Aggregate Signature without Pairing from Certificateless Cryptography

Lunzhi Deng[1,2], Yixian Yang[2,3], Yuling Chen[2], Xiong Wang[1]

1 School of Mathematical Science, Guizhou Normal University, China
[2] Guizhou University, Guizhou Provincial Key Laboratory of Public Big Data, China
[3] Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China
denglunzhi@163.com, yxyang@bupt.edu.cn, {61997525, 291056635}@qq.com

Abstract

In some real-world applications, many messages must be processed at the same time with low computational costs. In an aggregate signature scheme, anyone can combine *n* signatures on *n* messages from *n* users into a single signature, the resulting signature can convince a verifier that the *n* users indeed signed the *n* corresponding messages. All of the aggregate signature schemes currently known used bilinear pairings, however, the computational cost of the pairing is much higher than that of the exponentiation in a RSA group and that of the scalar multiplication over the elliptic curve group. In this paper, we propose a certificateless aggregate signature based on RSA and discrete logarithm (DL) problem, and prove the security in the random oracle model. To the best of author's knowledge, the scheme is the first certificateless aggregate signature scheme without pairing.

Keywords: Certificateless cryptography, Aggregate sign-ature, RSA, DL problem

1 Introduction

It is required that a large amount of data must be processed simultaneously in some real-world applications.

In a high-density traffic scenario, each roadside monitori-ng equipment needs to verify around 500-2000 messages. In a shopping spree day (For example, on November 11 in China), electronic payment platform needs to process about 200 transactions per minute. In some multicast, the root node needs to collect the data from leaf nodes, when lots of data be transmitted simultaneously, the root node will be swamped.

In traditional public key infrastructure (PKI), there is a trusted certification authority (CA) to issue digital certificate binding the user to his public key. So the certificate management problem arises. To solve the problem, Shamir [12] introduced identity-based public key cryptography. In this setting, there is a trusted private key generator (PKG) to generate private key for the user through his identity. However, which brings the key escrow problem. To solve the two problems, Al-Riyami et al. [1] put forward the notion of certificateless public key cryptography. In this notion, there is a semi-trusted key generation center (KGC), which generates partial private key for the user with respect to his identity. A user's full private key includes two parts: partial private key issued by KGC and a secret value chosen by himself.

1.1 Related Work

Al-Riyami and Paterson [1] presented the first certificateless signature (CLS) scheme, however, they did not give the formal proof of security. Yum and Lee [20] proposed a generic construction of CLS scheme. Huang et al. [11] showed a security drawback of the original scheme and proposed a secure one. Hu et al. [10] pointed out that Yum and Lee's construction is insecure and proposed a new one in the standard model. Xiong et al. [14] presented a security model for certificateless authenticated key agreement protocols and proposed a construction from bilinear pairings. Xiong [17] put forward a scalable certificateless remote authentication protocol, which achieves forward security and anonymity for wireless body area networks. He et al. [9] constructed a certificateless public auditing scheme for cloud-assisted wireless body area networks, which yields better performance over a previously proposed scheme. Xiong and zhang [18] presented a remote authentication protocol, which achieves client anonymity, non-repudiation, key escrow resistance, and revocability in the wireless body area networks. Zhang and Mao [24] constructed a CLS scheme based on RSA without bilinear pairing. He et al. [7] proposed a CLS scheme on the elliptic curve group, which does not use the bilinear pairing. Xiong et al. [15] proposed a certificateless threshold

*Corresponding Author: Lunzhi Deng; E-mail: denglunzhi@163.com
DOI: 10.3966/160792642018091905019

signature scheme, which is secure against the malicious-but-passive KGC attack in the standard model. Xiong et al. [19] put forward a pairing-free key insulated signature scheme based on certificate, which eliminates the costly pairing operations.

Boneh et al. [2] introduced the concept of aggregate signature. In this setting, given n signatures on n messages from n users, anyone can combine all of these signatures into a single signature. The resulting signature can convince a verifier that the n messages were signed by the n corresponding users.

Castro and Dahab [4] proposed the first certificateless aggregate signature (CLAS) scheme. Gong et al. [6] presented two CLAS schemes which are provably secure in a relatively weak model. Zhang and Zhang [23] constructed a CLAS scheme which is provably secure in a stronger model. Zhang et al. [21] proposed a CLAS scheme which requires a certain synchronization, i.e., all signers must share the same synchronized clocks to generate a aggregate signature. However, it is not easy to achieve synchronization in many mobile computing scenarios. Recently, Xiong et al. [16] presented a new CLAS scheme which requires constant pairing computations. Zhang et al. [22] gave the security analysis to Xiong et al.'s scheme [16] by showing four kinds of concrete attacks, and they put forward a secure CLAS scheme. Cheng et al. [5] pointed out that Xiong et al.'s scheme [16] is insecure even against "honest-but-curious" KGC attack, and they proposed an improved scheme.

1.2 Motivation and Our Contributions

The main goal of aggregate signature is reduce computation burden and storage burden. In most CLAS schemes, the number of using pairings grows linear with the number of signers. There is only two CLAS schemes [5, 21] which require constant pairing operations, independent of the number of signers. However, Zhang et al.'s scheme [21] requires all signers to share one-time-use state information to generate aggregate signature. In fact, it is not applicable in much real life.

In this paper, we constructed a new CLAS scheme and proved the security in the random oracle model, which has the following features:

- The scheme is secure in a strong security model. Namely, the super Type I/II adversaries can obtain the valid signatures for the replaced public key, without additional submission.
- The scheme does not need pairing operation.
- The scheme does not require synchronization for aggregating randomness, which makes it more suitable for practical applications.

2 Preliminaries

2.1 Elliptic Curve Group

Let E/F_p denote an elliptic curve E over a prime finite field F_p, defined by an equation:

$$y^2 = x^3 + ax + d \ (\mathrm{mod}\ p), a, d \in F_p$$
$$\text{And } 4a^3 + 27d^2 \neq 0 \ (\mathrm{mod}\ p).$$

The points on E/F_p together with an extra point O called the point at infinity form a group:

$$\Re = \{(x, y) : x, y \in F_p, E(x, y) = 0\} \cup \{O\}.$$

2.2 Complexity Assumption

Definition 1. Let $N = pq$, where p and q are two k-bit prime numbers. Let b be a random prime number, greater than 2^l for some fixed parameter l, such that $\gcd(b, \varphi(N)) = 1$. Given $Y \in Z_N^*$, RSA problem is to find $X \in Z_N^*$ such that $X^b = Y \bmod N$.

Definition 2. Let $\tau = (E, +)$, where E is an elliptic curve over a finite field F_p, $P \in E$ is a point having prime order $b = |E|/2$. Let $G = (P) \leq \tau$, given $xP \in G$, the discrete logarithm (DL) problem is to compute x.

2.3 System Model

A certificateless aggregate signature scheme consists of the following seven algorithms:

- Setup: This algorithm takes as input a security parameter k and returns the $params$ (system parameters) and msk (master secret key).
- Partial-Private-Key-Extract: This algorithm takes as input the $params$, msk and a user $ID_i \in \{0,1\}^*$, KGC generates the partial private key D_i for the user ID_i.
- Secret-Value-Set: This algorithm takes as input the $params$ and a user ID_i, the user selects a secret value t_i.
- User-Public-Key-Generate: This algorithm takes as input the $params$ and a user ID_i, the user outputs his public key P_i.
- Sign: This algorithm takes as input the $params$, signer's full private key (t_i, D_i) and a message m_i, then outputs the signature σ_i.
- Aggregate: This algorithm takes as input the $params$, and the signature σ_i on message m_i under the identity/public key $ID_i / P_i (i = 1, 2, \cdots, n)$, then outputs an aggregate signature σ on a message set

$M = \{m_1, \cdots, m_n\}$.

- Aggregation verify: This algorithm takes as input the $params$, an aggregate signature σ on a messages set $M = \{m_1, \cdots, m_n\}$ under an aggregating set $A = W \bigcup \{P_i : ID_i \in W\}$, where $W = \{ID_1, ID_2, \cdots, ID_n\}$ is a set of n identities. It outputs 1 if the aggregate signature is valid or 0 otherwise.

Definition 3. A certificateless aggregate signature (CLAS) scheme is unforgeable (UNF-CLAS) if the advantage of any polynomially bounded adversary is negligible in the following two games against Type I/II adversaries.

Game I. Now we illustrate the first game performed between a challenger ℓ and a Type I adversary A_1 for a CLAS scheme.

Initialization. ℓ runs the setup algorithm to generate the master secret key msk and public system parameters $params$. ℓ keeps msk secret and gives $params$ to A_1.

Query. A_1 performs a polynomially bounded number of queries.

- Hash functions query: A_1 can ask for the values of the hash functions for any input.
- User public key query: A_1 requests the public key of a user ID_i, ℓ returns the corresponding public key P_i.
- Partial private key query: A_1 requests the partial private key of a user ID_i, ℓ responds with the partial private key D_i.
- User public key replacement: A_1 supplies a new public key value P_i' with respect to a user ID_i. ℓ then replaces the current public key with the value P_i'.
- Secret value query: A_1 requests the secret value of a user ID_i, ℓ returns the secret value t_i. If a user's public key has been replaced, A_1 can not request the corresponding secret value.
- Signature query: A_1 submits the signer's identity/public key ID_i / P_i and a message m_i to the challenger. ℓ outputs a valid signature σ_i on the message m_i under the identity/public key ID_i / P_i.

Forge. A_1 outputs an aggregate signature σ^* on a message set $M^* = \{m_1, \cdots, m_n\}$ under an aggregating set $A^* = W^* \bigcup \{P_i : ID_i \in W^*\}$, where $W^* = \{ID_1, ID_2, \cdots, ID_n\}$ is a set of n identities. The adversary wins if the result of verify (σ^*, A^*, M^*) is the symbol 1 and the following conditions hold:

1. There exists at least a user $ID_j \in W^*$ whose partial private key was not queried by A_1. And the corresponding tuple (ID_j, P_j, m_j) has never been queried during the signature queries.

2. A_1 cannot query the secret value for any user if the corresponding public key has already been replaced.

The advantage of A_1 is defined as:

$$Adv_{A_1}^{UNF-CLAS} = \Pr[A_1 wins].$$

Game II. A Type II adversary A_2 plays the second game with a challenger ℓ as follows.

Initialization. A_2 runs the setup algorithm to obtain the master secret key msk and public system parameters $params$. A_2 then gives the $params$ and msk to ℓ.

Query. A_2 adaptively makes a polynomially bounded number of queries as those in Game I. Obviously, A_2 can compute the partial private key of any user by itself with the master secret key.

Forge. A_2 outputs an aggregate signature σ^* on a message set $M^* = \{m_1, \cdots, m_n\}$ under an aggregating set $A^* = W^* \bigcup \{P_i : ID_i \in W^*\}$,where $W^* = \{ID_1, ID_2, \cdots, ID_n\}$ is a set of n identities. The adversary wins if the result of verify (σ^*, A^*, M^*) is the symbol 1 and the following conditions hold:

1. There exists at least a user $ID_j \in W^*$ whose secret value was not queried and whose user public key was not replaced by A_2. And the corresponding tuple (ID_j, P_j, m_j) has never been queried during the signature queries.

2. A_2 cannot query the secret value for any user if the corresponding public key has already been replaced.

The advantage of A_2 is defined as:

$$Adv_{A_2}^{UNF-CLAS} = \Pr[A_2 wins].$$

3 Our Scheme

- Setup: Given the security parameter k, KGC generates two random k-bit prime numbers p and q, computes $N = pq$. For some fixed parameter l (for example $l = 200$), KGC chooses at random a prime number b satisfying $2^l < b < 2^{l+1}$ and gcd $(b, \varphi(N)) = 1$. Then it chooses a group G of prime order b as defined in Definition 2, a generator P of G and computes $a = b^{-1} \bmod \varphi(N)$. Furthermore, KGC chooses two cryptograph-ic hash functions:

$$H_0 : \{0,1\}^* \to Z_N^*,\ H_1 : \{0,1\}^* \to Z_b^*.$$

Finally, KGC outputs the set of public parameters:

$$params = \{N, b, G, P, H_0, H_1\}.$$

The master secret key is $msk = (p, q, a)$.

- Partial private key extract: For a user $ID_i \in \{0,1\}^*$, KGC computes $Q_i = H_0(ID_i)$ and sends $ID_i = Q_i^a$ to the user ID_i via a secure channel.
- Secret value set: The user ID_i randomly chooses $t_i \in Z_b^*$.
- User public key generate: The user ID_i computes his public key $P_i = t_i P$.
- Sign: For a message $m_i \in \{0,1\}^*$, the signer ID_i performs the following steps:
  1. Randomly selects $c_i \in Z_b^*$, $A_i \in Z_N^*$, computes $T_i = c_i P$, $B_i = A_i^b \bmod N$, $h_i = H_1(m_i, T_i, B_i, ID_i, P_i)$.
  2. Computes $r_i = c_i + t_i h_i \bmod b$, $R_i = A_i D_i^{h_i} \bmod N$.
  3. Outputs $\sigma_i = (T_i, B_i, r_i, R_i)$ as the signature.
- **Aggregate:** On receiving message-signature pairs $(m_i, \sigma_i = (T_i, B_i, r_i, R_i))$ under the identity/public ID_i / P_i for $i = 1, 2, \cdots, n$. Anyone can computes $r = \sum_{i=1}^{n} r_i$, $R = \prod_{i=1}^{n} R_i$ and outputs an aggregate signature $\sigma = ((T_1, B_1), \cdots, (T_n, B_n), r, R)$ on the message set $M = \{m_1, \cdots, m_n\}$.
- **Aggregation verify:** To verify the signature $\sigma = ((T_1, B_1), \cdots, (T_n, B_n), r, R)$ on the message set $M = \{m_1, \cdots, m_n\}$ under the aggregating set $A^* = W^* \bigcup \{P_i : ID_i \in W^*\}$, where $W^* = \{ID_1, ID_2, \cdots, ID_n\}$ is a set of n identities. The verifier performs the following steps:
  1. Computes $h_i = H_1(m_i, T_i, B_i, ID_i, P_i)$ for $i = 1, 2, \cdots, n$.
  2. Checks whether $rP = \sum_{i=1}^{n}(T_i + h_i P_i)$, $R^b = \prod_{i=1}^{n}(B_i Q_i^{h_i})$.

  If both of equations hold, accepts the signature. Otherwise, rejects.

4 Security

Theorem 1. The scheme is unforgeable against the super Type I adversary if the RSA problem is hard in randomly oracle model.

Proof. Suppose the challenger ℓ receives a random instance (Y, N, b) of the RSA problem and has to find an element $X \in Z_N^*$ such that $X^b = Y \bmod N$. ℓ runs A_1 as a subroutine and acts as A_1's challenger in the Game I.

Initialization. ℓ runs the setup program with the parameter k, then gives A_1 the system parameters $params = \{N, b, G, P, H_0, H_1\}$.

Queries. Without loss of generality, we assume that all the queries are distinct and A_1 will make $H_0(ID)$ query before a user ID_i is used in any other queries. A_1 sets several lists to store the queries and answers. All the lists are initially empty.

- H_0 queries: ℓ maintains the list L_0 of tuple (ID_i, V_i). When A_1 issues a query $H_0(ID_i)$, ℓ responds as follows:
  At the s^{th} H_0 query, ℓ sets $ID_s = ID^*$ and $H_0(ID^*) = Y$. For $i \neq s$, ℓ randomly picks a value $V_i \in Z_N^*$ and sets $H_0(ID_i) = V_i^b$, the query and the answer then are stored in the list L_0.
- H_1 queries: ℓ maintains the list L_1 of tuple (α_i, h_i). When A_1 issues a query $H_1(\alpha_i)$. ℓ randomly picks a value $h_i \in Z_b^*$, sets $H_1(\alpha_i) = h_i$ and adds (α_i, h_i) to the list L_1.
- User public key queries: ℓ maintains the list L_U of tuple (ID_i, t_i). When A_1 issues a user public key query for user ID_i, ℓ randomly picks a value $t_i \in Z_b^*$, returns $P_i = t_i P$ and adds (ID_i, t_i) to the list L_U.
- Partial private key queries: ℓ maintains the list L_D of tuple (ID_i, D_i). When A_1 issues a partial private key query for user ID_i. If $ID_i = ID^*$, ℓ fails and stops. Otherwise, ℓ finds (ID_i, V_i) in the list L_0, responds with $ID_i = V_i$ and adds (ID_i, V_i) to the list L_D.
- User public key replacement requests: ℓ maintains the list L_R of tuple (ID_i, P_i, P_i'). When A_1 issues a user public key replacement request for user ID_i with a new value P_i'. ℓ replaces the current public key P_i with P_i' and adds (ID_i, P_i, P_i') to the list L_R.
- Secret value queries: ℓ maintains the list L_E of tuple (ID_i, t_i). When A_1 issues a secret value query for the user ID_i. ℓ checks the list L_U, if (ID_i, t_i) is found in the list L_U, ℓ responds with t_i. Otherwise, ℓ randomly picks a new value $t_i \in Z_b^*$, responds with t_i and adds (ID_i, t_i) to the list L_E and L_U.
- Signature queries: When A_1 submits a signer's identity/public key ID_i / P_i and a message m_i to challenger. ℓ outputs a signature as follow:
  If $ID_i \neq ID^*$ and $ID_i \notin L_R$, ℓ gives a signature by calling the signing algorithm. Otherwise, ℓ does as follow:
  1. Randomly selects $R_i \in Z_N^*$ and $r_i, h_i \in Z_b^*$.
  2. Computes $T_i = r_i P - h_i P_i$, $B_i = R_i^b Q_i^{-h_i}$.
  3. Adds $h_i = H_1(m_i, T_i, B_i, ID_i, P_i)$ to the list L_1. If collision occurs, repeats the steps 1-3.
  4. Outputs $\sigma_i = (T_i, B_i, r_i, R_i)$ as the signature.

Forge. A_1 outputs a forged signature $\sigma^* = ((T_1, B_1), \cdots,$

$(T_n, B_n)), r, R)$ on the message set $M^* = \{m_1, \cdots, m_n\}$ under the aggregating set $A^* = W^* \bigcup \{P_i : ID_i \in W^*\}$, where $W^* = \{ID_1, ID_2, \cdots, ID_n\}$ is a set of n identities, and fulfills the following conditions:

1. There exists at least a user $ID_j \in W^*$ whose partial private key was not queried by A_1. And the corresponding tuple (ID_j, P_j, m_j) has never been queried during the signature queries.

2. A_1 cannot query the secret value for any user if the corresponding public key has already been replaced.

Solve RSA problem. Note that $r = \sum_{i=1}^{n} r_i$, $R = \prod_{i=1}^{n} R_i$, the tuple (T_i, B_i, r_i, R_i) is the signature on the message m_i under the identity/public key ID_i / P_i for $i = 1, 2, \cdots, n$. And there exists at least a user $ID_j \in W^*$ whose partial private key was not queried by A_1. Which implies that (T_j, B_j, r_j, R_j) is a forge signature on the message m_j. Using general forking lemma [3], after replaying A_1 with the same random tape but different h_j returned by H_1 query of the forged message m_j, ℓ gets two aggregate signatures with at least probability $\varepsilon \cdot (\frac{\varepsilon}{q_{H_1}} - \frac{1}{b})$:

$$((T_1, B_1), \cdots, (T_n, B_n), r, R), ((T_1, B_1), \cdots, (T_n, B_n), r', R'),$$

where $R = \prod_{i=1}^{n} R_i$, $R' = \prod_{i=1}^{n} R'_i$, $R_j \neq R'_j$, $R_i = R'_i$ for $i \neq j$ .If $ID_j = ID^*$, then $R_j = A_j Y^{ah_j}$ and $R'_j = A_j Y^{ah'_j}$. It follows that $(R'R^{-1})^b = Y^{h'_j - h_j} \bmod N$. Since $h_j, h'_j \in Z_b^*$, then $\left| h'_j - h_j \right| < b$. By the element b is a prime number, then $\gcd(b, h'_j - h_j) = 1$. This means that there exist two integers c and d such that $cb + d(h'_j - h_j) = 1$. Finally, ℓ solves the RSA problem by computing:

$X = (R'R^{-1})^d Y^c \bmod N$. In effect, $X^b = (R'R^{-1})^{bd} Y^{bc} = Y^{d(h'_j - h_j)} Y^{bc} = Y^{cb + d(h'_j - h_j)} = Y$.

Probability. Let $q_{H_i} (i = 0, 1)$ and q_D be the numbers of $H_i (i = 0, 1)$ queries and partial private key queries.

The probability that ℓ does not fail during the queries is $\frac{q_{H_0} - q_D}{q_{H_0}}$. The probability that $ID_j = ID^*$ is $\frac{1}{q_{H_0} - q_D}$. So the combined probability is $\frac{q_{H_0} - q_D}{q_{H_0}} \cdot \frac{1}{q_{H_0} - q_D} = \frac{1}{q_{H_0}}$.

Therefore, if the adversary A_1 can win the EUF-CLAS Game I with advantage ε, then ℓ can solve the RSA problem with the probability $\frac{\varepsilon}{q_{H_0}} (\frac{\varepsilon}{q_{H_1}} - \frac{1}{b})$.

Theorem 2. The scheme is unforgeable against the super Type II adversary if the DL problem is hard in randomly oracle model.

Proof. Suppose the challenger ℓ receives a random instance (xP, P) of the DL problem and has to compute the value of x. ℓ runs A_2 as a subroutine and acts as A_2's challenger in the game II.

Initialization. A_2 runs the setup program with the parameter k to obtain the system parameters $params = \{N, b, G, P, H_0, H_1\}$ and master secret key $msk = (p, q, a)$. A_2 then gives ℓ the $params$ and msk.

Queries. Without loss of generality, we assume that all the queries are distinct and A_2 will ask for the user public key before a user ID_i is used in any other queries. A_2 sets several lists to store the queries and answers. All the lists are initially empty.

- User public key queries: ℓ maintains the list L_U of tuple (ID_i, t_i). When A_2 issues a user public key query for the user ID_i, ℓ responds as follows:
  At the s^{th} query, ℓ sets $ID_s = ID^*$, $P_s = P^* = xP$. For $i \neq s$, ℓ randomly picks a value $t_i \in Z_b^*$, returns $P_i = t_i P$ and adds (ID_i, t_i) to the list L_U.
- H_0 queries: ℓ maintains the list L_0 of tuple (ID_i, Q_i). When A_2 issues a query $H_0(ID_i)$, ℓ randomly picks a value $Q_i \in Z_N^*$, sets $H_0(ID_i) = Q_i$ and adds (ID_i, Q_i) to the list L_0.
- H_1 queries: Same as that in the proof of Theorem 1.
- Partial private key queries: Since A_2 knows master secret key $msk = (p, q, a)$, he can compute the partial private key for any user by himself. Hence A_2 does not need issue partial private key query.
- User public key replacement requests: Same as that in the proof of Theorem 1.
- Secret value queries: ℓ maintains the list L_E of tuple (ID_i, t_i). When A_2 issues a secret value query for the user ID_i. If $ID_i = ID^*$, ℓ fails and stops. Otherwise, ℓ finds (ID_i, t_i) in the list L_U, responds with t_i and adds (ID_i, t_i) to the list L_E.
- Signature queries: Same as that in the proof of Theorem 1.

Forge. A_2 outputs a forged signature $\sigma^* = ((T_1, B_1), \cdots,$ $(T_n, B_n)), r, R)$ on the message set $M^* = \{m_1, \cdots, m_n\}$ under the aggregating set $A^* = W^* \bigcup \{P_i : ID_i \in W^*\}$, where $W^* = \{ID_1, ID_2, \cdots, ID_n\}$ is a set of n identities, and fulfills the following conditions:

1. There exists at least a user $ID_j \in W^*$ such that his secret value was not queried and his user public key was not replaced by A_2. And the corresponding tuple

(ID_j, P_j, m_j) has never been queried during the signature queries.

2. A_2 cannot query the secret value for any user if the corresponding public key has already been replaced.

Solve DL problem. Note that $r = \sum_{i=1}^{n} r_i$, $R = \prod_{i=1}^{n} R_i$, the tuple (T_i, B_i, r_i, R_i) is the signature on the message m_i under the identity/public key ID_i / P_i for $i = 1, 2, \cdots, n$. And there exists at least a user $ID_j \in W^*$ such that his secret value was not queried and his user public key was not replaced by A_2. Which implies that (T_j, B_j, r_j, R_j) is a forge signature on the message m_j.Using general forking lemma [3], after replaying A_2 with the same random tape but different h_j returned by H_1 query of the forged message m_j, ℓ gets two aggregate signatures with at least probability $\varepsilon \cdot (\frac{\varepsilon}{q_{H_1}} - \frac{1}{b})$:

$$((T_1, B_1), \cdots, (T_n, B_n), r, R), ((T_1, B_1), \cdots, (T_n, B_n), r', R'),$$

where $r = \sum_{i=1}^{n} r_i$, $r' = \sum_{i=1}^{n} r'_i$, $r_j \neq r'_j$ and $r_i = r'_i$ for $i \neq j$.If $ID_j = ID^*$, then $r_j = c_j + xh_j$ and $r'_j = c_j + xh'_j$, ℓ can solve DL problem by computing $x = (h_j - h'_j)^{-1} \cdot (r - r') \bmod b$.

Probability. Let $q_{H_i} (i = 0,1)$, q_U, q_R and q_E be the numbers of $H_i (i = 0,1)$ queries, user public key replacement requests, user public key queries and secret value queries.

Without loss of generality, we may assume that $L_E \cap L_R = \Phi$.

The probability that ℓ does not fail during the queries is $\frac{q_U - q_E}{q_U}$. The probability that $ID_j = ID^*$ is $\frac{1}{q_U - q_E - q_R}$. So the combined probability is $\frac{q_U - q_E}{q_U} \cdot \frac{1}{q_U - q_E - q_R} \geq \frac{1}{q_U}$.

Therefore, if the adversary A_2 can win the EUF-CLAS Game II with advantage ε, then ℓ can solve the DL problem with the probability $\frac{\varepsilon}{q_U}(\frac{\varepsilon}{q_{H_1}} - \frac{1}{b})$.

5 Efficiency

In this section, we compare the performance of our scheme with several CLAS schemes in Table 2, we define some notations as follows.

P: a pairing operation.

E_G: a pairing-based scalar multiplication operation.

E_S: a scalar multiplication operation.

E_N: a modular exponent operation in Z_N.

By using Windows XP operation system and PIV 3-GHZ processor with 512-MB memory. He et al. [8] obtained the running time for cryptographic operations. To achieve 1024-bit RSA level security, Tate pairing was used, which is defined on a supersingular curve $E / F_p : y^2 = x^3 + x$ with embedding degree 2, where q is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p is a 512-bit prime satisfying $p + 1 = 12qr$. To achieve the same security level, the parameter secp160r1 [13] was used too, where $p = 2^{160} - 2^{31} - 1$. The running times are listed in Table 1.

Table 1. Cryptographic operation time (in milliseconds)

P	E_G	E_S	E_N
20.04	6.38	2.21	5.31

We use a simple method to evaluate the computational cost. For example, Zhang and Zhang's scheme [23] requires $3n$ pairing-based scalar multiplication operations and $n + 3$ pairing operations. So the resulting computation time is $6.38 \times 3n + 20.04 \times (n + 3) = 39.18n + 60.12$. In order to facilitate the comparison, we let $n = 100$, then the computation time is $39.18 \times 100 + 60.12 = 3978.12$. Based on the above parameter and ways, the detailed comparison results of several different CLAS schemes are illustrated in Table 2.

Table 2. Comparison of several CLAS schemes

Scheme	Sign	Verify	Execution time/(n=100)
Castro and Dahab's scheme [4]	2n E_G	(2n+1)P+n E_G	59.22n+20.04/5942.04
Cheng et al.'s scheme [5]	4n E_G	3P+2n E_G	38.28n+60.12/3888.12
Gong et al.'s scheme 1 [6]	2n E_G	(2n+1)P	52.84n+20.04/5304.04
Gong et al.'s scheme 2 [6]	3n E_G	(n+2)P+n E_G	45.56n+40.08/4596.08
Zhang et al.'s scheme [22]	nP+2n E_G	2nP+(3n+1) E_G	92.02n+6.38/9208.38
Zhang and Zhang's scheme [23]	3n E_G	(n+3)P	39.18n+60.12/3978.12
Our scheme	n E_S +2n E_N	(n+1)(E_S+ E_N)	20.35n+7.52/2042.52

6 Conclusion

All of the known aggregate signature schemes used bilinear pairings. Some good results have been achieved in speeding up the computation of pairing in recent years, however, the computational cost of the pairing is much higher than of the exponentiation in a RSA group and that of the scalar multiplication over the elliptic curve group.

So it is still interesting to design aggregate signature scheme without pairing. In this paper, a new certificateless aggregate signature scheme based on RSA and discrete logarithm problem was proposed, which is unforgeable against type I/II adversaries in the random oracle model. To the best of author's knowledge, the scheme is the first CLAS scheme without pairing and which is more efficient than previous ones in computation. Due to the good properties of the scheme, it should be useful for practical applications.

Acknowledgments

The author is grateful to the anonymous referees for their helpful comments and insightful suggestions. This research is supported by the National Natural Science Foundation of China under Grants 61562012, the Innovation Group Major Research Projects of Department of Education of Guizhou Province under Grant No. KY [2016] 026.

References

[1] S. S. Al-Riyami, K. G. Paterson, Certificateless Public Key Cryptography, in: C.S. Laih (Ed.), *Advances in Cryptology-Asiacrypt 2003, Lecture Notes in Computer Science*, Vol. 2894, Springer, Berlin, Heidelberg, 2003, pp. 452-473.

[2] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, in: E. Biham (Ed.), *Advances in Cryptology —EUROCRYPT'03, Lecture Notes in Computer Science, Vol 2656*, Springer, Berlin, Heidelberg, 2003, pp. 416-432.

[3] M. Bellare, G. Neven, Multi-Signatures in the Plain Public Key Model and a General Forking Lemma, *CCS'06 Proceedings of the 13th ACM Conference on Computer and Communications Security*, Alexandria, VA, 2006, pp. 390-399.

[4] R. Castro, R. Dahab, *Efficient Certificateless Signatures Suitable for Aggregation*, http://eprint.iacr.org/2007/454.

[5] L. Cheng, Q. Wen, Z. Jin, H. Zhang, L. Zhou, Cryptanalysis and Improvement of a Certificateless Aggregate Signature Scheme, *Information Sciences*, Vol. 295, pp. 337-346, February, 2015.

[6] Z. Gong, Y. Long, X. Hong, K. Chen, Two Certificateless Aggregate Signatures from Bilinear Maps, *IEEE Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, Vol. 3, Qingdao, China, 2007, pp. 188-193.

[7] D. He, J. Chen, R. Zhang, An Efficient and Provably Secure Certificateless Signature Scheme without Bilinear Pairings, *International Journal Of Communication Systems*, Vol. 25, No. 11, pp. 1432-1442, November, 2012.

[8] D. He, J. Chen, J. Hu, An ID-based Proxy Signature Schemes without Bilinear Pairings, *Annals of Telecommunications-Annales des Telecommuni: Cations*, Vol. 66, No. 11-12, pp. 657-662, December, 2011.

[9] D. He, S. Zeadally, L. Wu, Certificateless Public Auditing Scheme for Cloud-Assisted Wireless Body Area Networks, *IEEE Systems Journal*, Vol. 12, No. 1, pp. 64-73, March, 2018. doi:10.1109/JSYST.2015.2428620

[10] B. C. Hu, D. S. Wong, Z. Zhang, X. Deng, Key Replacement Attack against a Generic Construction of Certificateless Signature, *Proceedings of the 11th Australasian Conference on Information Security and Privacy (ACISP 2006),* Melbourne, Australia, 2006, pp. 235-246.

[11] X. Huang, W. Susilo, Y. Mu, F. Zhang, On the Security of Certificateless Signature Schemes from Asiacrypt 2003, *Proceedings of the 4th International Conference on Cryptology and Network Security (CANS 2005)*, Xiamen, China, 2005, pp. 13-25.

[12] A. Shamir, Identity-based Cryptosystems and Signature Schemes, *In: Advances in Cryptology-Crypto 1984,* Santa Barbara, CA, 1984, pp. 47-53.

[13] The Certicom Corporation, SEC2: Recommended Elliptic Curve Domain Parameters, http://www.secg.org/collateral/sec2-final.pdf.

[14] H. Xiong, Z. Chen, F. Li, Provably Secure and Efficient Certificateless Authenticated Tripartite Key Agreement Protocol, *Mathematical and Computer Modelling*, Vol. 55, No. 3-4, pp. 1213-1221, February, 2012.

[15] H. Xiong, F. Li, Z, Qin, Certificateless Threshold Signature Secure in the Standard Model, *Information Sciences*, Vol. 237, pp. 73-81, July, 2013.

[16] H. Xiong, Z. Guan, Z. Chen, F. Li, An Efficient Certificateless Aggregate Signature with Constant Pairing Computations, *Information Sciences*, Vol. 219, pp. 225-235, January, 2013.

[17] H. Xiong, Cost-Effective Scalable and Anonymous Certificateless Remote Authentication Protocol, *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 12, pp. 2327-2339, December, 2014.

[18] H. Xiong, Z. Qin, Revocable and Scalable Certificateless Remote Authentication Protocol with Anonymity for Wireless Body Area Networks, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 7, pp. 1442-1455, July, 2015.

[19] H. Xiong, S. Wu, J. Geng, E. Ahene, S. Wu, Z. Qin, A Pairing-free Key-insulated Certificate-based Signature Scheme with Provable Security, *KSII Transactions on Internet and Information Systems*, Vol. 9, No. 3, pp. 1246-1259, March, 2015.

[20] D. H. Yum, P. J. Lee, Generic Construction of Certificateless

Signature, *Proceedings of Information Security and Privacy: 9th Australas-ian Conference (ACISP' 2004)*, Sydney, Australia, 2004, pp. 200-211.

[21] L. Zhang, B. Qin, Q. Wu, F. Zhang, Efficient Many-to-one Authentication with Certificateless Aggregate Signatures, *Computer Networks*, Vol. 54, No. 14, pp. 2482-2491, October, 2010.

[22] F. Zhang, L. Shen, G. Wu, Notes on the Security of Certificateless Aggregate Signature Schemes, *Information Sciences*, Vol. 287, pp. 32-37, December, 2014.

[23] L. Zhang, F. Zhang, A New Certificateless Aggregate Signature Scheme, *Computer Communications*, Vol. 32, No. 6, pp. 1079-1085, April, 2009.

[24] J. Zhang, J. Mao, An Efficient RSA-based Certificateless Signature Scheme, *Journal of Systems and Software*, Vol. 85, No. 3, pp. 638-642, March, 2012.

Biographies

Lunzhi Deng received his B.S. from Guizhou Normal University, Guiyang, PR China, in 2002; M.S. from Guizhou Normal University, Guiyang, PR China, in 2008; and Ph.D. from Xiamen University, Xiamen, PR China, in 2012. He is now a professor in the School of Mathematics and Computer Science, Guizhou Normal University, Guiyang, PR China. His recent research interests include algebra and information safety.

Yixian Yang is a professor of Beijing University of Posts and Telecommunications, Beijing, PR China. He is a member of the China Science and Technology Commission of the Ministry of Education, has been published more than 300 papers in the IEEE Trans. On AES, IEEE Trans.On Comm., IEEE Trans. On EMC and Discrete Applied Mathematics and other international most authoritative academic journals.

Yuling Chen received her B.S. from Taishan University, Taian, PR China, in 2006; MS from Guizhou University, Guiyang, PR China, in 2009. She is now a associate professor in Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, PR China. Her recent research interests include cryptography and information safety.

Xiong Wang received his B.S. from Xianyang Normal University, Xianyang, PR China, in 2014; M.S. from Guizhou Normal University, Guiyang, PR China, in 2017. His recent research interests include cryptography and information safety.