Efficient Certificateless Aggregate Signature Scheme

Yu-Chi Chen, Gwoboa Horng, Chao-Liang Liu, Yuan-Yu Tsai, and Chi-Shiang Chan

Abstract—**In ubiquitous computing, data should be able to be accessed from any location, and the correctness of data becomes vital during the communication. Suppose that many users sign different messages respectively, before forwarding or sending these messages, then the verifier must spend a lot of computing time to verify their signatures. Consequently, the aggregate signature scheme is an effective method of improving efficiency in this kind of systems, which provides the convenience for the verifier. In this paper, we propose a new certificateless aggregate signature scheme which is efficient in generating a signature and verification. This scheme is provably secure under the extended computational Diffie-Hellman assumption.**

Index Terms—**Aggregate signature, certificateless aggregate signature, certificateless cryptography, digital signature.**

1. Introduction

Digital signatures is a very important application in the public key cryptosystem (PKC), which can provide benefits of authenticity, integrity, and non-repudiation. A traditional digital signature scheme consists of a signer and a verifier, where a signer gives the signature. However, different kinds of signature schemes are proposed for different applications, e.g., the proxy signature and blind signature[1], [2]. However, there is a problem when many users sign different messages, respectively, while the verifier will need lots of computing time for the verification. The aggregate signature scheme is proposed to deal with this kind of problems[3], [4]. In other words, when it is required for *n* users to generate individual *n* signatures, the requests will be delivered to the aggregate signature generator (ASG). Upon receiving the *n* signatures, the ASG must combine those signatures and generate an aggregate signature to the verifier.

Comparing with the traditional PKC, the certificateless public key cryptosystem (CL-PKC)[5], [6] does not need the extra trusted party to manage certificates. Differing from the ID-based public key cryptosystem (ID-PKC)[7]–[12], the key generation center (KGC) in CL-PKC is unable to hold the user's whole private key, which does not have the highest priority for key generation. CL-PKC practically solves the key escrow problem of ID-PKC and is widely applied in the certificateless signature scheme. Since the KGC in certificateless aggregate signature schemes (CLAS) does not hold users' private keys, a malicious KGC cannot compute the valid aggregate signature. The technique of CLAS is not only used to easily combine signatures into an aggregate signature, but is also applied to save the computational cost for the verification.

For example, the organization of a commercial company is the most common application with CLAS. The users, i.e., employees, sign the individual messages and send them to the secretary. The secretary as ASG will compute the aggregate signature and give it to the corresponding manager. The manager, playing the role of a verifier, can directly verify whether the aggregate signature is valid or not. In this paper, we propose an efficient certificateless aggregate signature scheme, which is provably secure in the random oracle model under the extended computational Diffie-Hellman assumptions.

The rest of the paper is organized as follows. In Section 2, we briefly review the preliminaries of CLAS. The proposed CLAS scheme and its security analysis are presented in Section 3, while performance comparisons with previous schemes[13], [14] are provided in Section 4. Finally, in Section 5, we conclude this paper.

2. Preliminary of CLAS

The notations of CLAS are defined in Section 2.1. We briefly present a bilinear pairing and the construction of CLAS in Sections 2.2 and 2.3, respectively, while the security model and hardness assumptions are discussed in Sections 2.4 and 2.5, respectively.

2.1 Notations of CLAS

The notations of CLAS used in this paper are shown in Table 1. The flowchart of aggregate signature is shown in Fig. 1.

Manuscript received May 23, 2012; revised July 23, 2012. This work was supported by the National Science Council under Grant No. NSC 100-2221-E-005-062 and NSC 100-2221-E-468-014.

Y.-C. Chen and G. Horng are with the Department of Computer Science and Engineering, National Chung-Hsing University, Taichung (e-mail: s9756034@cs.nchu.edu.tw; gbhorng@cs.nchu.edu.tw).

C.-L. Liu, Y.-Y. Tsai, and C.-S. Chan are with the Department of Applied Informatics and Multimedia, Asia University, Taichung (e-mail: jliu@asia.edu.tw; yytsai@asia.edu.tw; cschan@asia.edu.tw).

Digital Object Identifier: 10.3969/j.issn.1674-862X.2012.03.004

Table 1: Notations used in CLAS

Notation	Meaning
CLAS	Certificateless aggregate signature
KGC	Key generation center
ASG	Aggregate signature generator
ID_i	The ith user's identity
X_i	The ith user's public key
S_i	The ith user's partial private key
l	A security parameter
$\hat{e}$	A bilinear pairing
M_i	A message from the i-th user
M	The message space
σ_i	A signature from the i-th user
σ	An aggregate signature
$\varDelta$	A state information
H	A hash function
$\perp$	Denoting that the value is empty

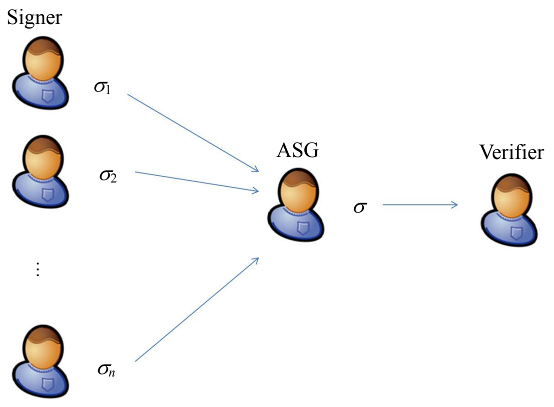

Fig. 1. Flowchart of CLAS.

2.2 Bilinear Pairing

Bilinear pairing is a mapping function. Let G_1 be an additive group of points of an elliptic curve, and G_2 be a multiplicative group in the finite field. G_1 and G_2 are cyclic groups with the same prime order q. The bilinear pairing is denoted as $\hat{e}: G_1 \times G_1 \rightarrow G_2$, and it has the following properties.

1) Computable: given P, $Q \in G_1$, then there exists a polynomial time algorithm to compute $\hat{e}(P, Q) \in G_2$.

2) Bilinear: for any x, $y \in Zq^*$ we have $\hat{e}(xP, yP) = \hat{e}(P, P)^{x,y}$ for any $P \in G_1$.

3) Non-degenerate: if P is a generator of G_1, then $\hat{e}(P, P) \neq 1$ in which $\hat{e}(P, P)$ is also the generator of G_2.

Moreover, there are many articles which present approaches to improve the efficiency of pairing computation, such as [14] to [16]. In the following, the hardness assumptions are presented based on bilinear pairings.

1) Computational Diffie-Hellman (CDH) problem. Given (P, aP, bP) then compute abP, where $P \in G_1$ is the generator and a, $b \in Zq^*$ is unknown.

2) Extended computational Diffie-Hellman (eCDH) problem. Given $(P, a_1P, b_1P, a_2P, b_2P)$ then compute $a_1b_1P + a_2b_2P$, where $P \in G_1$ is a generator and a_1, b_1, a_2, and $b_2 \in Zq^*$ is unknown. The eCDH problem can be reduced to two CDH problems, i.e. $(P, a_1P, b_1P, a_2P, b_2P)$ is divided to (P, a_1P, b_1P) and (P, a_2P, b_2P).

2.3 Construction of CLAS

The components of CLAS are the KGC, n users $(U_1, U_2, \cdots, U_n)$, an ASG, and a verifier, while CLAS scheme consists of the following algorithms:

1) Setup: This algorithm, run by KGC, takes the security parameter l as input and generates the master-key s and system parameter *params*.

2) Partial-private-key: KGC runs the algorithm and inputs *params*, master-key s, and the user's identity ID_i. This algorithm computes the user's partial private key S_i.

3) UserKeyGen: This algorithm performed by the user takes ID_i as input, then outputs the user's private key (x_i, S_i), and public key X_i. Hence, we note that the publication of use's public keys does not need the certificates in CL-KPC.

4) Sign: This algorithm performed by the user inputs the message $M_i \in M$, user's identity ID_i, private keys (x_i, S_i), public key X_i, and state information $\varDelta$. It returns a signature σ_i as the output.

5) Aggregate: The ASG receives the n signatures $\sigma_1, \sigma_2, \cdots, \sigma_n$ from n users and runs this algorithm to obtain the aggregate signature σ.

6) Verify: The algorithm performed by the verifier takes the aggregate signature σ, corresponding $\mathrm{ID}_1, \mathrm{ID}_2, \cdots, \mathrm{ID}_n$, public keys $X_1, X_2, \cdots, X_n$, messages $M_1, M_2, \cdots, M_n$, and the state information $\varDelta$ as inputs. As a result, it returns 'true' if the aggregate signature is valid; otherwise, returns 'false'.

Note that all the users with $\mathrm{ID}_1, \mathrm{ID}_2, \cdots, \mathrm{ID}_n$ in an aggregate set apply the same state information $\varDelta$, see [14] for more details.

2.4 Security Model of CLAS

There are two types of adversaries in CL-PKC[14], namely Type 1 and Type 2 adversaries. Assume that Type 1 adversary A_1 being the attacker with no access to the master-key, A_1 can replace the public key of any chosen entity. On the other hand, Type 2 adversary A_2 playing as the KGC is able to access the master-key, but it does not demand to replace the public key.

The security model proposed by Zhang *et al.*[14] uses Game 1 and Game 2 to simulate Type 1 and Type 2 adversaries. In this paper, we combine Game 1 and Game 2 to form a new Game 3, in order to simulate a Type 3 adversary. The Type 3 adversary A_3 does not have the ability to access the master-key, and it also does not perform the public key replacement. In the rest of this paper, by applying Game 3 to imply Game 1 and Game 2, we will show the result of security analysis of the proposed scheme.

The security model of the proposed CLAS scheme is based on the following game and interaction between a challenger B and an adversary A_3.

Game 3 (for Type 3 adversary)

Setup: B performs the defined Setup algorithm in Section 2.3 and inputs a security parameter. This algorithm returns the master-key and system parameter *params*, while B sends *params* to A_3 and keeps the master-key secret.

Attack: A_3 can adaptively perform the following polynomially bounded queries.

1) Partial-private-key queries: A_3 can obtain the partial private key of any user whose identity is ID_i, and B returns the partial private key S_i to A_3.

2) Public-key queries: A_3 is able to acquire the public key of any user with ID_i, and B outputs the public key X_i.

3) Secret-value queries: A_3 can query the secret value of any user with ID_i, and B returns the secret value x_i to A_3.

4) Sign queries: A_3 is able to send user's identity ID_i, message M_i, and state information Δ_i to request a signature. B generates a signature σ_i which contains (ID_i, M_i, Δ_i) and the corresponding public key X_i, then delivers σ_i to A.

Forgery: A_3 outputs an aggregate signature σ^* for a set of n users with corresponding identities $\{\mathrm{ID}_1^*, \mathrm{ID}_2^*, \cdots, \mathrm{ID}_n^*\}$, public keys $\{X_1^*, X_2^*, \cdots, X_n^*\}$, n messages form n users $\{M_1^*, M_2^*, \cdots, M_n^*\}$, and the corresponding forged state information Δ^*.

Furthermore, we assume that adversary A_3 wins Game 3 if and only if A_3 can successfully obtain a valid forged aggregate signature.

Definition 1. The aggregate signature forgery is defined as follows. Assume that an attacker has obtained an aggregated signature from $(n-1)$ users with identities ID_1, ID_2, $\cdots$, ID_{n-1} , then the whole n-user aggregate signature can be forged when the signature of ID_n is never obtained.

3. Novel and Efficient CLAS Scheme

3.1 Proposed CLAS Scheme

The proposed CLAS scheme consists of six randomized algorithms as follows.

1) Setup (l): KGC inputs the security parameter l. This algorithm chooses two groups G_1, G_2 of prime order q and a bilinear pairing $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$. G_1 is a cyclic additive group with a generator P, and G_2 is a cyclic multiplicative group. The algorithm randomly chooses the master-key $s \in Zq^*$ and computes a public master-key $P_{\mathrm{pub}}=sP$, and chooses three hash functions H_1:$\{0, 1\}^* \rightarrow G_1$, H_2:$\{0, 1\}^* \rightarrow G_1$, H_3:$\{0, 1\}^* \rightarrow Zq^*$. The system parameter is *params*=(G_1, G_2, $\hat{e}$, P, P_{pub}, H_1, H_2, H_3).

2) Partial-private-key (*params*, s, ID_i): This algorithm takes *params*, s, and the user's identity ID_i as inputs to compute $Q_i=H_1(\mathrm{ID}_i)$ and the user's partial private key $S_i=sQ_i$.

3) UserKeyGen (ID_i): This algorithm takes ID_i, then randomly chooses $x_i \in Zq^*$ and outputs the user's complete private key (x_i, S_i) and public key $X_i = x_iP$.

4) Sign (M_i, ID_i, x_i, S_i, X_i, Δ): The user, as the signer, signs the message $M_i \in M$ by using the private key (x_i, S_i), public key X_i, and state information Δ, while detailed steps are described as follows:

a. Select a value $r_i \in Zq^*$ randomly, then compute $D = H_2(\Delta)$ and $R_i=r_iD$;

b. Compute $w_i=H_3(\Delta \parallel M_i \parallel \mathrm{ID}_i \parallel X_i \parallel R_i)$;

c. Compute $V_i = S_i+r_iw_ix_iD$;

d. Return the signature $\sigma_i=(R_i, V_i)$ on M_i.

5) Aggregate(σ_1, σ_2, $\cdots$, σ_n): The ASG receives the n signatures σ_1, σ_2, $\cdots$, σ_n from the n users with identities ID_1, ID_2, $\cdots$, ID_n and runs this algorithm. The algorithm computes $V=\sum_{i=1}^{n} V_i$ and returns the aggregate signature σ=(R_1, R_2, $\cdots$, R_n, V).

6) Verify (σ, L_{ID}, L_X, L_M, Δ): Let $L_{\mathrm{ID}}=\{\mathrm{ID}_1, \mathrm{ID}_2, \cdots, \mathrm{ID}_n\}$, $L_X=\{X_1, X_2, \cdots, X_n\}$, and $L_M=\{M_1, M_2, \cdots, M_n\}$. This algorithm run by the verifier takes the aggregate signature σ, corresponding identities ID_1, ID_2, $\cdots$, ID_n, public keys X_1, X_2, $\cdots$, X_n, messages M_1, M_2, $\cdots$, M_n, and the state information Δ as input. It performs as follows:

(1) Compute $w_i = H_3(\Delta \parallel M_i \parallel \mathrm{ID}_i \parallel X_i \parallel R_i)$ and $Q_i = H_1(\mathrm{ID}_i)$ for all i where $1 \le i \le n$;

(2) Verify the following equation holds or not.

$$\hat{e}(V, P)=\hat{e}(P_{\mathrm{pub}}, \textstyle\sum_{i=1}^{n} Q_i)\prod_{i=1}^{n}\hat{e}(w_iR_i, X_i).$$

It returns 'true' if the equation holds and the verifier accepts the aggregate signature σ; otherwise, returns 'false'.

3.2 Security analysis

According to the eCDH problem, the proposed CLAS scheme is provably secure in the random oracle model[17]. Hereafter, A denotes A_3 in Game 3 as defined in Section 2.4.

Theorem 1. If there exists an adversary A with probability ε, which wins Game 3 to successfully obtain the forged aggregate signature for the proposed scheme in the random oracle model, there exists another adversary B with probability ε' solving the eCDH problem where $\varepsilon' \ge \varepsilon \big/ q_{H_1} e^3$ at most $q_{H_1}H_1$ queries.

Proof. B is given (P, a_1P, b_1P, a_2P, b_2P) as the eCDH problem in G_1. A interacts B via Game 3 as following queries. Eventually, B may base on A's forgery to solve the eCDH problem.

Setup: B sets $P_{\mathrm{pub}}=a_1P$ and *params*=(G_1, G_2, $\hat{e}$, P, P_{pub},

H_1, H_2, H_3), then gives *params* to A.

Attack: A can adaptively perform the following polynomially bounded queries.

H_1 queries: B holds a H_1-list $\langle \mathrm{ID}_j,\ \alpha_j,\ Q_j,\ c_j \rangle$, where H_1-list initially is empty. A can query any user's identity ID_i. B responds as follows.

1) If ID_i has been queried, B returns Q_i from H_1-list;

2) If ID_i is never queried before. B generates a random value $\alpha_i \in Zq^*$ and a random coin $c_i \in \{0, 1\}$, where $\Pr[c_i=0]=1/q_{H_1}$ and $\Pr[\cdot]$ is the probability function;

3) If $c_i=0$, B returns $Q_i=\alpha_i(b_1P)$ to A and adds $\langle \mathrm{ID}_i,\ \alpha_i,\ Q_i,\ c_i \rangle$ into H_1-list;

4) Otherwise, B returns $Q_i=\alpha_iP$ to A and adds $\langle \mathrm{ID}_i,\ \alpha_i,\ Q_i,\ c_i \rangle$ into H_1-list.

H_2, H_3 queries: Since the inputs of H_2, H_3 contain the state information, B holds a H_2, H_3-list $\langle \Delta_j, M_j, \mathrm{ID}_j,\ X_j,\ R_i,\ \beta_j,\ W_j,\ c_j \rangle$, while H_2, H_3-list initially is empty. A can query (Δ_i, M_i , ID_i , X_i , R_i). B responds as follows:.

1) If (Δ_i , M_i , ID_i , X_i , R_i) has been queried, B returns W_i from H_2, H_3-list;

2) If (Δ_i , M_i , ID_i , X_i , R_i) is never queried before, based on c_i in H_1-list, B generates a random value $\beta_i \in Zq^*$;

3) If $c_i=0$, B returns $W_i=\beta_i(b_2P)$ to A and adds $\langle \Delta_i, M_i, \mathrm{ID}_i,\ X_i,\ R_i,\ \beta_i,\ W_i,\ c_i \rangle$ into H_2, H_3-list;

4) If $c_i=1$, B returns $W_i=\beta_iP$ to A and adds $\langle \Delta, M_i, \mathrm{ID}_i,\ X_i,\ R_i,\ \beta_i,\ W_i,\ c_i \rangle$ into H_2, H_3-list.

Partial-private-key queries: B holds a K-list $\langle \mathrm{ID}_j,\ x_j,\ S_j,\ X_j \rangle$, while K-list initially is empty. A can query ID_i. If ID_i has been queried for the partial-private-key before, B returns S_i from K-list. Otherwise, B goes back to perform H_1 query, then responds 'partial-private-key queries' as follows.

1) If $c_i=0$, B aborts and terminates;

2) If $c_i=1$ and S_i exists in K-list, B returns S_i;

3) If $c_i=1$ and S_i does not exist in K-list, B returns $S_i = \alpha_iP_{\mathrm{pub}}$ and adds S_i into K-list.

Public-key queries: A must query ID_i to acquire the public key. If ID_i has been queried before, B returns X_i from K-list. Otherwise, B goes back to perform H_1 query, then responds 'public-key queries' as follows.

1) If $c_i=0$, B sets $X_i = x_ia_2P$ and adds X_i into K-list;

2) If $c_i=1$ and X_i exists in K-list, B returns X_i;

3) If $c_i=1$ and X_i, x_i does not exist in K-list, B chooses a random value $x_i \in Zq^*$ and sets $X_i=x_iP$, then returns X_i and adds X_i into K-list.

Secret-value queries: A can query ID_i for the private key. B returns x_i from K-list. Note, it is possibly that x_i is $\perp$.

Sign queries: receiving a sign query (Δ_i , M_i , ID_i , X_i) from A, B must perform the following steps based on H_1-list, H_2-list, H_3-list, and K-list.

1) If ID_i corresponds to $c_i = 0$, B aborts and terminates.

2) Otherwise, B picks a value $\gamma_i=\alpha_i\beta_i^{-1}$, returns $\sigma_i = (R_i, V_i)$, where $R_i=\gamma_i P$ and $V_i = \alpha_iP_{\mathrm{pub}}+\gamma_i\beta_i X_i$.

3) If $c_i = 1$ and X_i and x_i do not exist in K-list, B chooses a random value $x_i \in Zq^*$ and sets $X_i=x_iP$, then returns X_i and adds X_i into K-list.

Forgery: A outputs a set of n users with identities ID_1^*, ID_2^*, $\cdots$, ID_n^*, corresponding public keys X_1^*, X_2^*, $\cdots$, X_n^*, n messages M_1^*, M_2^*, $\cdots$, M_n^*, and a state information Δ^*. Without loss of generality, we assume that $i = 1$, where ID_1^* has not been performed the sign queries. A outputs an aggregate signature $\sigma^*=(V^*, R_1^*, R_2^*, \cdots, R_n^*)$. This forged aggregate signature must satisfy the following equation:

$$\hat{e}(V^*,\ P) = \hat{e}(P_{\mathrm{pub}},\ Q_i^*)\prod_{i=1}^{n}\hat{e}(w_i^*R_i^*,\ X_i^*)\ .$$

If ID_1^* corresponds to $c_1 \neq 0$, B aborts and terminates. If A successfully forges a valid aggregate signature, B must be capable of computing

$$a_1b_1P + a_2b_2P = (V^* - V')\alpha^* - 1$$

where $V' = \sum_{i=2}^{n}\alpha_iP_{\mathrm{pub}} + \sum_{i=2}^{n}\gamma_i\beta_iX_i$.

We show the formal proof that B solves the eCDH problem with probability $\varepsilon' \geq \varepsilon/qH_1e^3$. There are two defined events, as follows:

1) ε_1: B does not abort in any A's query.

2) ε_2: B does not abort in 'forgery'.

B succeeds when all the above events occur.

Claim 1. The probability that B does not abort in any A's query at least probability $1/e^3$.

Proof. Since $\Pr[c_i=0]=1/q_{H_1}$ in 'sign', 'secret-value', and 'partial-private-key query', B does not abort with probability $\left(1-1/q_{H_1}\right)^{q_{H_1}} \geq 1/e$ then $\Pr[\varepsilon_1] \geq 1/e^3$. Here, e is the natural logarithm.

Claim 2. B does not abort to respond in 'forgery' with the probability $1/q_{H_1}$.

Proof. A outputs an aggregate signature forgery, where without loss of generality, $c_1^*=1$ and $c_i^*= 0$ for all i, $2 \geq i \geq n$. B does not abort with $\Pr[c_1=0]=1/q_{H_1}$, then $\Pr[\varepsilon_2]=1/q_{H_1}$.

Eventually, B does not abort with probability $\varepsilon' \geq \varepsilon/e^3q_{H_1}$ where ε is denoted by A to win Game 3. Hence, B solving eCDH problem is equivalent to A wining Game 3.

Therefore, we further argue that Game 3 implies the Game 1 and Game 2 in the following Theorem.

Theorem 2. According to the result of our previous analysis, if there exists an adversary A_3 which has the ability to win the Game 3, there are A_1 and A_2 which can win the Game 1 and Game 2, respectively.

Proof. The eCDH problem is equal to two CDH problems forming the result of the security analysis $a_1b_1P + a_2b_2P$. If we omit the second part (a_2b_2P) in which A_1 is able to perform the public key replacement, then (a_1b_1P) is the CDH problem. On the other hand, if we omit the first part (a_1b_1P) in which A_2 has the ability to access the master-key, then (a_2b_2P) is also the CDH problem. Therefore, our scheme in the security model of the Game 1 and Game 2 is provably secure.

4. Performance Comparisons

The proposed scheme is compared with two schemes proposed by Gong *et al.*[13] and Zhang *et al.*[14] in Table 2. The relative operations are defined as follows. pr denotes pairing, S as scalar multiplication in G_1, and mp represents the MapToPoint hash function. 'Sign' is the computational cost per signer, and 'verify' is the computational cost of the verification. |AS| and |PK| are the aggregate signature length and the public key length, respectively. L denotes the bit length of a point over the elliptic curve such as 160 bits. The pre-computed cost is omitted in comparison. Gong *et al.*'s scheme 1 and scheme 2 are represented by GLHC-1 and GLHC-2. Zhang *et al.*'s scheme is represented by ZZ.

We can easily show that in Sign the proposed scheme is more efficient than GLHC-2, as well as when comparing to ZZ and GLHC-1 in the verification. In addition, even though GLHC-2 needs less space for the aggregate signature than our scheme, ours has a similar length as ZZ, and requests less space for the public key than the ones for GLHC-1 and GLHC-2. Our scheme is more suitable for a signer with weak devices without using the MapToPoint hash function.

Table 2: Comparison of efficiency and other attributes

	GLHC-1	GLHC-2	ZZ	Proposed
|AS|	$(n+1)L$	$2L$	$(n+1)L$	$(n+1)L$
|PK|	$2L$	$2L$	L	L
Sign	$2S$+1mp	$3S$+1mp	$2S$+1mp	$2S$
Verify	$(2n+1)$pr	$(n+2)$pr	$(n+3)$pr	$(n+2)$pr

5. Conclusions

Recently, certificateless public key cryptography is a popular trend instead of ID-based public key cryptography. The certificateless aggregate signature is a noteworthy technology for wireless sensor networks, which combines the recent concepts of certificateless signature and aggregate signature. In this paper, we propose a new certificateless aggregate signature scheme. This CLAS scheme is provably secure in the random oracle model assuming that the eCDH problem is intractable. In terms of 'sign' and 'verify', our scheme effectively improves the efficiency, and it is more suitable for a signer with weak devices. However, how to reduce the communication cost without raising computation cost will be our interesting future work.

References

[1] Y. F. Chang and C. C. Chang, "Robust t-out-of-n proxy signature based on RSA cryptosystems," *Int. Journal of Innovative Computing, Information and Control*, vol. 4, no. 2, pp. 425–431, 2008.

[2] C. I. Fan, C. I. Wang, and W. Z. Sun, "Fast randomization schemes for Chaum blind signatures," *Int. Journal of Innovative Computing, Information and Control*, vol. 5, no. 11(A), pp. 3887–3900, 2009.

[3] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and veriably encrypted signatures from bilinear maps," *Lecture Notes in Computer Science*, vol. 2656, pp. 641, 2003, doi: 10.1007/3-540-39200-9_26.

[4] Z. Shao, "Fair exchange protocol of signatures based on aggregate signatures," *Computer Communications*, vol. 31, no. 10, pp. 1961–1969, 2008.

[5] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," *Lecture Notes in Computer Science*, vol. 2894, pp. 452–473, 2003, doi: 10.1007/978-3-540-40061-5_29.

[6] Z. Zhang, D. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: security model and efficient construction," *Lecture Notes in Computer Science*, vol. 3989, pp. 293–308, 2006, doi: 10.1007/11767480_20.

[7] Y. C. Chen, C. L. Liu, G. Horng, and K. C. Chen, "A provably secure certificateless proxy signature scheme," *Int. Journal of Innovative Computing, Information and Control*, vol. 7, no. 9, pp. 5557–5510, 2011.

[8] A. Shamir, "Identity based cryptosystems and signature schemes," *Lecture Notes in Computer Science*, vol. 196, pp. 47–53, 1985, doi: 10.1007/3-540-39568-7_5.

[9] T. Wu and R. Su, "ID-based group-oriented cryptosystem and its digital signature scheme," *Computer Communications*, vol. 21, no. 11, pp. 1961–1969, 1997.

[10] J. Xu, Z. Zhang, and D. Feng, "ID-based aggregate signatures from bilinear pairings," *Lecture Notes in Computer Science*, vol. 3810, pp. 110–119, 2005, doi: 10.1007/11599371_10.

[11] X. Cheng, J. Liu, L. Guo, and X. Wan, "Identity-based multisignature and aggregate signature schemes from m-torsion groups," *Journal of Electronics (China)*, vol. 23, no. 4, pp. 569–573, 2006.

[12] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in *Public Key Cryptography*, M. Yung, *et al.* Eds., New York: Springer-Verlag, 2006, pp. 257–273.

[13] Z. Gong, Y. Long, X. Hong, and K. Chen, "Two certificateless aggregate signatures from bilinear maps," in *Proc. of the Eighth ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Qingdao, 2007, pp. 188–193.

[14] L. Zhang and F. Zhang, "A new certificateless aggregate signature scheme," *Computer Communications*, vol. 32, no. 6 , pp. 1079–1085, 2009.

[15] C. L. Liu, G. Horng, and T. Y. Chen, "Further refinement of pairing computation based on Miller's algorithm," *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 395–409, 2007.

[16] D. P. Le and C. L. Liu, "Refinements of Miller's algorithm over weierstrass curves revisited," *Computer Journal*, vol. 54, no. 10, pp. 1582–1591, 2011.

[17] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proc. of the 1st ACM Conf. on Computer and Communications Security*, New York, 1993, pp. 62–73.

Yu-Chi Chen received the B.S. and M.S. degrees from the Department of Computer Science and Engineering, National Chung-Hsing University in 2008 and 2009, respectively. He is currently pursuing his Ph.D. degree with the Department of Computer Science and Engineering, National Chung-Hsing University. His research interests include cryptography and information security.

Gwoboa Horng received the B.S. degree in electrical engineering from National Taiwan University in 1981 and the M.S. and Ph.D. degrees from University of Southern California in 1987 and 1992 respectively, all in computer science. Since 1992, he has been with the faculty of the Department of Computer Science and Engineering, National Chung-Hsing University, Taichung. His current research interests include artificial intelligence, cryptography, and information security.

Chao-Liang Liu was born in Hsinchu in 1966. He completed his B.S. degree in mathematics from National Cheng-Kung University in 1989, and the Ph.D. degree from Institute of Computer Science, National Chung Hsing University in 2007. He is currently an assistant professor with the Department of Applied Informatics and Multimedia, Asia University. His research interests include information security, cryptography, elliptic curve cryptosystem, and pairing computation.

Yuan-Yu Tsai was born in Taichung in 1978. He received the B.S. degree from the Department of Computer Science and Information Engineering from National Central University in 2000, and the Ph.D. degree from Institute of Computer Science, National Chung Hsing University in 2006. He is currently an assistant professor with the Department of Applied Informatics and Multimedia, Asia University. His research interests include computer graphics and information hiding algorithms for three-dimensional models and images. He is a member of the ACM and the IEEE Computer Society.

Chi-Shiang Chan was born in Taiwan in 1975. He received the B.S. degree in computer science in 1999 from the National Cheng-Chi University and the M.S. degree in computer science and information engineering in 2001 from the National Chung Cheng University. He received his Ph.D. degree in computer engineering in 2005 also from the National Chung Cheng University. From 2007 to 2010, he worked as an assistant professor with the Department of Information Science and Applications, Asia University. He is currently an associate professor with the Department of Applied Informatics and Multimedia, Asia University. His research interests include image and signal processing, image compression, information hiding, and data engineering.