Wuhan University Journal of Natural Sciences
2017, Vol.22 No.2, 165-170

Article ID 1007-1202(2017)02-0165-06
DOI 10.1007/s11859-017-1230-9

An Efficient Certificateless Aggregate Signature Scheme

□ **KANG Baoyuan, WANG Mu , JING Dongya**
School of Computer Science and Software, Tianjin Polytechnic University, Tianjin 300387, China

Abstract: Aggregate signature can aggregate n signatures on n messages from n signers into a single signature that convinces any verifier that n signers sign the n messages, respectively. In this paper, by combining certificateless public key cryptography and aggregate signatures, we propose an efficient certificateless aggregate signature scheme and prove its security. The new scheme is proved secure against the two representative types adversaries in certificateless aggregate signature under the assumption that computational Diffie-Hellman problem is hard. Furthermore, from the comparison of the computation cost of the new scheme with some existing certificateless aggregate signature schemes in group sum computation, scalar multiplication computation, Hash computation and abilinear pairings computation, it concludes that the new scheme reduces the computation cost in scalar multiplication computation in half and maintains the same in the other computation costs.

Key words: digital signature; aggregate signature; certificateless aggregate signature; security; bilinear maps
CLC number: TP 309.7

Received date: 2016-09-02
Foundation item: Supported by the Applied Basic and Advanced Technology Research Programs of Tianjin (15JCYBJC15900)
Biography: KANG Baoyuan, male, Professor, research direction: information security. E-mail: baoyuankang@aliyun.com

0 Introduction

To solve the key escrow problem in identity-based public key cryptography, a certificateless public key cryptography was proposed [1]. In certificateless public key cipher, the private key of one user consists of a partial private key generated by the key generation center (KGC) and the secret value selected by the user himself.

Aggregate signature [2] proposed by Boneh and colleages can combine n signatures from n users on n messages into a signature. It is useful in secure routing and certificate chain compression. After the first aggregate signature scheme [2], many aggregate signature schemes were proposed [3-9]. Several certificateless aggregate signature (CLAS) schemes are proposed [10-19]. In these schemes, Xiong *et al* [11] proposed a certificateless aggregate signature with constant pairing computations. However, Refs. [14, 15, 20] showed that Xiong *et al*'s scheme is subjected to several attacks. Also some improved schemes were proposed in Refs.[14, 15, 20]. Kang *et al* [19] also proposed a certificateless aggregate signature scheme. But the scheme needed high computations cost. In Ref.[21], the author also investigated the security of aggregate signatures.

In this paper, we propose an efficient certificateless aggregate signature scheme, and prove that the proposed scheme is existentially unforgeable under adaptive chosen-message attacks under the assumption that the computational Diffie-Hellman problem is hard. The new scheme combines aggregate signature with certificateless public key cryptography and improves the signature generating algorithm. Compared with some existing certificateless aggregate signature schemes in computation cost, the scalar multiplication computation cost of new scheme is reduced in half.

The rest of the paper is organized as follows. Section 1 introduces cryptographic hardness assumptions and the definition and security model of certificateless aggregate signature. In Section 2, a new certificateless aggregate signature scheme is proposed. Section 3 discusses the security of the proposed scheme. A comparison of computation cost is shown in Section 4. Finally, Section 5 concludes this paper.

1 Preliminaries

1.1 Bilinear Maps and Complexity Assumption

By Ref. [19], the bilinear map and related complexity assumption are depicted in following.

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

1) Bilinear: $e(aP,bQ)=e(P,Q)^{ab}$ for all $P,Q \in G_1$, $a,b \in Z_q^*$.

2) Non-degeneracy: There exists $P,Q \in G_1$ such that $e(P,Q) \neq 1$.

3) Computable: There exists an efficient algorithm to compute $e(P,Q)$ for any $P,Q \in G_1$.

Computational Diffie-Hellman (CDH) Problem: Given a generator P of an additive cyclic group G with order q, and given (ap,bp) for unknown, $a,b \in Z_q^*$ to compute abP.

1.2 Definition and Security Model of Certificateless Aggregate Signature Scheme

The definition and security model of CLAS are identical to that of Refs.[1, 13, 19].

A certificateless aggregate signature scheme includes a KGC, an aggregating set U of n users $U_1,\cdots,U_n$ and an aggregate signature generator. There are six algorithms: Setup, Partial-Private-Key-Extract, UserKeyGen, Sign, Aggregate and Aggregate Verify.

There are two types adversaries A_1 and A_2 for CLAS. Type 1 adversary A_1 does not have access to the master key, but it can replace the public key of any user. While type 2 adversary A_2 has access to the master-key but cannot perform public key replacement. The security of CLAS scheme is modeled by two games [19] between a challenger N and an adversary A_1 or A_2.

For more description, readers can refer to Refs. [1, 13, 19].

2 An Efficient Certificateless Aggregate Signature Scheme

In this section we proposed an efficient CLAS scheme. The new scheme consists of six algorithms:

Setup: Given a security parameter τ, the KGC selects an additive cyclic group G_1 and a multiplicative cyclic group G_2 with the same order q, and chooses a bilinear map $e: G_1 \times G_1 \to G_2$. Then, KGC selects a generator of G_1, P and random $s \in Z_q^*$ as the master key and sets system public key $P_{\text{pub}}=sP$. KGC also picks four cryptographic Hash functions.

$H_1,H_3,H_4:\{0,1\}^* \to G_1, H_2:\{0,1\}^* \times \{0,1\}^* \times G_1 \times G_1 \to Z_q^*$

The system parameter list is $\text{params}=(G_1,G_2,e,P,P_{\text{pub}}, H_1,H_2,H_3,H_4)$.

Partial-Private-Key-Extract: KGC generates the partial private key D_i for the user with identity ID_i as follows:

1) Calculate $Q_i=H_1(\text{ID}_i)$.

2) Output $D_i=sQ_i$.

UserKeyGen: This algorithm selects a random $x_i \in Z_q^*$ as one user's secret value, and generates the user's public key as $P_i=x_iP$.

Sign: Given a state information w, one user U_i with identity ID_i and public key P_i signs a message m_i as follows:

1) Select a random number $r_i \in Z_q^*$, calculates $R_i=r_iP$.

2) Calculate $h_i=H_2(m_i,\text{ID}_i,P_i,R_i)$, $Z=H_3(w)$, $F=H_4(w)$.

3) Calculate $T_i=h_iD_i+x_iZ+r_iF$.

4) Output signature $\sigma_i=(R_i,T_i)$.

Aggregate: For n message-signature pairs $((m_1, \sigma_1=(R_1,T_1)),\cdots,(m_n,\sigma_n=(R_n,T_n))$ from n users $U_1, \cdots,U_n$ (they has the same state information), respectively, any aggregate signature generator can compute $T=\sum_{i=1}^{n}T_i$ and output the aggregate signature $\sigma=(R_1,\cdots, R_n,T)$.

Aggregate Verify: To verify an aggregate signature $\sigma=(R_1,\cdots,R_n,T)$ on messages $m_1,\cdots,m_n$ from n user $U_1,\cdots,U_n$ with identities $\text{ID}_1,\cdots,\text{ID}_n$, corresponding public keys $P_1,\cdots,P_n$, and same state information w, the verifier does the following steps:

1) Calculate $Q_i=H_1(\text{ID}_i)$, $h_i=H_2(m_i,\text{ID}_i,P_i,R_i)$, for all $i, 1\leqslant i \leqslant n$, and $Z=H_3(w)$, $F=H_4(w)$.

2) Verify $e(T,P)\overset{?}{=}e(P_{\text{pub}},\sum_{i=1}^{n}h_iQ_i)e(Z,\sum_{i=1}^{n}P_i)$ $e(F,\sum_{i=1}^{n}R_i)$.

3) If the equation holds, output true. Otherwise, output false.

3 Security Proof

Theorem 1 The proposed certificateless aggregate scheme is existentially unforgeable against type 1 adversary under the assumption that the CDH problem is intractable.

Proof To prove the proposed scheme is existentially unforgeable against type 1 adversary, we show how a CDH attacker N_1 uses type 1 adversary A_1 to compute abP from (P,aP,bP).

Setup: N_1 sets system public $P_{\text{pub}}=aP$ and params $=(G_1,G_2,e,P,P_{\text{pub}},H_1,H_2,H_3,H_4)$ and sends params to A_1.

A_1 executes the following types of queries in an adaptive manner.

H_1 queries: There is a list H_1^{list} to record H_1 queries. When A_1 queries $H_1(\text{ID}_i)$, the same answer will be given if the query can be found on H_1^{list}. Otherwise, N_1 picks $\varepsilon_i\in Z_q^*$ at random and flips a coin $c_i\in\{0,1\}$. If $c_i=0$, N_1 sets $Q_i=\varepsilon_i(bP)$, adds $(\text{ID}_i,\perp,Q_i,c_i)$ to H_1^{list} and returns Q_i. Otherwise, N_1 sets $Q_i=\varepsilon_iP$, adds $(\text{ID}_i,\varepsilon_i,Q_i,c_i)$ to H_1^{list} and returns Q_i.

H_2 queries: There is a list H_2^{list} to record H_2 queries for same query getting same answer. When A_1 queries $H_2(m_i,\text{ID}_i,P_i,R_i)$, and the query cannot be found on H_2^{list}, N_1 picks a random $\delta_i\in Z_q^*$, adds $(m_i,\text{ID}_i,P_i,R_i,\delta_i)$ to H_2^{list} and then returns δ_i.

H_3 queries: There is a list H_3^{list} to record H_3 queries for same query getting same answer. When A_1 queries $H_3(w_i)$, and the query cannot be found on H_3^{list}, N_1 selects a random $\phi_i\in Z_q^*$, calculates $Z_i=\phi_iP$, adds (w_i,ϕ_i,Z_i) to H_3^{list} and then returns Z_i.

H_4 queries: There is a list H_4^{list} of tuples (w_i,μ_i,F_i) to record H_4 queries for same query getting same answer. Whenever A_1 issues a query $H_4(w_i)$, and the query cannot be found on H_4^{list}, N_1 picks a random $\mu_i\in Z_q^*$, calculates $F_i=\mu_iP$, adds (w_i,μ_i,F_i) to H_4^{list} and then returns F_i.

Partial-Private-Key queries: N_1 keeps a list K^{list} for same query getting same answer. When A_1 queries a Partial-Private-Key for ID_i, and the query can be found on K^{list}, N_1 first does an H_1 query on ID_i and finds the tuple $(\text{ID}_i,\varepsilon_i,Q_i,c_i)$ on H_1^{list}, then N_1 does as follows:

1) If $c_i=0$, N_1 aborts.

2) Else if there is a tuple $(\text{ID}_i,x_i,D_i,P_i)$ on K^{list}, N_1 sets $D_i=\varepsilon_iP_{\text{pub}}$, and returns D_i.

3) Otherwise, calculate $D_i=\varepsilon_iP_{\text{pub}}$, set $x_i=P_i=\perp$, return D_i as answer and add $(\text{ID}_i,x_i,D_i,P_i)$ on K^{list}.

Public-Key queries: To a Public-Key query on ID_i, if the query can be found on K^{list}, the same answer will be given. Otherwise, N_1 does as follows:

1) If there is a tuple $(\text{ID}_i,x_i,D_i,P_i)$ on K^{list} (in this case, the public key P_i of ID_i is $\perp$), choose $x_i'\in Z_q^*$, compute $P_i'=x_i'P$, return P_i' as answer and update $(\text{ID}_i,x_i,D_i,P_i)$ to $(\text{ID}_i,x_i',D_i,P_i')$.

2) Otherwise, select $x_i\in Z_q^*$ at random, calculate $P_i=x_iP$, return P_i as answer, set $D_i=\perp$ and add $(\text{ID}_i,x_i,D_i,P_i)$ to K^{list}.

Secret-Value queries: On receiving a Secret-Value query on ID_i, firstly, N_1 makes public-key query on ID_i, then finds $(\text{ID}_i,x_i,D_i,P_i)$ on list K^{list}, returns x_i (Note that the value of x_i maybe $\perp$).

Public-Key-Replacement queries: When N_1 receives a Public-Key-Replacement query, N_1 first finds $(\text{ID}_i,x_i,D_i,P_i)$ on list K^{list}, if such a tuple does not exist on list K^{list} or $P_i=\perp$, N_1 first makes Public-Key query on ID_i, then updates P_i to P_i'.

Sign queries: When N_1 receives a Sign query on m_i by one user with identity ID_i, firstly N_1 recovers $(\text{ID}_i,\varepsilon_i,Q_i,c_i)$ from H_1^{list}, $(m_i,\text{ID}_i,P_i,R_i,\delta_i)$ from H_2^{list}, then does as follows:

1) If $c_i=0$, select $k_i,n_i\in Z_q^*$, set $R_i=k_i^{-1}(n_iP-P_i-P_{\text{pub}})$, $Z=\delta_iQ_i$, $F=k_i\delta_iQ_i$ and record $(w_i,\perp,\delta_iQ_i)$ on list H_3^{list}, and $(w_i,\perp,k_i\delta_iQ_i)$ on list

H_4^{list}, then compute $T_i = n_i\delta_i Q_i$, output $\sigma_i = (R_i, T_i)$. Here δ_i is found in $(m_i, \text{ID}_i, P_i, R_i, \delta_i)$ from H_2^{list}.

2) If $c_i = 1$, randomly select $R_i \in G_1$, sets $T_i = \varepsilon_i\delta_i P_{\text{pub}} + \phi_i P_i + \mu_i R_i$, output $\sigma_i = (R_i, T_i)$. Here ϕ_i is found from (w_i, ϕ_i, Z_i) on H_3^{list}, μ_i is found from (w_i, μ_i, F_i) on H_4^{list}.

Forgery: A_1 outputs a forged aggregate signature $\sigma^* = \{R_1^*, \cdots, R_n^*, T^*\}$ under a set U of n users with identities set $L_{\text{ID}}^* = \{\text{ID}_1^*, \cdots, \text{ID}_n^*\}$ and the corresponding public keys set $L_{\text{PK}}^* = \{P_1^*, \cdots, P_n^*\}$, messages set $L_m^* = \{m_1^*, \cdots, m_n^*\}$, and a state information w^*. There exists $I \in \{1, \cdots, n\}$ such that A_1 has not asked the partial private key for ID_I, and the sign query on $(m_I^*, \text{ID}_I^*, P_I^*)$. Let $I = 1$. Then the forged aggregate signature satisfies

$$e(T^*, P) = e(P_{\text{pub}}, \sum_{i=1}^n h_i^* Q_i^*) e(F^*, \sum_{i=1}^n R_i^*) = e(Z^*, \sum_{i=1}^n P_i^*) \quad (1)$$

where

$Q_i^* = H_1(\text{ID}_i^*)$, $h_i^* = H_2(m_i^*, \text{ID}_i^*, P_i^*, R_i^*)$,

$Z^* = H_3(w^*)$, $F^* = H_4(w^*)$.

N_1 finds $(\text{ID}_i^*, \varepsilon_i^*, Q_i^*, c_i^*)$ from H_1^{list}, $(m_i^*, \text{ID}_i^*, P_i^*, R_i^*, \delta_i^*)$ from H_2^{list}, (w_i^*, ϕ_i^*, Z_i^*) from H_3^{list} and (w_i^*, μ_i^*, F_i^*) from H_4^{list} for all i, $1 \leqslant i \leqslant n$. N_1 proceeds only if $c_1^* = 0$, $c_i^* = 1$ for all i, $2 \leqslant i \leqslant n$. Otherwise, N_1 aborts.

From

$$e(T^*, P) = e(P_{\text{pub}}, \sum_{i=1}^n h_i^* Q_i^*) e(F^*, \sum_{i=1}^n R_i^*) e(Z^*, \sum_{i=1}^n P_i^*) \quad (2)$$

It holds

$$e(P_{\text{pub}}, h_1^* Q_1^*) = e(T^*, P)(e(P_{\text{pub}}, \sum_{i=2}^n h_i^* Q_i^*) \bullet e(Z^*, \sum_{i=2}^n P_i^*) e(F^*, \sum_{i=2}^n R_i^*))^{-1} \quad (3)$$

But, $Q_1^* = \varepsilon_1^*(bP)$, $h_1^* = \delta_1^*$, $Z^* = \phi^* P$, $F^* = \mu^* P$, and for all i, $2 \leqslant i \leqslant n$, $Q_i^* = \varepsilon_i^* P$, $h_i^* = \delta_i^*$. So, N_1 can calculate

$$abP = (\delta_1^* \varepsilon_1^*)^{-1}(T^* - \sum_{i=2}^n \delta_i^* \varepsilon_i^* P_{\text{pub}} - \sum_{i=1}^n (\phi^* P_i^* + \mu^* R_i^*))$$

Theorem 2 The proposed certificateless aggregate scheme is existentially unforgeable against type 2 adversary under the assumption that the CDH problem is intractable.

Proof To prove the proposed scheme is existentially unforgeable against type 2 adversary, we show how a CDH attacker N_2 uses type 2 adversary A_2 to compute *abp* from (P, Ap, bP).

Setup: Firstly, N_2 picks a random $\eta \in Z_q^*$ as the master-key, and sets system public key $P_{\text{pub}} = \eta P$ and system parameters

$$\text{params} = (G_1, G_2, e, P, P_{\text{pub}}, H_1, H_2, H_3, H_4)$$

Then he sends params and the master key η to A_2. Since A_2 has access to the master-key, there is no need to handle H_1 as random oracle.

To carry on attack, A_2 does the following types of queries in an adaptive manner.

H_2 queries: N_2 keeps a list H_2^{list} for same query getting same answer. When A_2 queries $H_2(m_i, \text{ID}_i, P_i, R_i)$, and the query cannot be found on H_2^{list}, N_2 picks a random $\delta_i \in Z_q^*$, and adds $(m_i, \text{ID}_i, P_i, R_i, \delta_i)$ to list H_2^{list}, then returns δ_i.

H_3 queries: N_2 keeps a list H_3^{list} for same query getting same answer . When A_2 queries $H_3(w_i)$, and the query cannot be found on H_3^{list}, N_2 picks a random $\phi_i \in Z_q^*$, calculates $Z_i = \phi_i(aP)$, adds (w_i, ϕ_i, Z_i) to H_3^{list}, then returns Z_i.

H_4 queries: There is a list H_4^{list} for same query getting same answer . When A_2 queries a query $H_4(w_i)$, and the query cannot be found on H_4^{list}, N_2 picks a random $\mu_i \in Z_q^*$, calculates $F_i = \mu_i P$, adds (w_i, μ_i, F_i) to H_4^{list}, and returns F_i.

Public-Key queries: To answer a Public-Key query on ID_i, if the request can be found on K^{list}, the same answer will be given. Otherwise, N_2 picks $x_i \in Z_q^*$ and flips a coin $c_i \in \{0,1\}$. If $c_i = 0$, N_2 returns $x_i(bP)$, adds $(\text{ID}_i, \perp, D_i, P_i, c_i)$ to list K^{list}. Otherwise, it calculates $P_i = x_i P$, and adds $(\text{ID}_i, x_i, D_i, P_i, c_i)$ to K^{list} and returns P_i.

Secret-Value queries: To answer Secret-Value query on ID_i, N_2 first finds the tuple on K^{list} .If $c_i = 0$, N_2 aborts, otherwise, returns x_i.

Sign queries: To answer Sign query on m_i by one user with identity ID_i. N_2 firstly finds $(\text{ID}_i, x_i, P_i, c_i)$ on K^{list}, (w_i, ϕ_i, Z_i) from $H_3^{\text{list}}(w_i, \phi_i, Z_i)$, and (w_i, μ_i, F_i) from H_4^{list}, then does as follows:

1) If $c_i = 0$, N_2 randomly selects $R_i \in G_1$, and $n_i \in Z_q^*$, sets $Z = n_i P$, adds $(w_i, n_i, n_i P)$ on H_3^{list}, and calculates $T_i = \eta\delta_i H_i(\text{ID}_i) + n_i x_i(bP) + \mu_i R_i$, outputs $\sigma_i = (R_i, T_i)$.

2) If $c_i = 1$, N_2 executes the standard sign algorithm to generate and outputs $\sigma_i = (R_i, T_i)$.

Forgery: Finally, A_1 returns a forged aggregate signature $\sigma^* = \{R_1^*, \cdots, R_n^*, T^*\}$ under a set U of n users with identities set $L_{\text{ID}}^* = \{\text{ID}_1^*, \cdots, \text{ID}_n^*\}$, the corresponding public keys set $L_{\text{PK}}^* = \{P_1^*, \cdots, P_n^*\}$, messages set $L_m^* = \{m_1^*, \cdots, m_n^*\}$, a state information w^*. There exists $I \in \{1, \cdots, n\}$ such that A_1 has not asked the partial private key for ID_I, and sign query on $(m_I^*, \text{ID}_I^*, P_I^*)$. Let $I = 1$. The forged aggregate signature satisfies

$$e(T^*, P) = e(P_{\text{pub}}, \sum_{i=1}^{n} h_i^* Q_i^*) e(F^*, \sum_{i=1}^{n} R_i^*) e(Z^*, \sum_{i=1}^{n} P_i^*) \tag{4}$$

Where $Q_i^* = H_1(\text{ID}_1^*)$, $h_i^* = H_2(m_i^*, \text{ID}_i^*, P_i^*, R_i^*)$, $Z^* = H_3(w^*)$, $F^* = H_4(w^*)$.

N_2 finds $(m_i^*, \text{ID}_i^*, P_i^*, R_i^*, \delta_i^*)$ from list H_2^{list}, (w_i, γ_i, Z_i) from list H_3^{list} and (w_i, μ_i, F_i) from list H_4^{list} for all $i, 1 \leqslant i \leqslant n$. N_2 proceeds only if $c_1^* = 0$, $c_i^* = 1$ for all $i, 2 \leqslant i \leqslant n$. Otherwise, N_2 aborts.

From

$$e(T^*, P) = e(P_{\text{pub}}, \sum_{i=1}^{n} h_i^* Q_i^*) e(F^*, \sum_{i=1}^{n} R_i^*) e(Z^*, \sum_{i=1}^{n} P_i^*) \tag{5}$$

It holds

$$e(Z^*, P_1^*) = e(T^*, P)(e(Z^*, \sum_{i=2}^{n} P_i^* e(P_{\text{pub}}, \sum_{i=1}^{n} h_i^* Q_i^*) \bullet e(F^*, \sum_{i=1}^{n} R_i^*)^{-1} \tag{6}$$

But, $h_1^* = \delta_1^*$, $Z^* = \phi_1^*(aP)$, $P_1^* = x_1^*(bP)$, $F^* = \mu^* P$, and for all $i, 2 \leqslant i \leqslant n$, $h_i^* = \delta_i^*$, $Z^* = \phi_i^* aP$, $P_i^* = x_i^* P$. Hence, N_2 can calculate

$$abP = (\phi_1^* x_1^*)^{-1} (T^* - \sum_{i=1}^{n} (\eta\delta_i^* Q_i^* + \mu^* R_i^*) - \sum_{i=2}^{n} \phi_i^*(aP))$$

4 Comparisons

In this section, we compare the proposed scheme with Refs. [14, 18] in computation cost. We laid stress on the comparisons of the computation in Partial-Private-Key-Extract Algorithm, Sign Algorithm, and Aggregate Verify Algorithm. The comparison result is show in Table 1. Obviously, the proposed scheme needs low computation cost.

Table 1 Comparisons of the computation cost

Scheme	P1	P2	P3
Cheng *et al* [14]	1H+1S	2H+5S+4D	2*n*H+2*n*S+2*n*D+3B
Chen *et al* [18]	1H+1S	4H+4S+3D	(3*n*+2)H+2*n*S+3*n*D+4B
Our scheme	1H+1S	3H+4S+3D	(2*n*+2)H+*n*S+3*n*D+4B

P1: Partial-Private-Key-Extract Algorithm; P2: Sign Algorithm; P3: Aggregate Verify Algorithm;

D: Group sum computation; S: Scalar multiplication computation; H: Hash computation; B: bilinear pairings computation

5 Conclusion

In this paper, a new certificateless aggregate signature scheme is proposed. It is proved that the new scheme is existentially unforgeable under adaptively chosen-message attacks assuming the computational Diffie-Hellman problem is hard. Furthermore, a comparison of the new scheme with some existing certificateless aggregate signature schemes indicates that the new scheme is more efficient. But, in Hash and bilinear pairings computation cost, the new scheme has insufficient predominance.

References

[1] AI-Riyami S, Paterson K. Certificateless public key cryptography [C] //*ASIACRYPT*'03, *LNCS* 2894. Heidelberg: Springer-Verlag, 2003:452-473.

[2] Boneh D, Gentry C, Shacham H, *et al*. Aggregate and verifiably encrypted signatures from bilinear maps [C] //*EUROCRPYT*'03, *LNCS* 2656. Heidelberg: Springer-Verlag, 2003:416-432.

[3] Cheng X, Liu J, Wang X. Identity-based aggregate and verifiably encrypted signatures from bilinear pairing [C] //

ICCSA'05, *LNC*S 3483. Heidelberg: Springer-Verlag, 2005: 1046-1054.

[4] Gentry C, Ramzan Z. Identity-based aggregate signature [C] // *PKC*'06, *LNCS*3958. Heidelberg: Springer-Verlag, 2006: 257-273.

[5] Lu S, Ostrovsky R, Sahai A, *et al*. Sequential aggregate signatures and multisignatures without random oracles [C] // *EUROCRPYT*'06, *LNCS* 4004. Heidelberg: Springer-Verlag, 2006:465-485.

[6] Ruckert M, Schrode D. Aggregate and verifiably encrypted signatures from multilinear maps without random oracles [C] // *ISA*'09, *LNCS* 5576. Heidelberg: Springer-Verlag, 2009: 750-759.

[7] Shao Z. Enhanced aggregate signature from pairings [C] // *CISC*'05, *LNCS* 3822. Heidelberg: Springer-Verlag, 2005: 140-149.

[8] Shim K. An Id-based aggregate signature scheme with constant pairing computations [J]. *The Journal of System and Software*, 2010, **83**: 1873-1880.

[9] Kang B Y. ID-based aggregate signature scheme with constant pairing computations: attack and new construction [J]. *Journal of Computational Information Systems*, 2012, **16** : 6611- 6618.

[10] Gong Z, Long Y, Hong X, *et al*. Two certificateless aggregate signatures from bilinear maps [C] // *Eighth ACIS International Conference on Software Engineering*, *Artificial Intelligence*, *Networking*, *and Parallel/Distributed Computing*, *ACIS* 2007. Washington D C: IEEE Press, 2007: 188-193.

[11] Xiong H, Guan Z, Chen Z, *et al*. An efficient certificateless aggregate signature with constant pairing computations [J]. *Information Sciences*, 2013, **10** : 225-235.

[12] Yanai N, Tso R, Mambo M, *et al*. Certificateless ordered sequential aggregate signature scheme [C] // *Third International Conference on Intelligent Networking and Collaborative Systems*, *INCos* 2011.Washington D C: IEEE Press, 2011: 662-667.

[13] Zhang L, Zhang F. A new certificateless aggregate signature scheme [J]. *Computer Communication*, 2009, **32**: 1079-1085.

[14] Cheng L, Wen Q, Jin Z, *et al*. Cryptanalysis and improvement mof a certificateless aggregate signature scheme [J]. *Information Sciences*, 2015, **295** : 337-346.

[15] Zhang F, Shen L, Wu G. Notes on the security of certificateless aggregate signature schemes [J]. *Information Sciences*, 2014, **287** : 32-37.

[16] Horng S, Tzeng S, Huang P, *et al*. An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks [J]. *Information Sciences*, 2015, **317** : 48-66.

[17] Du H, Huang M, Wen Q. Efficient and provably-secure certificateless aggregate signature scheme [J]. *Acta Electronica Sinica* , 2013, **41**(1): 72-76.

[18] Chen H, Wei S, Zhu C, *et al*. Secure certificateless aggregate signature scheme [J]. *Journal of Software*, 2015, **26**(5): 1173-1180 .

[19] Kang B Y, Xu D. A Secure certificateless aggregate signature scheme [J]. *International Journal of Security and Its Applications*, 2016, **10**(3): 55- 68.

[20] He D, Tian M. A note on an efficient certificateless aggregate signature with constant pairing computations [EB/OL]. [2012-08-05]. *http*: *//eprint.iacr.org/*2012/445.

[21] Kang B Y. On the security of some aggregate signature schemes [J]. *Journal of Applied Mathematics*, 2012, Article ID 416137, DOI:10.1155/2012/416137.

□