that $\mathcal{CAS}$-1 is also unforgeable against Type-II adversary under the same assumption. The proof goes similar to Theorem 1 and hence, it is omitted. □

3.3 $\mathcal{CAS}$-2 Scheme

Here we give the description of $\mathcal{CAS}$-2 scheme. Compares with $\mathcal{CAS}$-1, $\mathcal{CAS}$-2 minimizes the storage but sacrifices the communication. We notice that the scheme can be seen as a slight modification version of Gentry and Ramzan's identity-based scheme [8].

Setup: KGC generates system parameters params and secret value $\mathsf{master\text{-}key}$ as follows:

1. randomly selects a generator $P \in \mathbb{G}_1$;
2. chooses a random value $s \in \mathbb{Z}_q^*$ as the $\mathsf{master\text{-}key}$;
3. chooses three cryptographic hash functions $H_1, H_2, H_3 : \{0,1\}^* \to \mathbb{G}_1$;
4. publishes the system parameters $\mathsf{params} = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, Q, H_1, H_2, H_3\}$.

Set Secret Value: The i-th user chooses a secret random value $x_i \in \mathbb{G}_1$ and save x_i securely.

Set Public Key: The i-th user computes the user's public key $P_i = (X_i, Y_i)$ where $X_i = x_i P$ and $Y_i = x_i Q$. Anyone can check if P_i is valid by the equation (1).

Partial Private Key Extract: The i-th user sends his identifier $ID_i \in \{0,1\}^*$ and the public key P_i to the KGC, KGC constructs the partial private key $D_i = (D_{i,1}, D_{i,2})$ for the i-th user, while $D_{i,1} = sH_1(ID_i||X_i)$ and $D_{i,2} = sH_1(ID_i||Y_i)$.

Set Full Private Key: After received his partial private key D_i from the KGC, the i-th user computes $S_{i,1} = x_i D_{i,1}$ and $S_{i,2} = x_i D_{i,2}$, sets $S_i = (S_{i,1}, S_{i,2})$ as full private key.

Sign: Given an arbitrary message $m_i \in \{0,1\}^*$, the first user choose $\alpha \in_R \mathbb{G}_1$, Each subsequent signer checks that α has not used. Alternatively, different signers may arrive at the same α according to a pre-established negotiation. The i-th user processes the signing algorithm as follows:

1. selects $r_i \in_R \mathbb{G}_1$, then computes $P_\alpha = H_2(\alpha)$;
2. computes $c_j = H_3(ID_i, m_j, \alpha)$;
3. computes its signature (α, U_i, V_i), where $U_i = r_i P$ and $V_i = r_i P_\alpha + S_{i,1} + c_j S_{i,2}$.

Aggregate: For n individual signatures given by n distinct users, the aggregation goes:

1. parses σ_i into U_i, V_i;
2. computes $\overline{V} = \sum_{i=1}^{n} V_i$;
3. computes $\overline{U} = \sum_{i=1}^{n} U_i$;
4. outputs aggregated signature $\overline{\sigma} = (\overline{U}, \overline{V})$.

Verify: Given a signature $\overline{\sigma}$, we check if the following equation holds.

$$\hat{e}(P, \overline{V}) = \hat{e}(\overline{U}, P_\alpha) \cdot \prod_{i=1}^{n} \hat{e}(Y_i, H_1(ID_i||X_i) + c_i H_1(ID_i||Y_i)). \quad (5)$$

This ends the descriptions of $\mathcal{CAS}$-2 scheme. The security analysis is similar to those of $\mathcal{CAS}$-1 as stated in Section 3.2 and the proof sketch in [8].

Theorem 2. *If there exists an adversary $\mathcal{A}$ can $(t, \epsilon, n, q_{H_1}, q_{H_2}, q_{H_3}, q_E, q_S)$-breaks $\mathcal{CAS}$-2, then we can construct an algorithm $\mathcal{B}$ can solve CDH problem in the polynomial time bound with a non-negligible probability.*

Proof. Assume that $\mathcal{B}$ is given an instance (q, P, aP, bP) of the CDH problem, and will interact with a Type-I adversary $\mathcal{A}$ as follows to computes abP.

Setup: $\mathcal{B}$ sets the KGC's params $= \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, Q, H_1, H_2\}$, public key $Q = aP$. H_1, H_2 are two random oracles controlled by $\mathcal{B}$.

Hash Queries: $\mathcal{A}$ can make hash query at any time. $\mathcal{B}$ maintaining a list to each random oracle.

For H_1-query on (ID_i, P_i), where $P_i = \{X_i, Y_i\}$:

1. If (ID_i, P_i) is queried previously, $\mathcal{B}$ retrieves $(k_{i,1}, l_{i,1})$ from H_1-list.
2. Else, with the probability $1 - \frac{1}{q_{H_1}}$, $\mathcal{B}$ generates $k_{i,1} \in_R \mathbb{Z}_q^*$, $l_{i,1} = 0$ and H_1-$coin_i$ = 0; with the probability $\frac{1}{q_{H_1}}$, $\mathcal{B}$ generates $k_{i,1}, l_{i,1} \in_R \mathbb{Z}_q^*$ and H_1-$coin_i$ = 1. $\mathcal{B}$ logs $(ID_i, P_i, H_1\text{-}coin_i, k_{i,1}, l_{i,1})$ in the H_1-list.
3. $\mathcal{B}$ responds with $H_1(ID_i||X_i) = k_{i,1}P$ and $H_1(ID_i||Y_i) = bl_{i,1}P$.

For H_2-query on α:

1. If α is queried before, $\mathcal{B}$ retrieves r from the H_2-list.
2. Else, $\mathcal{B}$ chooses $r \in_R \mathbb{G}_1$, then he logs (r, α) in the H_3-list.
3. $\mathcal{B}$ responds with $H_2(\alpha) = rP$.

For H_3-query on (ID_i, m_i, α):

1. If (ID_i, m_i, α) is queried previously, $\mathcal{B}$ retrieves $k_{i,2}$ from H_3-list.
2. Else, $\mathcal{B}$ chooses $k_{i,2} \in_R \mathbb{Z}_q$, then he logs $(ID_i, m_i, \alpha, k_{i,2})$ in the H_3-list.
3. $\mathcal{B}$ responds with $H_3(ID_i, m_i, \alpha) = k_{i,2}P$.

Partial Private Key Extraction: For $\mathcal{A}$ asks the partial private key for (ID_i, P_i):

1. If (ID_i, P_i) is queried previously, $\mathcal{B}$ retrieves $(H_1\text{-}coin_i, k_{i,1}, l_{i,1})$ from H_1-list.
2. Else, $\mathcal{B}$ makes H_1-query on (ID_i, P_i).
3. If H_1-$coin_i = 0$, $\mathcal{B}$ responses $X_i = k_{i,1}Q$ and $Y_i = bl_{i,1}Q$. If H_1-$coin_i = 1$, $\mathcal{B}$ aborts.

Signing Query: While $\mathcal{A}$ requests a signature on (ID_i, P_i, m_i), $\mathcal{B}$ retrieves H_1-$coin_i$ from H_1-list. If H_1-$coin_i = 0$, $\mathcal{B}$ processes as below:

1. selects $r_i \in_R \mathbb{Z}_q$, computes $U_i = r_iP$;
2. computes $P_\alpha = H_2(\alpha)$;
3. computes $c_j = H_3(ID_i, m_j, \alpha)$;
4. computes $V_i = r_iP_\alpha + S'_{i,1} + c_jS'_{i,2}$, where $S'_{i,1} = x_ik_{i,1}Q$ and $S'_{i,2} = bl_{i,1}Y_i$;
5. outputs $\sigma_i = (U_i, V_i)$ as the signature on m_i.

If H_1-$coin_i = 1$, $\mathcal{B}$ aborts. This is the point that $\mathcal{B}$ uses the forgeability of the adversary $\mathcal{A}$ to solve the CDH problem.

Takes $\mathcal{B}$'s answers to the verifying equation (5), it is easily to prove that the simulation is perfect. If $\mathcal{B}$ does not abort during the interaction, the algorithm is undistinguishable from a legal signer.

Output: After adaptive training, $\mathcal{A}$ forges a valid signature $\sigma_j = (U_j, V_j)$ on the message m_j and identifier ID_i, while m_j never showed in the signature query phase.

If it is not the case that $H_1\text{-}coin_i = 0$, then $\mathcal{B}$ returns failure. Since Type-I adversary can select a new secret value x' and compute the corresponding public key $(X'_{\mathsf{ID}}, Y'_{\mathsf{ID}})$ for the user ID, we can derive its CDH problem answer bQ from the following equation.

$$\begin{aligned} V_j &= r_j P_\alpha + S'_{i,1} + c_j S'_{i,2} \\ &= rr_j P + x_i k_{i,1} Q + c_j b l_{i,1} Y_i \\ &= rr_j P + k_{i,1} Y_i + x_i c_j l_{i,1} bQ. \end{aligned} \tag{6}$$

It is easily to derive that $\mathcal{CAS}$-2 is also unforgeable against Type-II adversary under the same assumption. The proof goes similar to Theorem 2 and hence, it is omitted. □

4 Performance Comparison

Here we give a performance comparison amongst some related schemes and ours. In order to show the trade-off for the schemes' certificateless and the trust level 3, we choose a PKI-based scheme [4] and an ID-based scheme [8]. Moreover, we also select a general certificateless signature scheme [1] to show the advantages of aggregation. T_p denotes time for one pairing operation in the elliptic curve groups. T_e denotes time for one exponential operation. n is the number of the individual signatures. ℓ denotes the length of a group element.

Table 2. The comparison of the aggregate signature schemes

	Boneh03[4]	Gentry06[8]	Al-Riyami03[1]	$\mathcal{CAS}$-1	$\mathcal{CAS}$-2
Sign Costs	nT_e	$3nT_e$	$nT_p + 3nT_e$	$2nT_e$	$3nT_e$
Verify Costs	$(n+1)T_p$	$nT_e + 3T_p$	$2nT_p + nT_e$	$(2n+1)T_p$	$nT_e + (n+2)T_p$
Aggregate Length	1ℓ	3ℓ	$2n\ell$	$(n+1)\ell$	2ℓ
Certificate	Need	Not Need	Not Need	Not Need	Not Need
Trust Level	3	1	3	3	3

From Table 2, we can understand that both $\mathcal{CAS}$-1 and $\mathcal{CAS}$-2 pay more computation costs for realizing certificateless and the trust level 3 simultaneously. The trade-off is valuable since an authority in low trust level is unacceptable in some implementations, e.g., military networks. Because $\mathcal{CAS}$-1 is less

computation in signing process, so it is feasible for the environments where the signer side is limited computational ability. Thus $\mathcal{CAS}$-2 is better for the limited storage applications since its aggregate signature length is a const value.

5 Conclusion

In this paper, two practical certificateless aggregate signature schemes are proposed. One can adaptively choose one of the above schemes by the consideration of which advantage is the most important in practice. Both of the schemes are proven secure in the random oracle model (ROM) by assuming the intractability of the computational Diffie-Hellman (CDH) problem over groups with bilinear maps, without using the forking lemma technique. An interesting open problem is to design such a scheme based on the CL-PKC that neither the storage nor the computation costs is linearly increased with the number of the signing messages or the involving parties.

References

1. S. S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. In *Advances in Cryptography-Asiacrypt 2003*, LNCS 2894, pp. 452-473, 2003.
2. K.C. Barr and K. Asanovic. Enery aware lossless data compression. In *Proc. of Mobisys 2005*, 2005.
3. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *SIAM J. of Computing*, 32(3):586-615, 2003.
4. D.Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and Verifiably Encrypted Signatures from Blinear Maps. In E. Biham, editor, *Advances in Cryptology-EUROCRYPT 2003*, LNCS 2656, pp. 416-432, 2003.
5. D. Boneh, C. Gentry and H. Shacham. A Survey of two signature aggregation techiques. *RSA's CryptoBytes*, 6(2), Summer 2003.
6. D. Boneh, B. Lynn and H. Shacham. Short signatures from Weil Pairing. In C. Boyd, editor, *Advances in Cryptology-ASIACRYPT 2001*, LNCS 2248, pp. 514-532, 2001.
7. X.G. Cheng, J.M. Liu, and X.M. Wang. Identity-Based Aggregate and Verifiably Encrypted Signatures from Bilinear Pairing O. Gervasi et al. (Eds.): *ICCSA 2005*, LNCS 3483, pp. 1046-1054, 2005.
8. C. Gentry and Z. Ramzan. Identity-Based Aggregate Signatures. Yung et al. (Eds.): *PKC 2006*, LNCS 3958, pp. 257-273, 2006.
9. M. Girault. Self-certified public keys. D.W. Davies. (EDs.): *Proc. EUROCRYPT 1991*, LNCS 547, pp. 490C497, 1992.
10. Z. Gong, Y. Long, X. Hong and K.F. Chen. "Two Certificateless Aggregate Signatures from Bilinear Maps". SNPD 2007, IEEE Computer Society Proceedings, pp. 188-193. August 2007.
11. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proceedings of ACM ASPLOS IX*, pp. 93-104, November 2000.
12. X. Huang, W. Susilo, Y. Mu and F. Zhang. On the Security of Certificateless Signature Schemes from Asiacrypt 2003. *CANS 2005*, LNCS 3810, pp. 13-25, 2005.

13. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *J. Cryptoloy*, 13:361-396, 2000.
14. A.Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology-Crypto'84*, LNCS 196, pp. 47-53, 1985.
15. Z.H. Shao. Enhanced Aggregate Signatures from Pairings. D. Feng, D. Lin, and M. Yung (Eds.): *CISC 2005*, LNCS 3822, pp. 140-149, 2005.
16. J. Xu, Z.F. Zhang and D.G. Feng. ID-Based Aggregate Signatures from Bilinear Pairings. Y.G. Desmedt et al.(Eds.): *CANS 2005*, LNCS 3810, pp. 110-119, 2005.
17. D.H. Yum and P.J. Lee. Generic Construction of Certificateless Signature. *Information Security and Privacy, ACISP 2004*, LNCS 3108, pp. 200-211, 2004.