A Revocable Certificateless Aggregate Signature Scheme with Enhanced Security

Fuxiao Zhou[1], Yanping Li[1], and Changlu Lin[2]
(Corresponding author: Yanping Li)
School of Mathematics and information Science, Shaanxi Normal University, Xi'an, Shaanxi 710119, China[1]
Fujian Normal University, Fuzhou, Fujian 350007, China[2]
(Email: lyp@snnu.edu.cn)
(Received Mar. 14, 2019; Revised and Accepted Sept. 3, 2019; First Online Sept. 21, 2019)

Abstract

In certificateless public key cryptosystem, a tough problem is how to revoke a user when the user's private key is compromised or expired. So the revocable certificateless schemes come into being. Certificateless aggregate signature (CLAS) is an efficient way to verify a large amount of signatures from different users simultaneously. However, none of CLAS schemes considers the user revocation currently. In this paper, we firstly demonstrate that an efficient certificateless aggregate signature (abbreviated to ECLAS) scheme proposed by Kang *et al.* is vulnerable to forged signature attack from the type II adversary by a concrete example, although they claimed that their scheme is existentially unforgeable against the adaptively chosen-message attacks. Furthermore, based on the ECLAS scheme and the revocable idea, we proposed a revocable certificateless aggregate signature scheme, which was proved to be existentially unforgeable against adaptive chosen-messages attacks under the hardness assumption of computational Diffie-Hellman problem. As far as we know, this is the first revocable CLAS scheme. Finally, numerical analyses and performance comparisons show our scheme saves computational cost, communication bandwidth and storage space than some related schemes.

Keywords: Certificateless Aggregate Signature; Cryptography; Existentially Unforgeable; Revocable

1 Introduction

In traditional public key infrastructure (PKI), a trusted entity called certificate authority (CA) often issues certificates to users by binding users true identities with their public keys. However, certificate management and authentication are quite complicated and expensive, which bring a heavy burden to CA in real-life. In 1984, Shamir first proposed the identity-based public key cryptosystem (ID-PKC) [13] to overcome the heavy certificate management and deep dependence on CA in PKI. The motive of this proposal was to choose the unique identity information such as social security number, telephone number of each party as user's public key. However, it needs a trusted third party named private key generator (PKG) to generate the private key for user. Hence the PKG possesses the private key of each user and can sign messages on behalf of any user at will, which makes the key escrow being the biggest criticism of the ID-PKC system. In order to eliminate the above problems in PKI and ID-PKC, Al-Riyami and Paterson introduced a new paradigm named certificateless public key cryptosystem (CL-PKC) in 2003 [1]. In a CL-PKC, a user's private key consists of two components: a partial private key issued by key generation center (KGC) and a secret value selected by the user. Since the KGC has no access to the user's entire private key, CL-PKC is not subject to the key escrow problem [14]. Additionally, CL-PKC also does not need certificates to authenticate public keys. Therefore, the CL-PKC is currently recognized as a promising public key cryptosystem.

Unfortunately, CL-PKC has the user revocation problem. It is well known that to revoke a user in PKC when the user's private key is compromised or expired is very cumbersome [2, 20]. The same problem inevitably exists in the CL-PKC and it gets more complex because the user's ID (*i.e.*, the public key) cannot change frequently. A previous revocation solution in CL-PKC was to use an on-line mediator called security mediator (SEM) [22]. In this kind of mechanism, the KGC divides a user's partial private key into two parts: One is delivered to the user and the other is delivered to the SEM. All these communications are conducted via secure channels, which greatly increase the communication costs. Later, Shen *et al.* [15] and Tsai *et al.* [17] successively presented two revocable certificateless encryption schemes. In both schemes, a user's private key consists of three parts: an initial partial private key, a time key and a secret value. The KGC controls the revocation of users by updating of the time key. It is noteworthy that the time key is renewed periodically over public channels by the KGC, which reduces

the need for secure channels and saves communication costs. Inspired by the idea used in [18], Sun *et al.* proposed the first revocable certificateless signature (RCLS) scheme [16], and soon after Zhang *et al.* put forward another improved RCLS scheme [24]. In both above RCLS schemes, the KGC generates a partial private key and a time key, where the time key is updated periodically. And the KGC just stops issuing the new time update key to revoke a user. Without the update time key, the user cannot sign a valid signature.

The notion of aggregate signature was introduced by Boneh *et al.* in 2003 [3]. Its primary focus is to aggregate n signatures on n messages from n users into a short signature, so a verifier can convince that the validity of n signatures by verifying the correctness of aggregate signature. Therefore, aggregate signatures greatly reduce the storage space, communication bandwidth and computational cost in verification and become a research hot spot. Combined with the prominent advantages of CL-PKC and aggregate signatures, a large number of certificateless aggregate signatures (CLAS) are put forward for various application scenarios [4–7, 10–12, 19, 21, 23, 25]. Gong *et al.* proposed two CLAS schemes to realize the aggregate signature scheme in CL-PKC [6]. However, Zhang *et al.* pointed out their schemes are insecure and proposed a new scheme and refined the security models [23]. In 2013, Xiong *et al.* put forward a CLAS scheme in [21] and claimed the scheme is more efficient than others. However, it was pointed out that an adversary could forge a legal signature for any message [7]. Li *et al.* proposed a novel and provably secure certificateless aggregate signature scheme in [11] and Nie *et al.* put forward a novel and efficient CLAS scheme [12]. Unfortunately, Nie's scheme was later proved that an adversary could forge any signer's signature on any message by obtaining a pair of message and its corresponding signature. Cui *et al.* proposed a CLAS scheme without pairings based on the elliptic curve cryptosystem [5]. Zhou *et al.* put forward a practical and compact certificateless aggregate signature with share extraction [25]. However, Chen *et al.* showed their scheme is in fact insecure against a type I adversary [4]. Wu *et al.* pointed out the CLAS scheme in [10] is vulnerable to signature forgery and proposed a new CLAS to fix the security flaws [19]. Recently, Kang *et al.* proposed an efficient CLAS scheme (ECLAS for short) and claimed their ECLAS scheme is existentially unforgeable against the adaptively chosen-message attacks [9]. In this article, we prove that the ECLAS scheme in [9] cannot satisfy the security they claimed by presenting a concrete example. As far as we know, there are no aggregation signature schemes with users' revocation at present. Therefore, we try to propose a revocable certificateless aggregate signature (RCLAS) scheme in this paper just in order to provide a secure revocation mechanism for CL-PKC-based aggregation signatures.

Our Contributions: In this paper, we propose a revocable certificateless aggregate signature (RCLAS) scheme. The contributions are summarized as follows:

1) Firstly, we demonstrate that the ECLAS scheme in [9] is not secure since it cannot resist the type II adversary. Specifically speaking, any type II adversary A_2 could forge any signer's signature on any message based on a valid signature, so that A_2 can forge a valid aggregate signature. At the same time, we analyze the reasons why the scheme is vulnerable to such attack and give the design principle of resisting this kind of attack;

2) Secondly, an improved scheme, namely, a revocable certificateless aggregate signature (RCLAS) scheme is proposed, which can revoke the user flexibly to meet the actual scenarios by using the time key. Then our RCLAS scheme is proven to be secure in the random oracle model under the hardness assumption of computational Diffie-Hellman problem (CDHP);

3) Thirdly, numerical analyses and performance comparisons demonstrate that our scheme has better performance than some existing schemes in [9, 16, 19]. Specifically, the length of aggregate signature in our RCLAS scheme only consists of two elements in G_1 which is far shorter than the aggregate signature in [19] and is independent of the number of signatures being aggregated. Additionally, the verification costs in the RCLAS scheme are relatively small.

The rest of this article is arranged as follows. In Section 2, some essential preliminaries are given. In Section 3, the ECLAS scheme is briefly reviewed and a specific attack on the ECLAS scheme is given. Our improved RCLAS scheme and its security proof are presented in Section 4 and Section 5, respectively. In Section 6, the performance of our scheme compares with some existing schemes. Finally, Section 7 concludes our paper.

2 Preliminaries

In this section, we introduce some necessary knowledge required in this paper.

2.1 Bilinear Pairing

Let G_1 be a cyclic additive group of prime order q and G_2 be a cyclic multiplicative group of the same order, P be a generator of G_1, $e : G_1 \times G_1 \to G_2$ be a bilinear map if it satisfies the following properties [8]:

1) Blinearity: $e(aP, bQ) = e(P, Q)^{ab}$,where $P, Q \in G_1$ and $a, b \in Z_q^*$;

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

3) Computability: It is efficient to compute $e(P, Q)$ for all $P, Q \in G_1$.

Definition 1. *Computational Diffie-Hellman problem (CDHP): Let G_1 be a cyclic additive group of prime order*

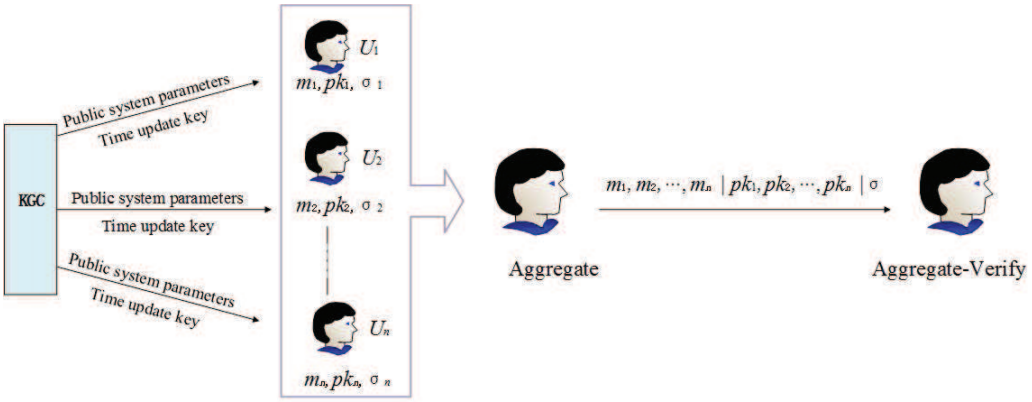

Figure 1: The proposed scheme

q and P be a generator of G_1. Given the elements P, aP and bP for the unknown $a, b \in Z_q^*$, it is hard to find abP.

2.2 Framework of a RCLAS Scheme

Generally, there are a KGC, n users and a signature aggregator in a RCLAS scheme, which consists of eight algorithms: Setup, Public-Key-Extract, Partial-Private-Key-Extract, Time-Key-Update, Private-Key-Extract, Sign, Aggregate, Aggregate Verify. The details of these algorithms will be described in Section 4 and not be repeated here because of the limit length. In the following, the system architecture of our RCLAS is given in Figure 1.

2.3 Security Model

In traditional RCLS schemes, three types of adversaries are considered. The type I adversary A_1 cannot obtain the master secret key msk, but can replace any user's public key, which describes an external adversary who did not know the msk.The type II adversary A_2 who has access to the msk but is unable to replace the user's public key, which depicts an internal adversary, such as the dishonest KGC. The type III adversary A_3 is used to describe the revoked malicious signers, who holds his/her partial private key and can replace other user's public key, but A_3 have no access to the msk and will no longer be issued the current time update key. Up to now, none of the existing revocable certificateless signature schemes can resist the collusion attack of KGC and revoked users. Hence, such attack is not considered in this article.

Definition 2. *The security model for the RCLAS scheme is defined by the following three games (**Game 1**, **Game 2** and **Game 3**) between a challenger C and three types of adversaries, respectively. The game details are given as follows.*

Game 1: A type I adversary A_1 interacts with the challenger C in this game. There are three phases in the game: **Setup**, **Queries**, **Forgery**.

Setup: C performs the setup algorithm that takes a security parameter l as input to obtain the master secret key msk and the system parameters $params$. Then C sends $params$ to A_1 while holds msk secret.

Queries: A_1 can perform a polynomially bounded number of the following types of queries in an adaptive way as follows:

- Hash queries: A_1 can request the hash values of any messages, C returns the corresponding results to A_1.
- Partial-Private-Key-Extract queries: When A_1 submits a private key query on an identity ID_i of a user U_i, C returns the corresponding private key D_i to A_1 by running the Partial-Key-Extract algorithm.
- Time-Key-Update queries: When A_1 submits a private key query, C runs the Time-Key-Extract algorithm to generate user's time key T_i and sends it to A_1.
- Public-key-Extract queries: When A_1 requests the public key of a user U_i with identity ID_i, C returns the corresponding public key pk_i by running the Public-key-Extract algorithm.
- Secret-Value-Extract queries: When A_1 requests the secret value of a user U_i with identity ID_i, C returns the corresponding secret value x_i by running Secret-Value-Extract algorithm. But note that A_1 is not allowed to ask for the secret value of a replaced public key.
- Public-key-Replacement queries: For any user U_i with identity ID_i, A_1 can select a new public key for the user U_i. C will record this replacement.
- Sign queries: When A_1 requests a user's signature query on a message m_i, C responds with the corresponding signature by running the Sign algorithm.

Forgery: The adversary A_1 outputs a tuple $(m^*, ID^*, t^*, w, \sigma^*)$ in which $t^* = (t_1^*, t_2^*, \cdots, t_n^*)$ was the expiration times, $m^* = (m_1^*, m_2^*, \cdots, m_n^*)$, $ID^* =$

$(ID_1^*, ID_2^*, \cdots, ID_n^*)$, w is a state information and σ^* is an aggregate signature. We say that A_1 wins **Game 1** if and only if:

1) σ^* is a valid aggregate signature on messages m^*.
2) At least one of the identities, without loss of generality, say $ID_1^* \in ID^*$ has never submitted during the Partial-Private-Key-Extract queries.
3) (m^*, ID^*, t^*, w) has never been submitted to the Sign queries.

Game 2: A type II adversary A_2 interacts with the challenger C in this game. There are three phases in the game: **Setup**, **Queries**, **Forgery**.

Setup: C performs the setup algorithm that takes a security parameter l as input to obtain the master secret key msk and the system parameters $params$. Then C sends the $params$ and msk to adversary A_2.

Queries: The adversary A_2 can perform a polynomially bounded number of queries as in **Game 1** in an adaptive way. Note that A_2 can make the Hash queries, Public-key-Extract queries, Secret-Value-Extract queries and Sign queries. But A_2 has no need to request the Partial-Private-Key-Extract queries and Time-Key-Update queries since the internal adversary A_2 who has access to the master secret key msk.

Forgery: The adversary A_2 outputs a tuple $(m^*, ID^*, t^*, w, \sigma^*)$ in which $t^* = (t_1^*, t_2^*, \cdots, t_n^*)$ was the expiration times,$m^* = (m_1^*, m_2^*, \cdots, m_n^*)$, $ID^* = (ID_1^*, ID_2^*, \cdots, ID_n^*)$, w is a state information and σ^* is an aggregate signature. We say that A_2 wins **Game 2** if and only if:

1) σ^* is a valid aggregate signature on messages m^*.
2) At least one of the identities, without loss of generality, say $ID_1^* \in ID^*$ has never submitted during the Secret-Value-Extract queries.
3) (m^*, ID^*, t^*, w) has never been submitted to the Sign queries.

Game 3: A type III adversary A_3 interacts with the challenger C in this game. There are three phases in the game: **Setup**, **Queries**, **Forgery**. It is worth noting that **Game 3** is very similar to **Game 1**, except that the conditions for the adversary to win the game are different. Details are given in the following.

Forgery: The adversary A_3 outputs a tuple $(m^*, ID^*, t^*, w, \sigma^*)$ in which $t^* = (t_1^*, t_2^*, \cdots, t_n^*)$ was the expiration times, $m^* = (m_1^*, m_2^*, \cdots, m_n^*)$, $ID^* = (ID_1^*, ID_2^*, \cdots, ID_n^*)$, w is a state information and σ^* is an aggregate signature. We say that A_3 wins **Game 3** if and only if:

1) σ^* is a valid aggregate signature on messages m^*.
2) At least one of the identities, without loss of generality, say $(ID_1^*, t_1^*) \in (ID^*, t^*)$ has never submitted during the Time-Key-Update queries.
3) (m^*, ID^*, t^*, w) has never been submitted to the Sign queries.

Definition 3. *A revocable certificateless aggregate signature scheme is said to be existential unforgeable against adaptive chosen-message attacks if no a probabilistic polynomial-time (PPT) adversary has non-negligible advantage in the above games (**Game 1**, **Game 2** and **Game 3**).*

3 Review and Security Analysis of the ECLAS Scheme

In this section, we briefly introduce the ECLAS scheme in [9] and give a specific attack.

3.1 Simple Review of the ECLAS Scheme

Setup: Given a security parameter l, the KGC picks two groups G_1 and G_2 with prime order q where G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group, generates a generator P of G_1 and a bilinear map $e : G_1 \times G_1 \to G_2$, randomly chooses $s \in Z_q^*$ as a master secret key and calculates the system public key as $P_{pub} = sP$, chooses four cryptographically secure hash functions: H_1, H_3 and H_4: $\{0,1\}^* \to G_1$, H_2: $\{0,1\}^* \times \{0,1\}^* \times G_1 \times G_1 \to Z_q^*$. Finally the KGC keeps s secret and makes the $params = \{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3, H_4\}$ public.

Partial-Private-Key-Extract: The KGC calculates $Q_i = H_1(ID_i)$, $D_i = sQ_i$ and outputs D_i as the partial private key of user U_i with identity ID_i.

User-Key-Generate: By performing the following steps, a user U_i randomly selects $x_i \in Z_q^*$ as secret value and computes $P_i = x_i P$ as the public key.

Sign: Given a message m_i and a state information w, a user U_i with identity ID_i executes the following procedures to generate the signature:

1) Select randomly $r_i \in Z_q^*$ to compute $R_i = r_i P$.
2) Compute $T_i = h_i D_i + x_i Z + r_i F$, where $h_i = H_2(m_i, ID_i, P_i, R_i)$, $Z = H_3(w)$ and $F = H_4(w)$.
3) Output the signature $\sigma_i = (R_i, T_i)$ on the message m_i.

Aggregate: When receiving n message-signature pairs $\{(m_1, \sigma_1), (m_2, \sigma_2), \cdots, (m_n, \sigma_n)\}$ from n users $(U_1, U_2, \cdots, U_n)$, a signature aggregator calculates

$T = \sum_{i=1}^{n} T_i$ and outputs $\sigma = (R_1, R_2, \cdots, R_n, T)$ as an aggregate signature on the message $(m_1, m_2, \cdots, m_n)$.

Aggregate Verify: Given n users $(U_1, U_2, \cdots, U_n)$ with identities $(ID_1, ID_2, \cdots, ID_n)$, n corresponding public keys $(pk_1, pk_2, \cdots, pk_n)$, the state information w and the aggregate signature σ on the messages $(m_1, m_2, \cdots, m_n)$, the verifier takes the following steps:

1) Calculate $Q_i = H_1(ID_i)$, $h_i = H_2(m_i, ID_i, P_i, R_i)$ for all $i(1 \le i \le n)$, $Z = H_3(w)$ and $F = H_4(w)$.

2) Check whether the following equation holds or not. The aggregate signature is accepted if the equation holds, otherwise it will be invalid and refused.

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

3.2 Security Analysis of ECLAS Scheme

The authors in [9] claimed that the ECLAS scheme is existentially unforgeable under adaptive chosen-message attacks against the two types of adversaries. In this subsection, we will prove the ECLAS scheme is insecure against the type II adversary by a concrete attack.

In **Game 2**, the type II adversary A_2 acts as a malicious KGC, it has access to the master secret key but cannot replace the public key of any user. Next, we show that how A_2 initiates an attack to forge a valid signature. The detailed steps are shown as follows.

First, suppose that A_2 intercepts a legal message-signature pair $(m_i, \sigma_i = (R_i, T_i))$, where $R_i = r_i P$, $T_i = h_i D_i + x_i Z + r_i F$, where r_i is a random value of Z_q^* and unknown to A_2.

Second, A_2 can compute $T_i' = T_i - h_i D_i$, where $h_i = H_2(m_i, ID_i, P_i, R_i)$. Since knowing the master secret key s, A_2 can compute the partial private key $D_i = sQ_i$ of the user with identity ID_i, where $Q_i = H_1(ID_i)$. Then A_2 computes $h_i' = H_2(m_i', ID_i, P_i, R_i')$, where $R_i' = R_i$, finally calculates $T_i'' = T_i' + h_i' D_i$.

Third, for a message $m_i'(m_i' \ne m_i)$, A_2 outputs the forged signature $\sigma_i' = (R_i, T_i'')$ on m_i'.

Obviously, the forged signature σ_i' is a valid signature on the message m_i' because it satisfies the equation $e(T_i'', P) = e(P_{pub}, h_i' Q_i) e(Z, P_i) e(F, R_i)$.

$$\begin{aligned} e(T_i'', P) &= e(T_i - h_i D_i + h_i' D_i, P) \\ &= e(T_i, P) e(h_i D_i, P)^{-1} e(h_i' D_i, P) \\ &= e(P_{pub}, h_i Q_i) e(Z, P_i) e(F, R_i) \\ &\quad e(h_i' D_i, P) e(h_i D_i, P)^{-1} \\ &= e(h_i D_i, P) e(Z, P_i) e(F, R_i) \\ &\quad e(h_i' D_i, P) e(h_i D_i, P)^{-1} \\ &= e(Z, P_i) e(F, R_i) e(h_i' D_i, P) \\ &= e(h_i' s Q_i, P) e(Z, P_i) e(F, R_i) \\ &= e(P_{pub}, h_i' Q_i) e(Z, P_i) e(F, R_i). \end{aligned}$$

Once intercepting n valid messages-signature pairs $(m_i, \sigma_i = (R_i, T_i))_{i=1}^n$, A_2 performs above attacks and forges n valid message-signature pairs $(m_i', \sigma_i' = (R_i, T_i'))_{i=1}^n$, where $(m_i' \ne m_i)_{i=1}^n$. Then A_2 outputs $T' = \sum_{i=1}^{n} T_i''$ and the forged aggregate signature $\sigma' = (R_1, R_2, \cdots, R_n, T')$. Obviously, since the individual equation $e(T_i'', P) = e(P_{pub}, h_i' Q_i) e(Z, P_i) e(F, R_i)$ holds, it is easily verified that the forged aggregate signature σ_i' is a legal signature by the following equation:

$$\begin{aligned} e(T_i', P) &= e((T_1'', T_2'', \cdots, T_n''), P) \\ &= e(T_1'', P) e(T_2'', P), \cdots, e(T_n'', P) \\ &= e(P_{pub}, h_1' Q_1) e(Z, P_1) e(F, R_1), \cdots, \\ &\quad e(P_{pub}, h_n' Q_n) e(Z, P_n) e(F, R_n) \\ &= e(P_{pub}, h_i' Q_i) e(Z, \sum_{i=1}^{n} P_i) e(F, \sum_{i=1}^{n} R_i). \end{aligned}$$

In conclusion, the ECLAS scheme is not secure as the authors claimed. In fact, A_2 is the most difficult to deal with in CL-PKC schemes since it knows the master secret key and can compute the partial private key D_i for any ID_i. By intercepting a legal signature $\sigma_i = (R_i, T_i)$ on message m_i, A_2 can create a new valid signature $\sigma_i = (R_i, T_i'')$ on the message $m_i'(m_i' \ne m_i)$ without changing R_i. The main reason is that a lot of variables in the linear expression of T_i are known or easy to compute for A_2. This is a taboo that must be avoided in designing an aggregate signature scheme.

4 A Revocable Certificateless Aggregate Signature Scheme

To resist the drawback of ECLAS and address the compromise or expiration of signing key, we put forward a revocable certificateless aggregate signature (RCLAS) scheme in this section. The RCLAS mainly consists of the following eight algorithms.

4.1 Setup

Input a security parameter l, the algorithm outputs two groups G_1 and G_2 with prime order q where G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group, a generator P of G_1, a bilinear map $e : G_1 \times G_1 \to G_2$ and five cryptographically secure hash functions, where H_1, H_2, H_3 and $H_4 : \{0,1\}^* \to G_1$, $H_5 : \{0,1\}^* \to Z_q^*$. Next it randomly chooses $s \in Z_q^*$ as the master secret key and computes the system public key $P_{pub} = sP$. Finally, the KGC makes

$params = \{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5\}$ public while keeps s secret.

4.2 Public-Key-Extract

Without losing generality, assume U_i has identity ID_i. The user U_i selects a random value $x_i \in Z_q^*$ as secret value and takes x_i as input to compute $pk_i = x_i P$ as the public key.

4.3 Partial-Private-Key-Extract

The KGC generates the partial private key D_i for each user U_i with the corresponding public key pk_i by the following steps:

1) Calculate $Q_i = H_1(ID_i, pk_i)$.
2) Output $D_i = sQ_i$ and send D_i to the user by a secure channel.

4.4 Time-Key-Update

Given the identity ID_i of U_i, the corresponding public key pk_i and an expiration time t_i, KGC executes the following operations:

1) Compute $V_i = H_2(ID_i, pk_i, t_i)$ and the user's time update key $T_i = sV_i$.
2) Send T_i to the user and make (ID_i, t_i, T_i) public.

The reason to make (ID_i, t_i, T_i) public is that anyone can easily compute V_i and verify that T_i is actually a time update key on the identity ID_i and the time period t_i by checking whether the equation $e(P, T_i) = (V_i, P_{pub})$ holds. When the verification equation does not hold, it means that the user does not update his/her time key T_i in time.

4.5 Private-Key-Extract

A U_i generates his/her private key by taking D_i, T_i and x_i as inputs, calculates private key $sk_i = (D_i + T_i, x_i)$. The sk_i will update accordingly to the change of expiration time t_i.

4.6 Sign

Given a message m_i, a state information w (w can be current time, system parameters or arbitrary strings, which is selected and broadcasted to each signer by the aggregator, like the roadside unit RSU periodically broadcasting information in vehicular networks.), a non-revoked user U_i with private/public key sk_i/pk_i to execute the following procedures to generate a signature:

1) Randomly select $r_i \in Z_q^*$ to compute $U_i = r_i P$.
2) Compute $h_i = H_5(m_i, ID_i, U_i, pk_i, t_i, w)$, $F = H_3(w)$, $W = H_4(w)$, $R_i = h_i U_i$.
3) Compute $S_i = D_i + T_i + x_i F + h_i r_i W$.
4) Output the signature $\sigma_i = (R_i, S_i)$ on the message m_i.

4.7 Aggregate

When receiving n message-signatures pairs $\{(m_1, \sigma_1), (m_2, \sigma_2), \cdots, (m_n, \sigma_n)\}$ from n distinct non-revoked users $(U_1, U_2, \cdots, U_n)$ under the same state information w with the expiration times $(t_1, t_2, \cdots, t_n)$, a signature aggregator can calculate $R = \sum_{i=1}^{n} R_i$, $S = \sum_{i=1}^{n} S_i$ and output $\sigma = (R, S)$ as an aggregate signature on message $(m_1, m_2, \cdots, m_n)$.

4.8 Aggregate Verify

Given n users $(U_1, U_2, \cdots, U_n)$ with identities $(ID_1, ID_2, \cdots, ID_n)$, n public keys $(pk_1, pk_2, \cdots, pk_n)$, n expiration times $(t_1, t_2, \cdots, t_n)$, the state information w and the aggregate signature $\sigma = (R, S)$ on message $(m_1, m_2, \cdots, m_n)$, any verifier takes the following steps:

1) Calculate $Q_i = H_1(ID_i, pk_i)$, $V_i = H_2(ID_i, pk_i, t_i)$, $F = H_3(w)$, $W = H_4(w)$.
2) Check whether the following equation (1) holds or not. If the equation holds, the aggregated signature σ is regarded as valid, otherwise, $\sigma = (R, S)$ is considered as an invalid signature.

$$e(S, P) = e(\sum_{i=1}^{n}(Q_i + V_i), P_{pub}) e(\sum_{i=1}^{n} pk_i, F) e(R, W). \quad (1)$$

5 Security Proof

In this section, the security (including correctness and unforgeability) of our RCLAS scheme will be proven.

5.1 Correctness

$$\begin{aligned} e(S, P) &= e(\sum_{i=1}^{n}(D_i + T_i + x_i F + h_i r_i W), P) \\ &= e(\sum_{i=1}^{n}(D_i + T_i), P) e(\sum_{i=1}^{n} x_i P, F) e(\sum_{i=1}^{n} h_i r_i P, W) \\ &= e(\sum_{i=1}^{n} s(Q_i + V_i), P) e(\sum_{i=1}^{n} pk_i, F) e(\sum_{i=1}^{n} R_i, W) \\ &= e(\sum_{i=1}^{n}(Q_i + V_i), P_{pub}) e(\sum_{i=1}^{n} pk_i, F) e(R, W). \end{aligned}$$

5.2 Unforgeability

In this subsection, the security proof of our RCLAS scheme is proved under the hardness assumption of

CDHP. The **Theorems 1**, **2** and **3** show that the RCLAS scheme is secure against three types of adversaries in **Game 1**, **Game 2** and **Game 3**, respectively. Among the three types of adversaries, A_2 simulates an adversary who has known the master key s. Generally, A_2 has the strongest attack force and is the most difficult adversary to resist, so we mainly take **Theorem 2** as an example to show how our RCLAS scheme can achieve security under A_2 attacks. In the following, t_m represents the time for computing a scalar multiplication in G_1 and n is the size of the aggregate set.

Theorem 1. *In the random oracle model, if there is a type I adversary A_1 who has a non-negligible advantage ε in forging a valid aggregate signature of the RCLAS scheme in an attack model of **Game 1** within a time span t after making at most q_i times queries to the random oracles $H_i(1 \leq i \leq 5)$, q_{ppk} times Partial-Private-Key-Extract queries, q_{tk} Time-Key-Update queries, q_{pk} times Public-key-Extract queries, q_{rep} times Public-key-Replacement queries and q_{sig} times Sign queries, then the CDHP can be solved within time $t' \leq t + O[(2q_1 + q_2 + q_3 + q_4 + 2q_{ppk} + q_{tk} + q_{pk} + 5q_{sig} + n + 2)t_m]$ and with non-negligible probability $\varepsilon' \geq \frac{\varepsilon}{e(q_{ppk}+n)}$.*

Proof. The proof process is very similar to **Theorem 2**. The details of the proof process will be omitted here duo to the limit length. □

Theorem 2. *In the random oracle model, if there is a type II adversary A_2 who has a non-negligible advantage ε in forging a valid aggregate signature of our RCLAS scheme in an attack model of **Game 2** within a time span t after making at most q_i times queries to the random oracles $H_i(3 \leq i \leq 5)$, q_{pk} times Public-key-Extract queries, q_s times Secret-Value queries and q_{sig} times Sign queries, then the CDHP can be solved within time $t' \leq t + O[(q_3 + q_4 + q_{pk} + q_s + 5q_{sig} + n + 1)t_m]$ and with non-negligible probability $\varepsilon' \geq \frac{\varepsilon}{e(q_s+n)}$.*

Proof. Let the challenge C receives a random CDHP instance (P, aP, bP) in G_1, here P is a generator of G_1. A type II adversary A_2 interacts with C as modeled in **Game 2**, and we show that how C can use A_2 as a subroutine to find the solution abP to the CDHP instance. □

Setup: C firstly chooses a master secret key s and compute $P_{pub} = sP$. C selects system parameters $params = \{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5\}$, then sends the master secret key s and $params$ to A_2.

Queries: A_2 can perform a polynomially bounded number of the following types of queries in an adaptive manner. Hash functions H_1, H_2, H_3, H_4 and H_5 are considered as random oracles. All inquiries-responses will be kept in the corresponding lists. Since A_2 knows the master secret key s, it can compute all partial private keys and all time keys, so A_2 has no need to request the H_1 queries, H_2 queries, Partial-Private-Key-Extract queries and Time-Key-Update queries.

- H_3 queries: C maintains an initially empty list L_3 with structure (w, γ_i, F_i). When A_2 issues a query $H_3(w)$, the same answer will be given if the query has been asked before. Otherwise, C selects randomly $\gamma_i \in Z_q^*$, sets $F_i = \gamma_i aP$, adds (w, γ_i, F_i) to L_3 and returns F_i to A_2.

- H_4 queries: C maintains an initially empty list L_4 with structure (w, δ_i, W_i). When A_2 submits a query $H_4(w)$, the same response will be given if the query has been asked before. Otherwise, C picks randomly $\delta_i \in Z_q^*$, sets $W_i = \delta_i P$, adds (w, δ_i, W_i) to L_4 and returns W_i to A_2.

- H_5 queries: C maintains an initially empty list L_5 with structure $(m_i, ID_i, U_i, pk_i, t_i, w, h_i)$. When A_2 issues a query $(m_i, ID_i, U_i, pk_i, t_i, w)$ to H_5, the same answer will be given if the query has been asked before. Otherwise, C selects randomly $h_i \in Z_q^*$, adds $(m_i, ID_i, U_i, pk_i, t_i, w, h_i)$ to L_5 and returns the answer h_i to A_2.

- Public-Key-Extract queries: C keeps an initially empty list L_{pk} with structure (ID_i, x_i, pk_i, c_i). When A_2 performs a query with the identity ID_i to this random oracle, the same answer will be given if the query has been asked before. Otherwise, C first chooses a random value $x_i \in Z_q^*$ as the secret value, and then flips a coin $c_i \in \{0, 1\}$ that yields 0 with probability θ and 1 with probability $1 - \theta$. If $c_i = 0$, C computes $pk_i = x_i bP$ and adds $(ID_i, \perp, pk_i, c_i)$ to L_{pk}. If $c_i = 1$, C computes $pk_i = x_i P$, and adds (ID_i, x_i, pk_i, c_i) to L_{pk} and returns pk_i to A_2.

- Secret-Value queries: When A_2 performs a Secret-Value query on ID_i, C first makes a Public-Key-Extract query and finds (ID_i, x_i, pk_i, c_i) in L_{pk}. If $c_i = 1$, C computes $pk_i = x_i P$ and returns x_i to A_2. Or else, C returns $\perp$.

- Sign queries: When A_2 performs a Sign query on the tuple $(m_i, ID_i, w, pk_i, t_i)$, C executes the following operations to generate a valid signature:

1) If $c_i = 0$, C selects $r_i \in Z_q^*$ at random, sets $h_i = \delta_i^{-1}$ and $W_i = \delta_i aP$, computes $U_i = r_i P - \gamma_i pk_i$, $R_i = h_i U_i$ and $S_i = D_i + T_i + r_i aP$, finally returns the signature $\sigma_i = (R_i, S_i)$.

2) If $c_i = 1$, C runs the Sign algorithm normally to get a regular signature $\sigma_i = (R_i, S_i)$.

Forgery: In the end, suppose A_2 can output a tuple $(m^*, ID^*, t^*, w, \sigma^*)$ in which w is a state information, $m^* = (m_1^*, m_2^*, \cdots, m_n^*)$, $ID^* = (ID_1^*, ID_2^*, \cdots, ID_n^*)$, $t^* = (t_1^*, t_2^*, \cdots, t_n^*)$ and $\sigma^* = (R^*, S^*)$ is a valid forged aggregate signature. For

$1 \le i \le n$, C finds tuples of (w, γ_i, F_i), (w, δ_i, W_i) and $(m_i, ID_i, U_i, pk_i, t_i, w, h_i)$ from L_3, L_4 and L_5, respectively. C proceeds only if $c_1^* = 0, c_i^* = 1(2 \le i \le n)$. Otherwise, C aborts. If the forged signature $\sigma^* = (R^*, S^*)$ meets the above conditions, then satisfies Equation (1), we have

$$e(pk_1^*, F^*) = e(S^*, P)e(\sum_{i=1}^{n}(Q_i + V_i), P_{pub})^{-1} e(\sum_{i=2}^{n} pk_i^*, F^*)^{-1} e(R^*, W^*).$$

Where $pk_1^* = x_1^* bP, F^* = \gamma^* aP, W^* = \delta^* P$ and $pk_i^* = x_i^* P(2 \le i \le n)$. So it is easy for C to obtain the solution to the given CDHP instance:

Now, we analyze the probability to solve a CDHP by type II adversary A_2 in the polynomial bounded time. We analyze the three events for C to succeed:

E_1: C does not abort all the queries of Secret-Value-Extract queries.

E_2: A_2 generates a valid and nontrivial aggregate signature forgery.

E_3: E_2 occurs, $c_1^* = 0, c_i^* = 1(2 \le i \le n)$.

C succeeds if the above events happen, so $\varepsilon' = Pr[E_1 \wedge E_2 \wedge E_3]$. We can know that $Pr[E_1] \ge (1-\theta)^{q_s}, Pr[E_2|E_1] \ge \varepsilon, Pr[E_3|E_1 \wedge E_2] \ge \theta(1-\theta)^{n-1}$, thus $\varepsilon' = Pr[E_1 \wedge E_2 \wedge E_3] \ge (1-\theta)^{q_s}\varepsilon\theta(1-\theta)^{n-1} = \theta(1-\theta)^{q_s+n-1}\varepsilon$

When $\theta = \frac{1}{q_s+n}$, $\theta(1-\theta)^{q_s+n-1}$ is maximized at $\frac{1}{q_s+n}(1-\frac{1}{q_s+n})^{q_s+n-1}$. When q_s is sufficient large, this probability approaches $\frac{\varepsilon}{e(q_s+n)}$. So we can get $\varepsilon' \ge \frac{\varepsilon}{e(q_s+n)}$.

The running time for C is the sum of $A_2's$ running time, the time for C to response the queries and the time for C to compute the CDHP instance. During H_3 queries, H_4 queries, Public-key-Extract queries, Secret-Value queries and Sign queries, it needs 1, 1, 1, 1, 5 scalar multiplications, respectively. And during C computing the CDHP instance, it needs $n+1$ scalar multiplication, so $t' \le t + O[(q_3 + q_4 + q_{pk} + q_s + 5q_{sig} + n + 1)t_m]$.

From all of the above, C can solve the CDHP instance with non-negligible probability that contradicts to the intractability assumption of CDHP.

Theorem 3. *In the random oracle model, if there is a type III adversary A_3 who has a non-negligible advantage ε in forging a valid aggregate signature of the RCLAS scheme in an attack model of **Game 3** within a time span t after making at most q_i times queries to the random oracles $H_i(1 \le i \le 5)$, q_{ppk} times Partial-Private-Key-Extract queries, q_{tk} Time-Key-Update queries, q_{pk} times Public-key-Extract queries and q_{sig} times Sign queries, then the CDHP can be solved with non-negligible probability $\varepsilon' \ge \frac{\varepsilon}{e(q_{tk}+n)}$ and within time $t' \le t + O[(2q_1 + 2q_2 + q_3 + q_4 + 2q_{ppk} + 2q_{tk} + q_{pk} + 5q_{sig} + n + 2)t_m]$.*

Proof. The proof process is very similar to **Theorem 2**. The details of the proof process will be omitted here because of the limit length.

According to **Theorem 1**, **Theorem 2** and **Theorem 3**, we can conclude that there is no PPT adversary of any type can forge a valid aggregate signature of the proposed RCLAS scheme with a non-negligible advantage in polynomial time. Hence, our scheme is secure under the hardness assumption of CDHP.

6 Performance Comparisons

In this section, we make performance comparisons between our RCLAS scheme and the schemes in [9, 16, 19]. Due to the limited knowledge of the authors, no revocable certificateless aggregate signature scheme has been found so far. Therefore, this paper compares the revocable certificateless signature (RCLS) scheme [16], the ECLAS scheme [9] which has been analyzed in our Section 3, and the latest new certificateless aggregate signature (NCLAS) scheme [19] as the comparison schemes. In comparison, we omit the computations which take little time such as Hash for simplicity.

From Table 1, we can see that our RCLAS scheme has relatively little computation and shorter length of aggregate signature than other schemes while realizing the function of user revocation. Compared with [16], our RCLAS scheme adds the property of signature aggregation which can greatly improve verification efficiency and may enjoy better practicality. As for the length of the aggregate signature, our RCLAS scheme only consists of two elements in G_1, which is far shorter than the schemes in [9, 19] and greatly saves the communication costs and storage space. In addition, our RCLAS scheme can realize user's revocation flexibly by time update key for practical scenarios while the schemes in [9, 19] cannot. In general, our RCLAS scheme has better comprehensive performance (Note: In Table 1, Sign and A-V cost denote the computational cost of generation and verification of aggregate signature, respectively; A-S size represents the size of an aggregate signature; s and p mean the computational cost of scalar multiplication and a bilinear pairing operation, respectively; $|G_1|$ represents the bit length of an element in G_1; "$\surd$" means "support"; "$\times$" means "not support"; "—" means "not mentioned").

Here, we give a more intuitively quantitative analyses for schemes in [9,19] and our scheme. We adopt the experiment in [10], which observes processing time for the Tate pairing on a 159-bit subgroup of an MNT curve with an implanting degree 6 at an 80-bit security level, running on an Intel i7 3.07 GHz machine. Thus the time consumed by various operations is as follows: P is 3.21ms and S is 0.39ms. Suppose that n=100 in the Aggregate Verify phase, the comparisons of computational cost are shown in Figure 2. From Figure 2, we can see that the computational cost of the three schemes is equal in Sign phase, yet in the Aggregate-Verify phase, the computational cost of

Table 1: Comprehensive comparisons between related schemes

Scheme	Sign cost	A-V cost	A-S size	Revocation
RCLS in [16]	3s	—	—	√
NCLAS in [19]	4s	$3p + 2ns$	$(n+1)\|G_1\|$	×
ECLAS in [9]	4s	$4p + ns$	$(n+1)\|G_1\|$	×
RCLAS	4s	$4p$	$2\|G_1\|$	√

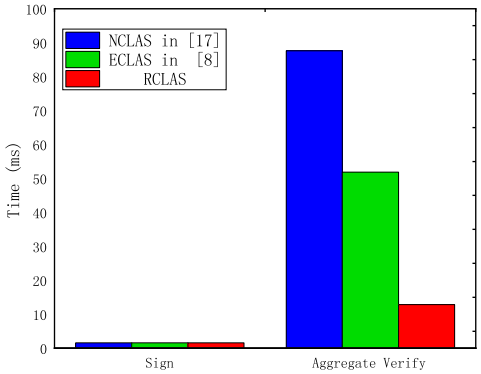

Figure 2: Computational cost comparisons

our scheme is much lower than other two schemes. Thus, the total computational cost of our scheme is reduced by 85.3, 75.2 percentage compared with those of schemes in [19] and [9], respectively.

7 Conclusion

In this paper, we first analyze the security of an efficient certificateless aggregate signature scheme (ECLAS) proposed in [9] and then give a specific attack. More specifically, any type II adversary A_2 can forge a valid aggregate signature on any set of messages as long as A_2 intercepts some legal message-signature pairs. In order to overcome this security flaw, we put forward an improved revocable certificateless aggregate signature (RCLAS) scheme, which not only can keep the advantages of aggregate signature, but also can flexibly deal with the problem of user's private key being compromised or expired in CL-PKC. The length of the aggregate signature in our RCLAS scheme only consists of two points in G_1 which is far shorter and greatly saves the communication cost and storage space. Finally, we show that our RCLAS scheme is proved to be existential unforgeable against adaptive chosen-message attacks under the hardness assumption of CDHP. And performance analyses show our RCLAS has better comprehensive performance while maintaining high computation and storage efficiency than some existing schemes.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China under Grant 61802243, U1705264; the Key R-D Program in industry

eld of Shaanxi Province under Grant 2019JY-013; the Fundamental Research Funds for the Central Universities under Grant2019CSLY002, GK201803005; the Natural Science Foundation of Fujian Province under Grant 2019J01275. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," *Asiacrypt, Lecture Notes in Computer Science*, vol. 2894, no. 2, pp. 452–473, 2003.

[2] D. Boneh, X. Ding, G. Tsudik, and C. M. Wong, "A method for fast revocation of public key certificates and security capabilities," in *Conference on 10th Usenix Security Symposium*, 2001. (`https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=2045&context=sis_research`)

[3] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 416–432, 2003.

[4] C. Chen, H. Chien, and G. Horng, "Cryptanalysis of a compact certificateless aggregate signature scheme," *International Journal of Network Security*, vol. 18, no. 4, pp. 793–797, 2016.

[5] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, "An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks," *Information Sciences*, vol. 451, pp. 1–15, 2018.

[6] Z. Gong, Y. Long, X. Hong, and K. F. Chen, "Two certificateless aggregate signatures from bilinear maps," in *The 8th ACIS International Conference on SPND*, vol. 3, pp. 183–193, 2007.

[7] D. He, M. Tian, and J. Chen, "Insecurity of an efficient certificateless aggregate signature with constant pairing computations," *Information Sciences*, vol. 268, pp. 458–462, 2014.

[8] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves",

Computer Standards & Interfaces, vol. 26, no. 2, pp. 73–84, 2004.

[9] B. Kang, M. Wang, and D. Jing, "An efficient certificateless aggregate signature scheme," *Wuhan University Journal of Natural Sciences*, vol. 22, no. 2, pp. 165–170, 2017.

[10] P. Kumar, S. Kumari, V. Sharma, A. Sangaiah, J. Wei, and X. Li, "A certificateless aggregate signature scheme for healthcare wireless sensor network," *Sustainable Computing: Informatics and Systems*, vol. 18, no. 1, pp. 80–89, 2017.

[11] Y. Li, H. Nie, Y. Zhou, and B. Yang, "A novel and provably secure certificateless aggregate signature scheme," *Journal of Cryptologic Research*, vol. 2656, no. 7, pp. 526–535, 2015.

[12] H. Nie, Y. Li, W. Chen, and Y. Ding, "Nclas: A novel and efficient certificateless aggregate signature scheme," *Security and Communication Networks*, vol. 9, no. 6, pp. 3141–3151, 2016.

[13] A. Shamir, "Identity-based cryptosystems and signature schemes," *Workshop on the Theory and Application of Cryptographic Techniques*, vol. 196, pp. 47–53, 1984.

[14] S. Shan, "An efficient certificateless signcryption scheme without random oracles," *International Journal of Electronics and Information Engineering*, vol. 11, no. 1, pp. 9-15, 2019.

[15] L. Shen, F. Zhang, and Y. Sun, "Efficient revocable certificateless encryption secure in the standard model," *Computer Journal*, vol. 57, no. 4, pp. 592–601, 2014.

[16] Y. Sun, F. Zhang, and L. Shen, "A revocable certificateless signature scheme," *Journal of Computers*, vol. 9, no. 8, pp. 355–364, 2014.

[17] T.T Tsai and Y.M Tseng, "Revocable certificateless public key encryption," *IEEE Systems Journa*, vol. 9, no. 3, pp. 824–833, 2015.

[18] Y. M. Tseng and T. T. Tsai, "Efficient revocable id-based encryption with a public channel," *The Computer Journal*, vol. 55, no. 4, pp. 475–486, 2012.

[19] L. Wu, Z. Xu, D. He, and X. Wang, "New certificateless aggregate signature scheme for healthcare multimedia social network on cloud environment," *Security and Communication Networks*, vol. 2018, pp. 1–13, 2018.

[20] T. Y. Wu, T. T. Tsai, and Y. M. Tseng, "Revocable id-based signature scheme with batch verifications," in *The 8th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 49–54, 2012.

[21] H. Xiong, Z. Guan, Z. Chen, and F. Li, "An efficient certificateless aggregate signature with constant pairing computations," *Information Sciences*, vol. 219, no. 10, pp. 225–235, 2013.

[22] W. S. Yap, S. S. M. Chow, S. H. Heng, and B. M. Goi, "Security mediated certificateless signatures," *Lecture Notes in Computer Science*, vol. 4521, pp. 459–477, 2007.

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

[24] Y. Zhang, C. Li, D. Zhou, and C. Wang, "Efficient revocable certificateless signature scheme," *Computer Engineering*, vol. 41, no. 7, pp. 157–162, 2015.

[25] M. Zhou, M. Zhang, C. Wang, and B. Yang, "Cclas: A practical and compact certificateless aggregate signature with share extraction," *International Journal of Network Security*, vol. 16, no. 3, pp. 174–181, 2014.

Biography

Fuxiao Zhou received her B.S. degree from Henan Normal University, Xinxiang, China, in 2017. She now is a M.S. degree candidate in Applied Mathematics with the School of Mathematics and Information Science, Shaanxi Normal University, Xi'an, China. Her research interests include certificateless signature and its applications.

Yanping Li received her M. S. degree from Shaanxi Normal University in 2004 and Ph. D degree from Xidian University in 2009, Xi'an, China. She now is an associate professor with the School of Mathematics and Information Science, Shaanxi Normal University. Her research interests include applied cryptography and its applications.

Changlu Lin received the B.S. and M.S. degrees in mathematics from Fujian Normal University, China, in 2002 and 2005, respectively, and the Ph.D. degree in information security from the State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, China, in 2010. He currently works with the College of Mathematics and Informatics, and the Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University. He is interested in cryptography and network security. He has conducted research in diverse areas, including secret sharing, multiparty computation, public key cryptography and their applications.