AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°1; pp 174-188
Submitted Jan. 2017; Revised March 15, 2017, Accepted April 15, 2017

Security Analysis and Improvement of the Certificateless Aggregate Signature Schemes

*Aiwan Fan, **Qiming Wang*

* Computer school, Pingdingshan University,
Pingdingshan 467002, Henan province, China (faw_1978@163.com)
** College of Information Engineering, Pingdingshan University,
Pingdingshan 467002, Henan province, China (wqm8157@126.com)

Abstract

The aggregate signer outputs one signature through the aggregate signature algorithm, and the aggregate signature verifier confirms that multiple users have signed through the aggregate signature verification algorithm. Due to the high signature efficiency and low requirements for broadband, the aggregate signature scheme is widely studied and applied. For most aggregate signature schemes based on the traditional cryptography, key escrow is very complicated, while for identity-based aggregate signature schemes, there is a problem with certificate management. The certificateless public key cryptosystem can effectively solve these two drawbacks, but most certificateless aggregate signature schemes are of low computation efficiency and have security problems. Regarding the scheme proposed by He et al., in this paper, we construct a concrete attack method to prove it cannot meet the unforgeability requirement and analyze the reason for the successful attack, which is that during the signature process, the attack scheme can solve the definite value composed of the user's secret value and the generator. Therefore, by bonding the hash function with the public key information, we design two improved certificateless aggregate signature schemes with strong security. One is to break the composition relations between the secret value and other parameters, and the other is to make the composition relations of the definite value unsolvable. In the random oracle model, we prove the second improved scheme to be unforgeable and its security is equivalent to solving hard mathematical problems. Compared with the current similar schemes, the second improved scheme requires less computation. We take the aggregate signature for the smart grid as an example and introduce the improved scheme

into the aggregate signature scenario. The test results show that the scheme can ensure security and at the same time achieve real-time data transmission.

Key words

Certificateless cryptosystem, Aggregate signature, Bilinear pairing, Random oracle model, Smart grid

1. Introduction

The concept of aggregate signature was proposed by Boneh et al. [1]. Its idea is that multiple users can sign multiple messages, that these signatures are output by the aggregate signer through the aggregate signature algorithm as one signature and that the aggregate signature verifier then confirms whether multiple users have signed through the aggregate signature verification algorithm. The aggregate signature can improve the efficiency of signature verification and at the same time save bandwidth. With the development of cryptography, aggregate signature schemes based on various cryptography theories have been proposed. [2] proposed an sequential aggregate scheme based on the trapdoor permutation theory. Based on [1], [3] proposed a new aggregate signature security model. Since Shamir proposed the identity-based public key cryptography [4], many applied researches have been carried out on the identity-based cryptosystem scenarios, and a large number of identity-based aggregate schemes [5-7] have also been proposed. But these identity-based aggregate signature schemes have problems in certificate management and key escrow.

Certificateless public key cryptosystem was proposed by AL-Riyami and Paterson [8] in 2003 to effectively address the key escrow problem in the identity-based cryptosystem and reduce the complexity of certificate storage and management in the certificate-based public key system. Gong et al. [9] introduced the certificateless idea into aggregate signature for the first time and designed a certificateless aggregate signature security model. Subsequently, certificateless aggregate signature schemes with various attributes were presented, mainly trying to improve the computation efficiency and guarantee security. In the certificateless aggregate signature algorithm proposed by Zhang et al. [10], with the increase in the number of signers, the aggregate signature will become longer and the amount of computation required for verification of the aggregate signature is also increased. Although Zhang et al. later improved the original scheme [11], the partial private key generated by the key generation center (KGC) for each signer requires two group elements, reducing its efficiency, and for the aggregate signature, users need to use a uniform clock, which is difficult to achieve in practice. In order to improve the efficiency

of aggregate signature, Xiong et al. [12] designed a new certificateless aggregate signature scheme. However, this scheme was proved by SHEN et al. [13] in the same year to be insecure against Type II attacks. Afterwards, Zhang et al. [14] analyzed the security of the previous certificateless aggregate signature scheme. In order to resist malicious-but-passive KGC attacks, they proposed two improved schemes. Luo et al. [15] studied and analyzed the two schemes, and after careful design of the attack, they proved that the second scheme could not resist the attacks by two kinds of attackers and put forward an improved scheme.

He et al. [16] also gave a new attack pattern for the certificateless aggregate signature scheme proposed by Xiong et al. and proposed an improved scheme. In this paper, we analyze the security of the scheme proposed by He et al. and find that this scheme cannot resist forgery attacks, and then we analyze the reason for such successful attacks and propose two more secure certificateless aggregate signature scheme. We take the second scheme as an example and prove that the improved scheme is unforgeable in the random oracle model. Compared with other similar schemes, the second scheme proposed requires less computation and is more suitable for use in many-to-one communication systems, especially in the smart grid aggregate signature scenarios.

2. Preliminaries

This section is the basis for the research on certificateless aggregate signature schemes. It consists of four parts - bilinear pairing, mathematical problem, composition of the certificateless aggregate signature system and security model. The common preconditions for the four parts are as follows.

Condition 1: let G_1 be an additive group of prime order q. P is a generator of G_1.

Condition 2: let G_2 be a multiplicative group of prime order q.

2.1 Sub-Section A -Subtitle

Based on Condition 1 and 2, U and V are two elements of G_1. e is a map of $G_1 \times G_1 \rightarrow G_2$. If e is a bilinear pairing, the following three conditions must hold.

(1) Non-degeneracy: $e(U, V) \neq 1$.

(2) Computability: there exists an efficient algorithm to compute $e(U, V)$.

(3) Bilinearity: if $a, b \in Z_q^*$, then $e(aU, bV)=e(U, V)^{ab}$.

2.2 Mathematical problem

Definition 1 Computation diffie-hellman problem (CDHP): based on Condition 1, given aP, bP, $cP \in G_1$ for unknown a, b, $c \in Z_q^*$, it is computationally intractable to compute abP.

2.3 Composition of the certificateless aggregate signature system

The certificateless aggregate signature system has four roles - trusted KGC, signer, aggregate signer and aggregate signature verifier, and the system consists of 7 algorithms.

(1) Setup

Participant: KGC; input: a random security parameter; output: a system foundation platform parameter, exposed by KGC; a system master key, saved by KGC.

(2) Generation of partial private key by KGC

Participant: KGC; input: users' identity, system foundation platform parameter and system master key; output: user' partial private key, sent to the user in a secure form.

(3) Setting of user secrete value

Participant: valid user; input: system foundation platform parameter and user' identity; output: user's secret value, saved by the user; user's public key, sent by the user to KGC.

(4) Setting of the private and public keys

Participant: KGC and valid user; input: user's partial private key, user's secret value and user's public key; output: private key pair saved by the user (user's partial private key and user's secret value) and the public key exposed by KGC.

(5) Signature

Participant: signer; input: system foundation platform parameter, identity, private key pair, public key and message; output: signature information.

(6) Signature aggregate

Participant: signature aggregator; input: messages and signature information of multiple signers; output: one piece of aggregate signature information.

(7) Aggregate signature verification

Participant: aggregate signature verifier; input: messages and public key information of multiple senders, aggregate signature information; output: if the aggregate signature passes verification, the algorithm outputs true; otherwise it outputs false.

2.4 Security model of the certificateless aggregate signature system

In the security model of certificateless aggregate signature, there are two types of attacks Type I and Type Ⅱ launched by two types of attackers $A1$ and $A2$.

(1) Type I attack

Condition: $A1$ can randomly replace the user's public key, but cannot acquire the master key of the trusted KGC.

Attacker: invalid user.

(2) Type Ⅱ attack

Condition: $A2$ can acquire the master key of the trusted KGC, but cannot randomly replace the user's public key.

Attacker: trusted KGC

3. Certificateless Aggregate Signature Scheme Proposed by He et al.

This section describes the scheme proposed by He et al. in detail. This scheme is a standard certificateless aggregate signature scheme, whose algorithms are as follows:

(1) Setup: input a random security parameter $k \in Z_q^*$. Output a system foundation platform parameter *params*, which is exposed by KGC, and a system master key s, which is saved by KGC.

1) Preparation for setup. KGC generates an additive group G_1 and a multiplicative group G_2 of random prime order q ($0<q<2^k$). P and Q are two generators of G_1. e is a linear map of $G_1 \times G_1 \rightarrow G_2$. Three hash functions are designed: H_1: $\{0,1\}^* \rightarrow G_1$, and H_2: $\{0, 1\}^* \rightarrow Z_q^*$.

2) Generation of the system master key and public key. KGC randomly uses $s \in Z_q^*$ as the system master key and computes $P_{pub}=sP$ as the system public key.

3) Generation of the system foundation platform parameter. KGC generates $params=(q, G_1, G_2, e, P, P_{pub}, H_1, H_2, H_3)$.

(2) Generation of the partial private key by KGC: input the user's identity ID_i, system parameter *params* and the master key s. KGC computes $Q_i= H_1(ID_i)$ and $d_i = sQ_i$; the user's partial private key d_i is sent to the user in a secure form.

(3) Setting of user's secret value: input system foundation platform parameter *params* and the user's identity ID_i; output a random numberof the user's secret value $x_i \in Z_q^*$, saved by user; a user's public key $PK_i=x_iP$, sent by the user to KGC.

(4) Setting of the private and public keys: input the user's partial private key d_i , the user's secret valuex_i and the user's public key PK_i; output a private key pair saved by the user (d_i, x_i) and the public key PK_i exposed by KGC.

(5) Signature: input system foundation platform parameter *params*, identity ID_i, private key pair (d_i, x_i), public key PK_i and message m_i; output a piece of signature information σ_i..

1) Randomly select $r_i \in Z_q^*$ and computes $U_i=r_iP$.

2) Compute $h_i = H_2(\text{“0”}, ID_i, PK_i, m_i, U_i)$ and $k_i = H_2(\text{“1”}, ID_i, PK_i, m_i, U_i)$; compute $V_i= h_ir_iP_{pub} + k_ix_iQ + d_i$.

3) Signature $\sigma_i = (U_i, V_i)$.

(6) Aggregate signature: input the messages and signature information of n senders (ID_i, m_i, PK_i, σ_i) ($1 \le i \le n$); the aggregator outputs one aggregate signature σ through the following steps.

1) Compute $Q_i= H_1(ID_i)$, $h_i = H_2(\text{“0”}, ID_i, PK_i, m_i, U_i)$ and $k_i = H_2(\text{“1”}, ID_i, PK_i, m_i, U_i)$.

2) Check the signatures sent by users to see if $e(P, V_i) = e(k_i\ PKi, Q)e(P_{pub}, Q_i+h_iU_i)$. If it holds, then, go to 3) to carry out the aggregate signature operation; otherwise, terminate the algorithm and reject the aggregate signature.

3) Compute $V = \sum_{i=1}^{n} V_i$.

4) Aggregate signature $\sigma = (U, V)$.

(7) Aggregate signature verification: input the messages and public key information of n senders (ID_i, m_i, PK_i) ($1 \le i \le n$), aggregate signature σ. After the aggregate signature verifier finishes the following steps, output the aggregate signature verification result.

1) Compute $Q_i= H_1(ID_i)$, $h_i = H_2(\text{“0”}, ID_i, PK_i, m_i, U_i)$ and $k_i = H_2(\text{“1”}, ID_i, PK_i, m_i, U_i)$.

2) Verify if $e(P,V) \stackrel{?}{=} e(P_{pub}, \sum_{i=1}^{n} Q_i + h_iU_i)e(\sum_{i=1}^{n} k_iPK_i, Q)$ holds. If it does, output *true*; otherwise output *false*.

4. Analysis on the Security of the Certificateless Aggregate Signature Scheme Proposed by He et al.

In [16], He et al. claimed the scheme was unforgeable under adaptive chosen-message, chosen identity and public key replacement attacks. In this section, we use a concrete attack to prove that this scheme is not unforgeable under Type II attacks.

Suppose User A and User B are the sender and the receiver, respectively. A's private key is (d_A, sk_A) and its public key is PK_A. The attacker $A2$ impersonates $A$and forges the valid message by taking the following steps.

(1) Preparation for forgery: according to the previous signature information that A sends to B $\sigma_A = (U'_A, V'_A)$ and m'_A, $A2$ calculates the interim parameter that can participate in forging signature $T=x_AQ$. According to $V'_A = d_A + h'_Ar'_AP_{pub} + k'_Ax_AQ$, we have $T=x_AQ= k'^{-1}_A(V'_A - d_A - h'_AsU'_A)$. As $A2$ can calculate $d_A = sH_1(ID_A)$, $h'_A = H_2(\text{“0”}, m'_A, ID_A, pk_A, U'_A)$ and $k'_A = H_2(\text{“1”},m_A, ID_A, pk_A, U'_A)$,

so $A2$ can successfully obtain the interim parameter that participates in the signature forgery $T=x_AQ$.

(2) Forgery: $A2$ forges the message m_A, and randomly selects $r_A \in Z_q^*$; it computes $U_A=r_A P$; and computes $h_A = H_2(m_A, ID_A, pk_A, U_A)$, $h_A = H_2(\text{“0”}, m_A, ID_A, PK_A, U_A)$, $k_A = H_2(\text{“1”}, m_A, ID_A, PK_A, U_A)$ and $V_A = d_A + h_A r_A P_{pub} + k_A x_A Q = d_A + h_A r_A P_{pub} + k_A + k_i T$.

(3) $A2$ impersonates A to send the message m_A and $\sigma_{iA}= (U_i, V_i)$ to B.

Theorem 1. The signature that $A2$ generates by the above method is valid.

Proof: The signature generated can pass the verification by the aggregator.

$$
\begin{aligned}
&e(P_{pub}, Q_A + h_A U_A) e(k_A PK_A, Q) \\
&= e(sP, Q_A + h_A r_A P) e(k_A x_A P, Q) \\
&= e(P, sQ_A + h_A r_A sP) e(P, k_A x_A Q) \\
&= e(P, sQ_A + h_A r_A P_{pub} + k_A T) \\
&= e(P, V_A)
\end{aligned}
$$

From this, it can be seen that if the KGC attacker $A2$ can forge a user's invalid signature through the forging algorithm, it can also forge the invalid signatures of multiple users through the forging algorithm, and the aggregate verifier will also accept the aggregate signature through aggregate operations.

The above analysis shows that the aggregator verifies the authenticity of the sender' signature by $e(P, V_i) = e(P_{pub}, Q_i + h_i U_i)\, e(k_i\, PK_i, Q)$. As $e(P_{pub}, Q_i + h_i U_i)\, e(k_i\, PK_i, Q) = e(P, d_i + s h_i U_i + k_i\, x_i Q)$, for the attacker $A2$, the only element unknown to him/her is x_i, but through observation and analysis, we can see that $x_i Q$ is a definite value in the signature process. According to $V_i = d_i + s h_i U_i + k_i\, x_i Q$, on condition that $A2$ can acquire the message m_i publicly sent by the user and the signature information V_i and U_i, $A2$ can calculate k_i and h_i, and then calculate $x_i Q$ according to $k_i^{-1}(V_i - d_i - h_i s U_i)$. So the attacker $A2$ can arbitrarily forge the user's signature with the definite value $x_i Q$.

5. Improvements of the Certificateless Aggregate Signature Scheme Proposed by He et al.

In order to resist $A2$'s attack, we either break the composition relations between x_i and other parameters that forms the definite value, or make the composition relations unsolvable. In this section, we propose two improved schemes which modify the signature and the aggregate signature verification algorithms respectively to resist $A2$'s attack.

5.1 Improved Scheme 1

In order to break the composition relations between x_i and other parameters that forms the definite value, we build a certificateless aggregate signature scheme, in which 4 algorithms - the KGC setup (generation of the system master key and the public key and generation of system foundation platform parameter), generation of the partial private key, setting of user's secret value and setting of the private and public keys – are the same as in the scheme proposed by He et al.. We improve 4 algorithms including the preparation for setup in setup, signature, aggregate signature and aggregate signature verification.

(1) Preparation for setup in setup: add one hash function H_3: $\{0,1\}^* \rightarrow G_1$.

(2) Signature: add one variable $K_i = H_3(ID_i, PKi, U_i)$, and modify the signature algorithm as $V_i = d_i + h_i r_i P_{pub} + k_i x_i K_i$.

(3) Aggregate signature: modify the aggregator verification algorithm as $e(P, V_i) = e(P_{pub}, Q_i + h_i U_i)\, e(k_i\, PK_i, K_i)$.

(4) Aggregate signature verification: modify the verifier verification algorithm as $e(P,V) \stackrel{?}{=} e(P_{pub}, \sum_{i=1}^{n} Q_i + h_i U_i) \prod_{i=1}^{n} e(k_i PK_i, K_i)$.

After the above changes are made, the variable K_i will vary with the value of U_i in the session, so that the $x_i K_i$ in each session will no longer be a definite value to prevent the forgery of V_i. However, in the aggregate signature verification process, due to $\prod_{i=1}^{n} e(k_i PK_i, K_i)$, the bilinear pairing computation will increase with the increasing number of users, reducing the computation efficiency.

5.2 Improved Scheme 2

In order to make the composition relations between x_i and other parameters that forms the definite value unsolvable, we construct a certificateless aggregate signature scheme, where 4 algorithms - the KGC setup (generation of the system master key and the public key and generation of system foundation platform parameter), generation of the partial private key, setting of user's secret value and setting of the private and public keys – are the same as in the scheme proposed by He et al.. We improve 4 algorithms including the preparation for setup in setup, signature, aggregate signature and aggregate signature verification.

(1) Preparation for setup in setup: add one hash function H_3: $\{0,1\}^* \rightarrow G_1$;

(2) Signature: add one variable $T = H_3(P_{pub})$, and modify the signature algorithm to $V_i = d_i + h_i r_i T + k_i x_i Q$.

(3) Aggregate signature: modify the aggregator verification algorithm as $e(P, V_i) = e(P_{pub}, Q_i)\, e(P, h_i U_i)\, e(k_i PK_i, Q)$.

(4) Aggregate signature verification: modify the verifier verification algorithm as $e(P,V) \stackrel{?}{=} e(P_{pub}, \sum_{i=1}^{n} Q_i) e(P, \sum_{i=1}^{n} h_i U_i) e(\sum_{i=1}^{n} k_i PK_i, Q)$.

After the above changes are made, r_i in each session is randomly generated, which is unknown to the attacker, so $h_i r_i T$ is also unknown and the value of each session is different. Even if the attacker $A2$ acquires the message m_i publicly sent by the user, and calculates k_i and h_i based on V_i and U_i, he/she still cannot solve $x_i Q = k_i^{-1}(V_i - d_i - h_i r_i T)$. This improvement can prevent the attacker $A2$ from forging V_i, and during the aggregate signature verification process, the bilinear pairing computation will not increase with the increasing number of users, indicating that the computation efficiency is high.

6. Analysis on the Security and Efficiency of Improved Scheme 2

As we already indicate in Section 5.1 that the first improved scheme is inefficient, in this section, we only analyze the security and efficiency of the second improved scheme.

6.1 Correctness

Theorem 2. The second improved scheme is correct.

Proof: For the second scheme, two correctness verifications need to be done.

(1) The aggregator verifies the correctness of users' signatures

$$\begin{aligned} e(P,V) &= e(P_{pub}, Q_i) e(P, h_i U_i) e(k_i PK_i, Q) \\ &= e(P, sQ_i) e(P, h_i U_i) e(k_i x_i P, Q) \\ &= e(P, sQ_i + h_i U_i + k_i x_i Q) \\ &= e(P, d_i + h_i U_i + k_i x_i Q) \end{aligned}$$

(2) Verify the correctness of the aggregate signature σ.

$$\begin{aligned} e(P,V) &= e(P_{pub}, \sum_{i=1}^{n} Q_i) e(P, \sum_{i=1}^{n} h_i U_i) e(\sum_{i=1}^{n} k_i PK_i, Q) \\ &= e(P, \sum_{i=1}^{n} sQ_i) e(P, \sum_{i=1}^{n} h_i U_i) e(P, \sum_{i=1}^{n} k_i x_i Q) \\ &= e(P, \sum_{i=1}^{n} sQ_i + h_i U_i + k_i x_i Q) \\ &= e(P, \sum_{i=1}^{n} V_i) \end{aligned}$$

6.2 Unforgeability

Theorem 3. In the random oracle model, if a Type I attacker $A1$ can successfully attack the second improved scheme proposed in this paper with a non-negligible probability, there must exist an algorithm F that can solve the CDHP with a non-negligible probability in polynomial time.

Setup: F computes $P_{pub}=aP$. C sends $params=(q, G_1, G_2, e, P, P_{pub}, H_1, H_2, H_3)$ to $A1$.

H_1query: F builds the H_1 list (ID_i, h_i). When $A1$ makes a query on ID_i to F, if there is a record of ID_i in the list, F sends the h_1 value of ID_i to $A1$; otherwise, F randomly selects $h_1 \in Z_q^*$, adds it to the H_1 list, and sends the h_i value to $A1$.

1) If $ID_i \neq ID_j$, F randomly selects $t_i \in Z_q^*$, and sends the value of $d_i=t_iP$ to $A1$.

2) If $ID_i= ID_j$, F randomly selects $t_i \in Z_q^*$, and sends the value of $d_i=t_ibP$ to $A1$.

H_2 query: F builds the H_2 list $(m_i, ID_i, PK_i, U_i, h_2)$. When $A1$ makes a query on ID_i to F, if there is a record of ID_i in the list, F sends the h_2 value to $A1$; otherwise, F randomly selects $h_2 \in Z_q^*$, adds it to the H_2 list, and sends the h_2 value to $A1$.

H_3 query: F builds the H_3 list (P_{pub}, h_3). . When $A1$ makes a query on ID_i to F, F randomly selects $f_i \in Z_q^*$, computes $h_3=f_iP$, adds it to the H_3 list and sends the h_3 value to $A1$.

Query on the generation of user keys: F builds the public and private key (ID_i, PKi, x_i) list. When $A1$ makes a query on ID_i to F. If there is a record of ID_i in the list, F sends the public and private key values of ID_i to $A1$; otherwise, F randomly selects $x_i \in Z_q^*$ as the private key of ID_i, computes the public key $PKi= x_iP$, and adds it in the public and private key list.

Query on the partial private key: F builds the partial private key (ID_i, PK_i, d_i) list. When $A1$ makes a query on ID_i to F:

1) If $ID_i \neq ID_j$, it turns to the query on H_1, and sends the value of d_i queried to $A1$.

2) If $ID_i= ID_j$, it terminates the algorithm.

Public key query: when $A1$ makes a query on ID_i to F, randomly selects $x_i \in Z_q^*$ as the private key of ID_i, computes the public key $PK_i= x_iP$, and records it in the public and private key list.

Public key replacement query: when $A1$ makes a query on ID_i to F, $A1$ first selects PK_i to replace the public key of ID_i, and then $A1$ sends the public and private key (ID_i, PK_i, x_i) list to F. F adds it to the public and private key list.

Signature query: $A1$ makes a query on (m, ID_i, PK_i) to F. If $ID_i \neq ID_j$, F calculates $\sigma_i= (U_i, V_i)$ according to the signature algorithm proposed in this paper, adds it to the appropriate list and sends the result to $A1$; otherwise, F terminates the algorithm.

If F does not return "$\perp$" in the query process, we obtain the following equations.

$$e(P,V) = e(P_{pub}, \sum_{i=1}^{n} Q_i) e(P, \sum_{i=1}^{n} h_i U_i) e(\sum_{i=1}^{n} k_i PK_i, Q)$$

$$= e(aP, \sum_{i=1, i\neq j}^{n} t_i P + t_j bP) e(P, \sum_{i=1}^{n} h_i U_i) e(P, \sum_{i=1}^{n} k_i x_i Q)$$

$$= e(P, \sum_{i=1, i\neq j}^{n} t_i aP + t_j abP + \sum_{i=1}^{n} h_i U_i + k_i x_i Q)$$

$$abP = t_j^{-1}(V - \sum_{i=1, i\neq j}^{n} t_i aP - \sum_{i=1}^{n} (h_i U_i + k_i x_i Q))$$

F obtains an instance of solving abP. If $A1$ successfully forges the signature, then F can utilize $A1$ to obtain a solution to CDHP. To avoid this situation, the probability of the adversary $A1$ encountering failure in the query on partial private key should be at least $\varepsilon' = \frac{1}{q_{sk}+n}(1-\frac{1}{q_{sk}+n})^{q_{sk}+n-1}\varepsilon$, where q_{sk} is the number of queries made on the partial private key. So in polynomial time, F should have at least an advantage of ε in solving CDHP. Therefore, the second improved scheme proposed in this paper can resist the adaptive chosen-message attacks from the adversary $A1$ in the random oracle model.

Theorem 4. In the random oracle model, if a Type Ⅱ attacker $A2$ can successfully attack the second improved scheme proposed in this paper with a non-negligible probability, there must exist an algorithm F that can solve the CDHP with a non-negligible probability in polynomial time.

Setup: F computes $P_{pub}=sP$. C sends $params=(q, G_1, G_2, e, P, P_{pub}, H_1, H_2, H_3)$ to $A2$.

H_1 query: F builds the H_1 list(ID_i, h_i). When $A2$ makes a query on ID_i to F, if there is a record of ID_i in the list, F sends the h_1 value of ID_i to $A2$; otherwise, F randomly selects $h_1 \in Z_q^*$, adds $d_i=sh_i$ to the H_1 list and sends the h_i value to $A2$.

H_2 query: F builds *the* H_2 list (m_i, ID_i, PK_i, U_i, h_2). When $A2$ makes a query on ID_i to F, if there is a record of ID_i in the list, F sends the h_2 value to $A2$; otherwise F randomly selects $h_2 \in Z_q^*$, adds it to the H_2 list and sends the h_2 value to $A2$.

H_3 query: F builds the H_3 list (P_{pub}, h_3). When $A2$ makes a query on ID_i to F, F computes $h_3=bP$, adds it to the H_3 list, and sends the h_3 value to $A2$.

Query on the partial private key: F builds the public and private key (ID_i, PK_i, x_i) list. When $A2$ makes a query on ID_i to F:

1) If $ID_i \neq ID_j$, F randomly selects $t_i \in Z_q^*$, and sends the value of $PK_i=x_iP$ to $A2$.

2) If $ID_i= ID_j$, F randomly selects $t_i \in Z_q^*$, and sends the value of $PK_i=x_iaP$ to $A2$.

Query on the partial private key: F builds the partial private key(ID_i, PK_i, d_i) list. When $A2$ makes a query on ID_i to F, it turns to the query on H_1, acquires the value of d_i and adds it to the list.

Signature query: $A2$ makes a query on (m, ID_i, PK_i) to F. If $ID_i \neq ID_j$, F calculates $\sigma_i = (U_i, V_i)$ according to the signature algorithm proposed in this paper, adds it to the appropriate list and sends the result to $A2$; otherwise, F terminates the algorithm.

If F does not return "$\perp$" in the query process, we obtain the following equations.

$$\begin{aligned} e(P,V) &= e(P_{pub}, \sum_{i=1}^{n} Q_i) e(P, \sum_{i=1}^{n} h_i U_i) e(\sum_{i=1}^{n} k_i PK_i, Q) \\ &= e(sP, \sum_{i=1}^{n} Q_i) e(P, \sum_{i=1}^{n} h_i U_i) e(\sum_{i=1, i\neq j}^{n} k_i x_i P + k_j x_j aP, bP) \\ &= e(P, \sum_{i=1}^{n} sQ_i + \sum_{i=1}^{n} h_i U_i + \sum_{i=1, i\neq j}^{n} bk_i x_i P + k_j x_j abP, Q) \\ abP &= k_j^{-1} x_j^{-1} (V - \sum_{i=1}^{n} sQ_i - \sum_{i=1}^{n} h_i U_i - \sum_{i=1, i\neq j}^{n} bk_i x_i P) \end{aligned}$$

F obtains an instance of solving abP. If $A2$ successfully forges the signature, then F can utilize $A2$ to obtain a solution to CDHP. To avoid this situation, the probability of the adversary $A2$ encountering failure in the query on partial private key should be at least $\varepsilon' = \frac{1}{q_x + n}(1 - \frac{1}{q_x + n})^{q_x + n - 1} \varepsilon$, where q_x is the number of queries made on the partial private key. So, in polynomial time, F should have at least an advantage of ε in solving CDHP. Therefore, the second improved scheme proposed in this paper can resist the adaptive chosen-message attacks from the adversary $A2$ in the random oracle model.

6.3 Comparison with Other Similar Cases in Security

From Table 1 Comparison between Certificateless Aggregate Signature Schemes in Security, we can see that: in terms of resistance against forgery attacks of random users, [14] has been proved in [15] that it cannot resist such attacks. [18] is also vulnerable to the forgery attacks of random users. All other schemes can resist Type I attacks. In terms of Type Ⅱ attacks, [12] and [17] have been proved to be insecure against Type Ⅱ attacks. [16] is analyzed in this paper and proved that it is not unforgeable. Therefore, the second improved scheme proposed in this paper is more secure than other similar schemes.

Tab.1. Comparison between Similar Schemes in Security

Scheme	Type I	Type Ⅱ	Security
Reference [17]	√	×	Insecure
Reference [12]	√	×	Insecure
Reference [16]	√	×	Insecure
Reference [14]	×	×	Insecure
Reference [18]	×	×	Insecure
This paper	√	√	Secure

6.4 Efficiency analysis

We measure the efficiency of a scheme by its computation amount. Suppose n stands for the number of users participating in the signcryption, P stands for 1 bilinear pairing operation, and M stands for the scalar multiplication. Based on the analysis of [3], the bilinear pairing is much larger than the scalar multiplication. Table 2 shows the comparison between similar schemes in computation amount. For the current certificateless aggregate signature schemes, we compare the time of the bilinear pairing operation in the aggregate signature and aggregate signature verification algorithms. [17] is the most efficient one, requiring only 2 operations. Then it is [12] and [16], requiring 3. Other schemes and the second improved scheme proposed in this paper require 4 operations. However, in terms of security, [17], [12] and [16] are all insecure, as shown in Table 1. Other schemes are compared by the time of scalar multiplication. [19] requires 5, [14], [18], [15] and [20] require 6 and the second improved scheme proposed in this paper requires only 4. Based on the above analysis, the scheme proposed in this paper is more efficient.

Tab.2. Comparison between Similar Schemes in Computation Amount

Scheme	Aggregate signcryption	Aggregate signcryption verification	Total amount of computation	Security
Reference [17]	*nM*	*2P+2nM*	*2P+3nM*	Insecure
Reference [12]	*2nM*	*3P+2nM*	*3P+4nM*	Insecure
Reference [16]	*2nM*	*3P+2nM*	*3P+4nM*	Insecure
Reference [14]	*4nM*	*4P+2nM*	*4P+6nM*	Insecure
Reference [18]	*4nM*	*4P+2nM*	*4P+6nM*	Insecure
Reference [19]	*3nM*	*4P+2nM*	*4P+5nM*	Secure
Reference [15]	*4nM*	*4P+2nM*	*4P+6nM*	Secure
Reference [20]	*4nM*	*4P+2nM*	*4P+6nM*	Secure
This paper	*2nM*	*4P+2nM*	*4P+4nM*	Secure

7. Conclusion

In this paper, in the security model of certificateless aggregate signature, we construct a concrete attack method to prove the scheme proposed by He et al. is not unforgeable against Type

II attacks. By bonding the hash function with the public key information, we design two improved certificateless aggregate signature schemes with strong security. One is to break the composition relations between the secret value and other parameters, and the other is to make the composition relations of the definite value unsolvable. In the random oracle model, we prove the second improved scheme to be unforgeable and its security is equivalent to solving hard mathematical problems. Compared with the current similar schemes, the second improved scheme requires less computation. We take the aggregate signature for the smart grid as an example and introduce the improved scheme into the aggregate signature scenario. The test results show that the scheme can ensure security and at the same time achieve real-time data transmission.

Acknowledgement

This work is supported by Science and technology project of Henan Province in 2015, item No.1521022101933.

References

1. D. Bonen, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, 2003, Lecture Notes in Computer Science, vol. 2656, no. 1, pp. 416-432.
2. A. Lysyanskaya, S. Micali, L. Reyzin, H. Shacham, Sequential aggregate signatures from trapdoor permutations, 2004, International Conference on Advances in Cryptology-eurocrypt, vol. 3027, no. 2-6, pp. 74-90.
3. Z. Shao, Enhanced aggregate signatures from pairings, 2005, Information Security and Cryptology, vol. 3822, pp. 140-149.
4. A. Shamir, Identity-Based Cryptosystems and Signature Schemes, 1984, Lecture Notes in Computer Science, vol. 21, no. 2, pp. 47-53.
5. H.J. Yoon, J.H. Cheon, Y. Kim, Batch verifications with ID-based signatures, 2004, International Conference on Information Security and Cryptology, vol. 3506, pp. 233-248.
6. K.A. Shim, An ID-based aggregate signature scheme with constant pairing computations, 2010, IEEE Journal on Selected Areas in Communications, vol. 83, no. 10, pp. 1873-1880.
7. H. Wang, Z. Liu, Z. Liu, D.S. Wong, Identity-based aggregate signcryption in the standard model from multilinear maps, 2016, Frontiers of Computer Science, vol. 10, no. 4, pp. 741-754.

8. S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, 2003, International Conference on the Theory and Application of Cryptology, vol. 2894, no. 2, pp. 452-473.
9. Z. Gong, Y. Long, X. Hong, K. Chen, Two certificateless aggregate signatures from bilinear maps, 2007, Eighth Acis International Conference on Software Engineering, vol. 3, pp. 188-193.
10. X.Y. Yu, D.K. He, A new certificateless aggregate signature scheme, 2009, Computer Communications, vol. 32, no. 6, pp. 1079-1085.
11. L. Zhang, B. Qin, Q. Wu, F. Zhang, Efficient many-to-one authentication with certificateless aggregate signatures, 2010, Computer Networks, vol. 54, no. 14, pp. 2482-2491.
12. H. Xiong, Z. Guan, Z. Chen, F. Li, An efficient certificateless aggregate signature with constant pairing computations, 2013, Information Sciences, vol. 219, no. 10, pp. 225-235.
13. K.A. Shim, On the security of a certificateless aggregate signature scheme, 2011, IEEE Communications Letters, vol. 5, no. 3, pp. 358-367.
14. Y.L. Zhang, C.Y. Li, C.F. Wang, Y.J. Zhang, Security analysis and improvements of certificateless aggregate signature schemes, 2015, Dianzi Yu Xinxi Xuebao/journal of Electronics & Information Technology, vol. 37, no. 8, pp. 1994-1999.
15. A.W. An, D.L. Xia, Z.F. Yang, Security analysis on two certificateless aggregate signature schemes, 2016, Journal of Electronics & Information Technology, vol. 38, no. 10, pp. 2695-2700.
16. D. He, M. Tian, J. Chen, Insecurity of an efficient certificateless aggregate signature with constant pairing computations, 2014, Information Sciences An International Journal, vol. 268, no. 2, pp. 458-462.
17. Y. Xu, L.S. Huang, M.M. Tian, H. Zhong, J. Cui. A provably secure and compact certificateless aggregate signature scheme, 2016, Acta Electronica Sinica, vol. 44, no. 8, pp. 1845-1850.
18. H.Z. Du, M. Huang, Q. Wen, Efficient and provably-secure certificateless aggregate signature scheme, 2013, Tien Tzu Hsueh Pao/acta Electronica Sinica, vol. 41, no. 1, pp. 72-76.
19. M. Zhou, M. Zhang, C. Wang, B. Yang, CCLAS: A practical and compact certificateless aggregate signature with share extraction, 2014, International Journal of Network Security, vol. 16, no. 3, pp. 174-181.
20. H. Chen, S.M. Wei, C.J. Zhu, Y. Yang, Secure certificateless aggregate signature scheme, 2015, Ruan Jian Xue Bao/Journal of Software, vol. 26, no. 5, pp. 1173-1180.