Certificateless Ring Signature Scheme from Elliptic Curve Group

Lunzhi Deng[1], Siwei Li[2], Huawei Huang[1], Yuhong Jiang[1], Bingqin Ning[1]

[1] School of Mathematical Sciences, Guizhou Normal University, China
[2] Graduate School, Guizhou Normal University, China
denglunzhi@163.com, 490613284@qq.com, hwhuang7809@163.com,
340703460@qq.com, 2592083781@qq.com

Abstract

Ring signature is an anonymous authentication that authenticates the message while protecting the identity of the authentic signer. It is suitable for many scenarios, such as electronic voting, anonymous access control, etc. Most certificateless ring signature schemes currently known used bilinear pairings. The computation cost of the pairings is much higher than that of the scalar multiplication over the elliptic curve group. Therefore, it is quite significant to design certificateless ring signature without pairing. In this paper, we constructed a new certificateless ring signature scheme and prove it to be secure in the random oracle model, under the assumption that it is hard to solve the discrete logarithm (DL) problem on elliptic curve group. Our scheme does not use pairing operation, taken into account the computation costs for signing and verifying, it is more efficient than all the certificateless ring signature schemes currently known.

Keywords: Certificateless cryptography, Ring signature, Elliptic curve, DL problem, Random oracle model

1 Introduction

In traditional public key infrastructure (PKI), there is a dependable certification authority (CA) which is responsible for issuing a certificate binding the user to his public key. That brings the certificate management problem. To solve the problem, Shamir [15] introduced identity-based public key cryptography, there is a credible private key generator (PKG) which is responsible for generating a private key for an user according to his identity. However, it results in the key escrow problem.

To solve the two problems, Al-Riyami et al. [1] introduced certificateless public key cryptography, there is a semi-trusted key generation center (KGC) which is responsible for generating a partial private key for user with respect to the identity of user. The full private key of user includes two parts: partial private key generated by KGC and a secret value chosen by user himself.

In 2001, Rivest et al. [14] put forward the concept of the ring signature. In this setting, a signer can choose several members to form a group and generate a signature without the assistance of the other group members. By verifying the generated signature, any verifier confirms that the message was signed by someone in the group but cannot identify the real signer in the group member.

Ring signatures are applied to some actual scenarios, such as leaking secret anonymously and accessing control anonymously. For example, a reputable official wants to reveal an important information to the media, but he does not want to reveal his own identity. He may choose a few similar officials to form a group, then generate a signature on the information on behalf of the group, and send it to the news department, after receiving the signature, the verifier is sure that this information come from someone in the group, but he can not find out the true signer. Another example, in access control anonymously, an authorized entity can select several other qualified personnel to form a group, then generate a signature on the access request on behalf of the entire group, after receiving the signature, the data manager can confirm the request come from someone in the group, but can not determine the real signer. This approach not only ensures the security of the data but also protects the accessor's personal privacy.

1.1 Related Work

Since ring signature was first formalized, many practical ring signature schemes and their variants have been proposed. Bender et al. [2] proposed the first constructions of ring signature scheme in the standard model. Yuen et al. [23] constructed the first linkable ring signature scheme, and gave the proof of security in the standard model. Chow et al. [3] presented an ID-based ring signature scheme, which requires only two

*Corresponding Author: Lunzhi Deng; E-mail: denglunzhi@163.com
DOI: 10.3966/160792642020052103010

pairing operations for any group size. Herranz [8] proposed an identity-based ring signature scheme, and prove it to be secure based on RSA problem. Nguyen [13] constructed an identity-based ring signature scheme, the size of signature is constant. Deng et al. [6] proposed two identity-based threshold ring signature schemes, and gave the proof of security based on computational Diffie-Hellman problem. Since the certificateless public key cryptosystem was formalized, many certificateless signature schemes [11, 12, 16-19, 22] have been proposed. However, there are only several work published on certificateless ring signature (CLRS) schemes [4-5, 7, 20-21, 24]. Chow et al. [4] put forward the security model of CLRS and proposed a concrete construction, which requires n pairing operations and $3n+1$ exponentiation operations. Zhang et al. [24] presented another CLRS scheme, which requires 5 pairing operations and $4n+3$ exponentiation operations. Chang et al. [5] proposed the formal security definition which captures the user partial key replacement attack, then designed a concrete CLRS scheme which requires 4 pairing operations and $4n+4$ exponentiation operations. Wang and Han [20] introduced the notion of certificateless threshold ring signature, and presented a concrete scheme. Wang [21] proposed a CLRS scheme from anonymous subsets. Deng [7] constructed a CLRS scheme, and prove it to be secure based on RSA problem and discrete logarithm (DL) problem. Zhang et al. [25] put forward a new CLRS scheme, and gave the proof of security based on computational Diffie-Hellman problem and computational co-Diffie-Hellman problem.

1.2 Motivations and Contributions

The computation cost of the pairing is higher than that of the scalar multiplication over the elliptic curve group. There is only one CLRS scheme [7] without using pairing, which used exponentiation operations in a RSA group. However, under the same safety requirements, the computation cost of the exponentiation in a RSA group is also higher than that of the scalar multiplication over the elliptic curve group (see Table 2). So it is attractive to construct a CLRS scheme with only using scalar multiplication over the elliptic curve group.

In this paper, a new CLRS scheme is proposed having the following features:

- The scheme is secure under the strong security model. Namely, the Type I/II adversary can obtain the valid signatures for the replaced public key, without additional submission.
- Most CLRS schemes currently known used pairings, our scheme does not use pairing and it is more efficient than previous ones.

1.3 Roadmap

The organization of the paper is sketched as follows: First, the preliminaries and the system models of CLRS scheme are introduced in Section 2 and Section 3, respectively. Second, the construction and the security proofs for a new CLRS scheme are given in Section 4 and Section 5, respectively. Next, the performance comparisons on several schemes are presented in Section 6. Lastly, some conclusions are given in Section 7.

2 Preliminaries

In this section, we introduce two mathematical tools: elliptic curve group and discrete logarithm problem. They will be used in the construction of the schemes and the proofs of security

2.1 Notation

The notations used throughout the paper are listed in Table 1.

2.2 Elliptic Curves Modulo a Prime

Let $p>3$ be prime. The elliptic curve $\mathbb{E}$ over F_p is the set of solutions $(x,y)\in F_p\times F_p$ to the congruence $y^2=x^3+ax+b(\mathrm{mod}\,p)$, where $a,b\in F_p$ are constants such that, $4a^3+27b^2\neq 0(\mathrm{mod}\,p)$ together with a special point O called the point at infinity.

The addition operation on $\mathbb{E}$ is defined as follows: (where all arithmetic operations are performed in F_p). Suppose $P=(x_1,y_1),Q=(x_2,y_2)\in\mathbb{E}$.

$$P+Q=\begin{cases}O & if\ x_1=x_2, y_2=-y_1\\ R=(x_3,y_3) & otherwise\end{cases}$$

$$\begin{cases}x_3=\lambda^2-x_1-x_2\\ y_3=\lambda(x_1-x_3)-y_1\end{cases}$$

$$and\ \lambda=\begin{cases}(3x_1^2+ax_3)(2y_1)^{-1} & if\ P=Q\\ (y_2-y_1)(x_2-x_1)^{-1} & otherwise\end{cases}$$

Define $P+O=O+P=P$

Definition 1. Discrete logarithm (DL) problem. Let $G=(P)\leq\mathcal{G}$, $P\in\mathcal{G}$ is a point with prime order q q. Given a point $aP\in G$, compute $a\in Z_q^*$.

3 System Models

In this section, we present the formal definition of certificateless ring signature and the security requirements.

Table 1. Notations

F_p	A prime finite field
$\mathbb{E}$	An elliptic curve over F_p.
$\mathcal{G}$	An addition group consisting of the point on $\mathbb{E}$ and an extra point O.
q	A prime number, where $q = \lvert\mathcal{G}\rvert / 2$.
Z_q^*	A set consisting of positive integers less than q.
G	An additive group with order q, where $G \le \mathcal{G}$.
P	A generator of the group G.
P_{pub}	The public key of system, where $P_{pub} = xP$.
$H_1 \sim H_3$	Three secure hash functions.
ID_i	The identity of i^{th} user.
D_i	The partial private key i^{th} user, where $D_i = (R_i, z_i)$.
t_i	The secret value i^{th} user and $T_i = t_i P$.
PK_i	The public key of i^{th} user, where $PK_i = (T_i, R_i)$ and $R_i = r_i P$.
W	A set consisting of n users, where $W = \{ID_1, ..., ID_n\}$.
U	A set consisting of the identities/public keys, where $U = W \cup \{PK_i : ID_i \in W\}$.
m	A message.
σ	A ring signature.

3.1 Formal Definition

A CLRS scheme consists of the following seven algorithms:

- Setup: Given a security parameter v, key generate center (KGC) generates the system parameters $params$ and the master secret key msk.
- Partial-Private-Key-Extract: Given the identity of user $ID_i \in \{0,1\}^*$, KGC generates the partial private key D_i
- Secret-Value-Set: The user ID_i selects a secret value t_i.
- Partial-Public-Key-Generate: The user ID_i generates his partial public key T_i.
- User-Public-Key-Set: The user ID_i outputs his user public key PK_i.
- Sign: Given a tuple (m,U), the real signer $ID_s \in W$ generates a signature σ.
- Verify: On receive the signature (m,σ,U), the verifier outputs 1 or 0, depending on whether σ is a valid ring signature on the message m.

3.2 Security Requirements

The security requirements of a CLRS scheme are presented as follows.

Definition 2. A CLRS scheme is unforgeable (UNF-CLRS) if the advantage of any polynomially bounded adversary is negligible in the following two games against Type I/II adversaries.

Game I. The first game was performed between a challenger $\mathcal{C}$ and a Type I adversary $\mathcal{A}_1$.

Initialization. $\mathcal{C}$ runs the setup algorithm to generate a master secret key msk and the public system parameters $params$. $\mathcal{C}$ keeps msk secret and gives $params$ to $\mathcal{A}_1$.

Query. $\mathcal{A}_1$ performs a polynomially bounded number of queries. Each query may depend on the answers to the previous queries.

- Hash functions query: $\mathcal{A}_1$ can query the values of the hash functions for any input.
- User public key query: $\mathcal{A}_1$ requests the user public key of a user ID_i, $\mathcal{C}$ returns the corresponding public key PK_i.
- Partial private key query: $\mathcal{A}_1$ requests the partial private key of a user ID_i, $\mathcal{C}$ responds with the partial private key D_i.
- Partial public key replacement: $\mathcal{A}_1$ supplies a new partial public key value T_i' with respect to a user ID_i.

$\mathcal{C}$ replaces the current partial public key with the value T_i'.

- Secret value query: $\mathcal{A}_1$ requests the secret value of a ID_i, $\mathcal{C}$ returns the secret value t_i. If the partial publickey of user has been replaced, $\mathcal{A}_1$ can not request the corresponding secret value.
- Signature query: $\mathcal{A}_1$ submits a tuple (m,U), $\mathcal{C}$ outputs a signature.

Forge. $\mathcal{A}_1$ outputs a new tuple (m,σ,U). The adversary wins if the following conditions hold.

1. σ is not obtained through a signature query.
2. Verify $(m,\delta,U)=1$.
3. $\mathcal{A}_1$ did not query the partial private of anyone in W.

The advantage of $\mathcal{A}_1$ is defined as:

$$Adv_{\mathcal{A}_1}^{UNF-CLRS} = \Pr[\mathcal{A}_1 wins].$$

Game II. The second game was performed between a challenger $\mathcal{C}$ and a Type II adversary $\mathcal{A}_2$.

Initialization. $\mathcal{C}$ runs the setup algorithm to obtain a master secret key *msk* and public system parameters *msk* . $\mathcal{C}$ gives them to $\mathcal{A}_2$.

Query. $\mathcal{A}_2$ makes a polynomially bounded number of queries as those in Game I.

Forge. $\mathcal{A}_2$ outputs a new tuple (m,δ,U). The adversary wins if following conditions hold.

1. δ is not obtained through a signature query.
2. Verify $(m,\delta,U)=1$.
3. $\mathcal{A}_2$ did not query the partial private of anyone in W.
4. $\mathcal{A}_2$ did not replace the partial public key of anyone in W.

The advantage of $\mathcal{A}_2$ is defined as:

$$Adv_{\mathcal{A}_2}^{UNF-CLRS} = \Pr[\mathcal{A}_2 wins]$$

Definition 3. A CLRS scheme is anonymous (ANO-CLRS) if the advantage of any polynomially bounded adversary is negligible in the following game.

Game III. A adversary $\mathcal{A}$ plays the third game with a challenger $\mathcal{C}$ as follows.

Initialization. $\mathcal{C}$ runs the setup algorithm to generate *msk* and *params*, and sends them to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ makes a polynomially bounded number of queries as those in Game I.

Challenge. $\mathcal{A}$ submits a tuple (m,U,ID_0,ID_1), where $U=W\cup\{PK_i,ID_i\in W\}$ and $ID_0,ID_1\in W$. $\mathcal{C}$ selects at random a bit $\mu\in\{0,1\}$ and provides $\mathcal{A}$ with $\delta=\mathrm{S}ign(m,U,z_\mu,t_\mu)$.

Phase 2. $\mathcal{A}$ makes queries as those in Phase 1.

Response. $\mathcal{A}$ returns a bit $\mu'\in\{0,1\}$. The adversary wins if $\mu'=\mu$.

The advantage of $\mathcal{A}$ is defined as:

$$Adv_{\mathcal{A}}^{ANO-CLRS} = \left|2\Pr[\mu'=\mu]-1\right|$$

4 Our Scheme

In this section, a new CLRS scheme is constructed as follows.

- Setup: Given a security parameter v, KGC chooses an elliptic curve group G of prime order $q>2^v$. KGC then chooses three cryptographic hash functions $H_1,H_2,H_3:\{0,1\}^*\to Z_q^*$. Finally, KGC picks a master secret key $x\in Z_q^*$ and sets the public key $P_{pub}=xP$. The set of public parameters is: $params=\{G,q,P,P_{pub},H_1,H_2,H_3\}$
- Partial-Private-Key-Extract: Given the identity of user $ID_i\in\{0,1\}^*$, KGC picks at random $r_i\in Z_q^*$ and computes $R_i=r_iP$, $k_i=H_1(ID_i,R_i)$, $z_i=r_i+k_ix$, then sends $D_i=(R_i,z_i)$ to the user via a secure channel.
- Secret-Value-Set: The user ID_i randomly chooses $t_i\in Z_q^*$.
- Partial-Public-Key-Generate: The user ID_i computes $T_i=t_iP$.
- User-Public-Key-Set: The user ID_i sets $PK_i=(T_i,R_i)$.
- Sign: Given a tuple (m,U), the actual signer $ID_s\in W$ carries out the following steps.

  (1) Computes $k_i=H_1(ID_i,R_i)$ for

  $$i=1,2,...,s-1,s+1,...,n.$$

  (2) Randomly chooses $d,c_i\in Z_q^*$ and computes

  $$l_i=H_2(m,c_i,ID_i,PK_i) \text{ for } i=1,2,...,s-1.s+1,...,n.$$

  (3) Computes.

  $$h=H_3(m,U,dP+\sum_{i=1,i\neq s}^{n}c_i(l_iT_i+R_i+k_iP_{pub})).$$

  (4) Computes $c_s=h-\sum_{i=1,i\neq s}^{n}c_i$.

  (5) Computes $l_s=H_2(m,c_s,ID_s,PK_s)$ and

  $$y=d-c_s(l_st_s+z_s).$$

  (6) Outputs $\sigma=(y,c_1,...,c_n)$ as the signature.
- Verify: In order to verify a signature $(m,\sigma=(y,c_1,...,c_n),U)$, the verifier performs the following steps:

  (1) Computes $k_i=H_1(ID_i,R_i)$ and

  $$l_i=H_2(m,c_i,ID_i,PK_i) \text{ for } i=1,2,...,n.$$

  (2) Checks whether

  $$\sum_{i=1}^{n}c_i=H_3(m,U,yP+\sum_{i=1}^{n}c_i(l_iT_i+R_i+k_iP_{pub})).$$

  If the equality holds, outputs 1. Otherwise, outputs 0.

• On correctness

$$H_3(m,U,yP+\sum_{i=1}^{n}c_i(l_iT_i+R_i+k_iP_{pub}))$$
$$=H_3(m,U,dP+\sum_{i=1,i\neq s}^{n}c_i(l_iT_i+R_i+k_iP_{pub}))=h=\sum_{i=1}^{n}c_i$$

5 Security of Scheme

In this section, the proposed CLRS scheme is proved to be unforgeable and anonymous against adaptive chosen message attacks in the random oracle model.

Theorem 1. In the random oracle model, if there is a Type I adversary $\mathcal{A}_1$ that can win with advantage ε within time T in the EUF-CLRS Game I, after making at most $q_{H_i}H_i(i=1,2,3)$ queries, q_U user public key queries, q_D partial private key queries, q_R partial public key replacement requests, q_E secret value queries, q_S signature queries, the DL problem can be solved with probability $\frac{\varepsilon^2}{66C_{q_{H_3}}^{n}}\frac{1}{q_U}$ within time $2T+(2q_U+q_D+(3n+1)q_S)T_s$, where n is the size of signer set and T_s is the time for a scalar point multiplication in G.

Proof. Suppose the challenger $\mathcal{C}$ receives a random instance (P,aP) of the DL problem and has to compute a. $\mathcal{C}$ will run $\mathcal{A}_1$ as a subroutine and act as $\mathcal{A}_1$'s challenger in the Game I.

Initialization. $\mathcal{C}$ runs the setup algorithm with a parameter v, then gives $\mathcal{A}_1$ the system parameters $params=\{G,q,P,P_{pub}=xP,H_1,H_2,H_3\}$.

Queries. $\mathcal{A}_1$ will query user public key before an identity ID_i is used in any other queries. $\mathcal{C}$ sets several lists to store the queries and answers, all of the lists are initially empty.

• User public key queries: $\mathcal{C}$ maintains the list L_U of tuple (ID_i,t_i,r_i). When $\mathcal{A}_1$ issues a user public key query for ID_i, $\mathcal{C}$ responds as follows:

At the f^{th} query, randomly picks $t_f\in Z_q^*$, sets $ID_f=ID^{\Diamond}$ and $PK^{\Diamond}=(t_fP,aP)$. For $i=f$, $\mathcal{C}$ randomly picks $t_i,r_i\in Z_q^*$ and returns $PK_i=(t_iP,r_iP)$, then the query and the answer will be stored in the list L_U.

• H_1 queries: $\mathcal{C}$ maintains the list L_1 of tuple (α_i,k_i). When $\mathcal{A}_1$ issues a query $H_1(\alpha_i)$, $\mathcal{C}$ randomly picks $k_i\in Z_q^*$, sets $H_1(\alpha_i)=k_i$ and adds (α_i,k_i) to list L_1.

• H_2 queries: $\mathcal{C}$ maintains the list L_2 of tuple (β_i,l_i). When $\mathcal{A}_1$ issues a query $H_2(\beta_i)$, $\mathcal{C}$ randomly picks $l_i\in Z_q^*$, sets $H_2(\beta_i)=l_i$ and adds (β_i,l_i) to list L_2.

• H_3 queries: $\mathcal{C}$ maintains the list L_3 of tuple (γ_i,h_i). When $\mathcal{A}_1$ issues a query $H_3(\gamma_i)$, $\mathcal{C}$ randomly picks $h_i\in Z_q^*$, sets $H_3(\gamma_i)=h_i$ and adds (γ_i,h_i) to list L_3.

• Partial private key queries: $\mathcal{C}$ maintains the list L_D of tuple (ID_i,D_i). When $\mathcal{A}_1$ issues a partial private key query for identity ID_i. If $ID_i=ID^{\Diamond}$, $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ finds the tuple (ID_i,t_i,r_i) in list L_U, gives the D_i by calling the partial private key extract algorithm and adds (ID_i,D_i) to list L_D.

• Partial public key replacement requests: $\mathcal{C}$ maintains the list L_R of tuple (ID_i,T_i,T_i'). When $\mathcal{A}_1$ issues a partial public key replacement request for identity ID_i with a new T_i'. $\mathcal{C}$ replaces the current partial public key value T_i with T_i' and adds (ID_i,T_i,T_i') to list L_R.

• Secret value queries: When $\mathcal{A}_1$ issues a secret value query for an identity ID_i, $\mathcal{C}$ finds (ID_i,t_i,r_i) in list L_U, responds with t_i and adds (ID_i,t_i) to list L_E.

• Signature queries: When $\mathcal{A}_1$ submits a tuple (m,U), $\mathcal{C}$ outputs a signature as follows:

If there exists a user $ID_s\in W$ such that $ID_s\neq ID^{\Diamond}$ and $ID_s\notin L_R$, $\mathcal{C}$ gives a signature σ by calling the signing algorithm, where ID_s is the actual signer. Otherwise, $\mathcal{C}$ does as follows:

(1) Computes $k_i=H_1(ID_i,R_i)$ for $i=1,2,...,n$.

(2) Randomly chooses $y,c_i\in Z_q^*$ for $i=1,2,...,n$.

(3) Computes $l_i=H_2(m,c_i,ID_i,PK_i)$ for $i=1,2,...,n$.

(4) Adds $\sum_{i=1}^{n}c_i=H_3(m,U,yP+\sum_{i=1}^{n}c_i(l_iP_i+R_i+k_iP_{pub}))$ to list L_3. If collision occurs, repeats the steps 2-4.

(5) outputs $\sigma=(y,c_1,...,c_n)$ as the signature.

Forge. $\mathcal{A}_1$ outputs a tuple (m^*,σ^*,U^*), and fulfills the requirements as defined in the Game I. Where $\sigma^*=(y^*,c_1^*,...,c_n^*)$ and $U^*=W^*\cup\{PK_i^*:PK_i^*\in W^*\}$.

Solve DL problem. By using forking lemma for ring signature scheme [9], after replays $\mathcal{A}_1$ with the same random tape except the result returned by H_3 query, $\mathcal{C}$ gets two valid ring signatures with probability $\frac{\varepsilon^2}{66C_{q_{H_3}}^{n}}$: $\sigma^*=(y^*,c_1^*,...,c_n^*)$ and $\sigma^{*\prime}=(y^{*\prime},c_1^{*\prime},...,c_n^{*\prime})$. It follows that $c_s^*\neq c_s^{*\prime}$ and $c_i^*=c_i^{*\prime}$ for $i\neq s$. If $ID_s^*=ID^{\Diamond}$, then $y^*=d^*-c_s^*(l_s^*t_s^*+a+k_s^*x)$, $y^{*\prime}=d^*-c_s^{*\prime}(l_s^{*\prime}t_s^*+a+k_s^*x)$. $\mathcal{C}$ finds $(ID_s^*,*,t_s^*)$ and (ID_s^*,R_s^*,k_s^*) in list L_U and L_1,

respectively. Follow on, $\mathcal{C}$ finds $(m^*, c_s^*, ID_s^*, PK_s^*, l_s^*)$ and $(m^*, c_s^{*\prime}, ID_s^*, PK_s^*, l_s^{*\prime})$ in list L_2, then solves DL problem by computing:

$$a = [(y^* - y^{*\prime}) - (c_s^{*\prime} l_s^{*\prime} - c_s^* l_s^*) t_s^*](c_s^{*\prime} - c_s^*)^{-1} - k_s^* x .$$

Probability. We denote some events as follows: π_1 : the partial private key of $ID^\diamond$ was not queried by $\mathcal{A}_1$. π_2: $ID^\diamond \in W$. π_3: $ID^\diamond$ is the actual signer. It is easy to get following results:

$$\Pr[\pi_1] = \frac{q_U - q_D}{q_U},\ \Pr[\pi_2 | \pi_1] = \frac{n}{q_U - q_D},$$

$$\Pr[\pi_3 | \pi_1 \wedge \pi_2] = \frac{1}{n}.$$

$$\begin{aligned}\Pr[\mathcal{C}\ success] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\ &= \Pr[\pi_1] \bullet \Pr[\pi_2 | \pi_1] \Pr[\pi_3 | \pi_1 \wedge \pi_2] \\ &= \frac{q_U - q_D}{q_U} \bullet \frac{n}{q_U - q_D} \bullet \frac{1}{n} = \frac{1}{q_U}\end{aligned}$$

Therefore, if $\mathcal{A}_1$ can win with advantage ε within time T in the Game I, then $\mathcal{C}$ can solve the DL problem with the probability $\frac{\varepsilon^2}{66 C_{q_{H_3}}^n} \frac{1}{q_U}$ within $2T + (2q_U + q_D + (3n+1)q_S)T_s$, where n is the size of signer set and T_s is the time for a scalar point multiplication in G.

Theorem 2. In the random oracle model, if there is a super Type II adversary $\mathcal{A}_2$ that can win the EUF-CLRS Game II with advantage ε within time T, after making at most $q_{H_i} H_i (i = 1,2,3)$ queries, q_U user public key queries, q_D partial private key queries, q_R partial public key replacement requests, q_E secret value queries, q_S signature queries, the DL problem can be solved with probability $\frac{\varepsilon^2}{66 C_{q_{H_3}}^n} \frac{1}{q_U}$ within time $2T + (2q_U + q_D + (3n+1)q_S)T_s$, where n is the size of signer set and T_s is the time for a scalar point multiplication in G.

Proof. Suppose the challenger $\mathcal{C}$ receives a random instance (P, aP) of the DL problem and has to compute a. $\mathcal{C}$ will run $\mathcal{A}_2$ as a subroutine and act as $\mathcal{A}_2$'s challenger in the Game II.

Initialization. $\mathcal{C}$ runs the setup algorithm with a parameter v, then gives $\mathcal{A}_2$ the system parameters $params = \{G, q, P, P_{pub} = xP, H_1, H_2, H_3\}$ and master master secret key $msk = \{x\}$.

Queries. $\mathcal{A}_2$ will query user public key before an identity ID_i is used in any other queries. $\mathcal{C}$ sets several lists to store the queries and answers, all of the lists are initially empty.

- User public key queries: $\mathcal{C}$ maintains the list L_U of tuple (ID_i, t_i, r_i). When $\mathcal{A}_2$ issues a user public key query for ID_i, $\mathcal{C}$ responds as follows:

At the f^{th} query, randomly picks $r_f \in Z_q^*$, sets $ID_f = ID^\diamond$ and $PK^\diamond = (aP, r_f P)$. For $i \neq f$, $\mathcal{C}$ randomly picks $t_i, r_i \in Z_q^*$ and returns $PK_i = (t_i P, r_i P)$, then the query and the answer will be stored in the list L_U.

- H_1, H_2, H_3 queries: Same as that in the proof of Theorem 1.
- Partial private key queries: $\mathcal{C}$ maintains the list L_D of tuple (ID_i, D_i). When $\mathcal{A}_2$ issues a partial private key query for identity ID_i. $\mathcal{C}$ finds the tuple (ID_i, t_i, r_i) in list L_U, gives the D_i by calling the partial private key extract algorithm and adds (ID_i, D_i) to list L_D.
- Partial public key replacement requests: Same as that in the proof of Theorem 1.
- Secret value queries: When $\mathcal{A}_2$ issues a secret value query for an identity ID_i. If $ID_i = ID^\diamond$, $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ finds (ID_i, t_i, r_i) in list L_U, responds with t_i and adds (ID_i, t_i) to list L_E.
- Signature queries: Same as that in the proof of Theorem 1.

Forge. $\mathcal{A}_2$ outputs a tuple (m^*, σ^*, U^*), and fulfills the requirements as defined in the Game II. Where $\sigma^* = (y^*, c_1^*, ..., c_n^*)$ and $U^* = W^* \cup \{PK_i^* : PK_i^* \in W^*\}$.

Solve DL problem. By using forking lemma for ring signature scheme [9], after replays $\mathcal{A}_2$ with the same random tape except the result returned by H_3 query, $\mathcal{C}$ gets two valid ring signatures with probability $\frac{\varepsilon^2}{66 C_{q_{H_3}}^n}$: $\sigma^* = (y^*, c_1^*, ..., c_n^*)$ and $\sigma^{*\prime} = (y^{*\prime}, c_1^{*\prime}, ..., c_n^{*\prime})$. It follows that $c_s^* \neq c_s^{*\prime}$ and $c_i^* = c_i^{*\prime}$ for $i \neq s$. If $ID_s^* = ID^\diamond$, then $y^* = d^* - c_s^*(l_s^* a + r_s^* + k_s^* x)$, $y^{*\prime} = d^* - c_s^{*\prime}(l_s^{*\prime} a + r_s^* + k_s^* x)$. $\mathcal{C}$ finds $(ID_s^*, *, r_s^*)$ and (ID_s^*, R_s^*, k_s^*) in list L_U and L_1, respectively. Follow on, $\mathcal{C}$ finds $(m^*, c_s^*, ID_s^*, PK_s^*, l_s^*)$ and $(m^*, c_s^{*\prime}, ID_s^*, PK_s^*, l_s^{*\prime})$ in list L_2, then solves DL problem by computing:

$$a = [(y^* - y^{*\prime}) - (c_s^{*\prime} - c_s^*)(r_s^* + k_{sx}^*)](c_s^{*\prime} l_s^{*\prime} - c_s^* l_s^*)^{-1}.$$

Probability. Without loss of generality, we may assume that $L_E \cap L_R = \varnothing$, and denote some events as follows: π_1: the partial public key of $ID^{\diamond}$ was not replaced and the secret value of $ID^{\diamond}$ was not queried by $\mathcal{A}_2$. π_2: $ID^{\diamond} \in W^*$. π_3: $ID^{\diamond}$ is the actual signer. It is easy to get following results:

$$\Pr[\pi_1] = \frac{q_U - q_R - q_E}{q_U},\ \Pr[\pi_2 | \pi_1] = \frac{n}{q_U - q_R - q_E},$$

$$\Pr[\pi_3 | \pi_1 \wedge \pi_2] = \frac{1}{n}.$$

$$\begin{aligned}\Pr[\mathcal{C}\ success] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\ &= \Pr[\pi_1] \bullet \Pr[\pi_2 | \pi_1] \Pr[\pi_3 | \pi_1 \wedge \pi_2] \\ &= \frac{q_U - q_R - q_E}{q_U} \bullet \frac{n}{q_U - q_R - q_E} \bullet \frac{1}{n} = \frac{1}{q_U}\end{aligned}$$

Therefore, if $\mathcal{A}_2$ can win with advantage ε within time T in the Game II, then $\mathcal{C}$ can solve the DL problem with the probability $\frac{\varepsilon^2}{66 C^n_{q_{H_3}}} \frac{1}{q_U}$ within $2T + (2q_U + q_D + (3n+1)q_S)T_s$, where n is the size of signer set and T_s is the time for a scalar point multiplication in G.

Theorem 3. Our scheme is anonymous.

Proof. $\mathcal{C}$ runs the setup algorithm with a parameter v, and sends the $params = \{G, q, P, P_{pub} = xP, H_1, H_2, H_3\}$ and $msk = \{x\}$ to the adversary $\mathcal{A}$.

First, $\mathcal{A}$ makes queries as those in the proof of Theorem 1.

Follow on, $\mathcal{A}$ submits a tuple (m, U, ID_0, ID_1), where $U = W \cup \{PK_i : ID_i \in W\}$ and $ID_0, ID_1 \in W$. $\mathcal{C}$ selects at random a bit $\mu \in \{0,1\}$ and provides $\mathcal{A}$ with $\sigma = \mathrm{Sign}(m, U, z_\mu, t_\mu)$.

Once again, $\mathcal{A}$ makes queries as those in the proof of Theorem 1.

In the end, $\mathcal{A}$ returns a bit $\mu' \in \{0,1\}$.

For a signature σ generated by sign algorithm, if $ID_i \in W$ is not the actual signer, c_i is chosen independently and distributed uniformly over Z_q^*. Since r_i is chosen uniformly at random from Z_q^*, then $R_i = r_i P$ is distributed uniformly over G. By k_i and l_i are the outputs of hash functions, then k_i and l_i are distributed uniformly over Z_q^*. If ID_s is the actual signer, then d is chosen uniformly at random from Z_q^*. Since h is an output of hash function, then $c_s = h - \sum_{i=1, i \neq s}^{n} c_i$ is distributed uniformly. Since l_s is an output of hash function, it follows that $y = d - c_s(l_s t_s + z_s)$ is also distributed uniformly over Z_q^*. In conclusion, all the mentioned parameters are uniformly distributed. Therefore, $\Pr = [\mu' = \mu] = \frac{1}{2}$. In other words, the advantage of $\mathcal{A}$ is negligible in the Game III.

6 Efficiency and Comparison

In this section, we compared the performance of our scheme with several other schemes. Some notations are defined as follows:

P: a pairing operation.

M_P: a pairing-based scalar multiplication operation.

M_E: an ECC-based scalar multiplication operation.

M_N: a modular exponent operation in Z_N.

H: a one-way hash operation.

Table 2. Cryptographic operation time (in milliseconds)

P	M_P	M_E	M_N	H
20.04	6.38	2.21	5.31	< 0.01

For fairness and reasonableness, third-party data is used to analyze three CLRS schemes. Implementing the related operations on a personal computer (PIV 3GHZ processor with 512MB memory and the Windows XP operation system). He et al. [10] obtained the running time on basic cryptographic operations. For the pairing-based scheme, to achieve 1024-bit RSA level security, a Tate pairing $e: G_1 \times G_1 \to G_2$ was used, where G_1 with the prime order q is an additive group defined on the supersingular elliptic curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2, where the sizes of p and q are 512 bits and 160 bits, respectively. For the ECC-based schemes, to achieve the same security level, they used the parameter secp160r1, recommended by the Certicom Corporation [26], where $p = 2^{160} - 2^{31} - 1$. The running times are listed in Table 2.

We use a simple method to evaluate the computation cost. For example, Chang et al.'s [5] scheme requires $3n+3$ pairing-based scalar multiplication operations, 4 pairing operations and 2 hash function operations. So the resulting computation time is $6.38 \times (4n+3) + 20.04 \times 4 + 0.01 \times 2 = 25.52n + 99.32$ ms. In order to facilitate the comparison, we let $n = 10$, then the computation time is $25.52 \times 10 + 99.32 = 354.52$ ms. The detailed comparison results of several different CLRS schemes are illustrated in Table 3 (Figure. 1).

Table 3. Comparison of several CLRS schemes

Scheme	Sign	Verify	Time ($n=10$)
Zhang [24]	$2P+(3n+1)M_P+nH$	$3P+2nM_P+(n+1)H$	425.79
Chow [4]	$P+2nM_P+(n+1)H$	$nP+(n+1)M_P+H$	418.34
Chang [5]	$2P+(2n+2)M_P+H$	$2P+(2n+1)M_P+H$	354.52
Zhang [25]	$(2n+4)M_P+2nH$	$3P+2nM_P+2nH$	341.24
Deng [7]	$nM_E+(n+2)M_N+(n+1)H$	$(n+1)(M_E+M_N+H)$	168.76
Our scheme	$(3n-2)M_E+2nH$	$(3n+1)M_E+2nH$	130.79

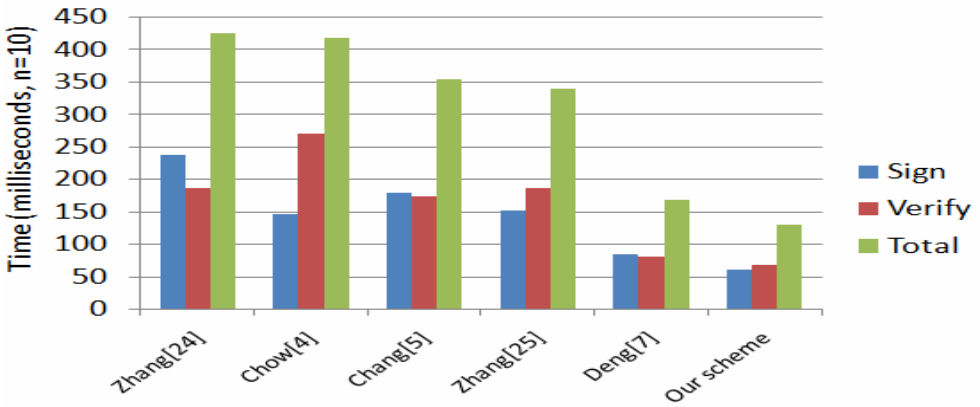

Figure 1. Computation cost

7 Conclusion

Most CLRS schemes currently known use bilinear pairings. Scientific researchers get some good results in speeding up the computation of pairings in recent years. However, the computation cost of the pairings is much higher than that of the scalar multiplication over the elliptic curve group. Therefore, it is quite significant to construct efficient CLRS scheme without bilinear pairings. In this paper, we propose a new CLRS scheme and gave the proof of security in the random oracle model. Our scheme does not require pairing operation, the analysis on performance shows that it is more efficient than previous ones.

Acknowledgments

The authors are 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 No. 61962011, the Innovation Group Major Research Projects of Department of Education of Guizhou Province under Grant No. KY [2016] 026, the Guizhou Provincial Science and Technology Foundation under Grant No. [2019] 1434.

References

[1] S. S. Al-Riyami, K. G. Paterson, Certificateless Public Key Cryptography, *Advances in Cryptology-Asiacrypt*, LNCS, Vol. 2894, Taipei, Taiwan, 2003, pp. 452-473.

[2] A. Bender, J. Katz, R. Morselli, Ring Signatures: Stronger Definitions, and Constructions without Random Oracles, *Journal of Cryptology*, Vol. 22, No. 1, pp. 114-138, December, 2008.

[3] S. S. M. Chow, S. M. Yiu, L. C. K. Hui, Efficient Identity Based Ring Signature, *International Conference on Applied Cryptography and Network Security*, LNCS, Vol. 3531, New York, USA, 2005, pp. 499-512.

[4] S. S. M. Chow, W. S. Yap, Certificateless Ring Signature, *Cryptology ePrint Archive*, Report 2007/236, June, 2007.

[5] S. Chang, D. S. Wong, Y. Mu, Z. F. Zhang, Certificateless Threshold Ring Signature, *Information Sciences*, Vol. 179, No. 20. pp. 3685-3696, September, 2009.

[6] L. Deng, J. Zeng, Two New Identity-based Threshold Ring Signature Schemes, *Theoretical Computer Science*, Vol. 535, pp. 38-45, May, 2014.

[7] L. Deng, Certificateless Ring Signature Based on RSA Problem and DL Problem, *RAIRO-Theoretical Informations and Applications*, Vol. 49, No. 4, pp. 307-318, October-November, 2015.

[8] J. Herranz, Identity-based Ring Signatures from RSA, *Theoretical Computer Science*, Vol. 389, No. 1-2, pp. 100-117, December, 2007.

[9] J. Herranz, G. Saez, Forking Lemmas for Ring Signature

Schemes, *International Conference on Cryptology in India*, LNCS, Vol. 2904, New Delhi, India, 2003, pp. 266-279.

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

[11] 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.

[12] D. He, B. Huang, J. Chen, New Certificateless Short Signature Scheme, *IET Information Security*, Vol. 7, No. 2, pp. 113-117, June, 2013.

[13] L. Nguyen, Accumulators from Bilinear Pairings and Applications, *Cryptographers' Track at the RSA Conference*, LNCS, Vol. 3376, San Francisco, USA, 2005, pp. 275-292.

[14] R. L. Rivest, A. Shamir, Y. Tauman, How to Leak a Secret, *Advances in Cryptology-Asiacrypt*, LNCS, Vol. 2248, Gold Coast, Australia, 2001, pp. 552-565.

[15] A. Shamir, Identity-based Cryptosystems and Signature Schemes, *Advances in Cryptology*, LNCS, Vol. 196, Santa Barbara, California, USA, 1984, pp. 47-53.

[16] K. Shim, Security Models for Certificateless Signature Schemes Revisited, *Information Sciences*, Vol. 296, pp. 315-321, March, 2015.

[17] J. Tsai, N. Lo, T. Wu, Weaknesses and Improvements of an Efficient Certificateless Signature Scheme without Using Bilinear Pairings, *International Journal of Communication Systems*, Vol. 27, No. 7, pp. 1083-1090, July, 2014.

[18] J. Tsai, A New Efficient Certificateless Short Signature Scheme Using Bilinear Pairings, *IEEE Systems Journal*, Vol. 11, No. 4, pp. 2395-2402, December, 2017.

[19] L. Wang, K. Chen, Y. Long, H. Wang, An Efficient Pairing-free Certificateless Signature Scheme for Resource-limited Systems, *Science China Information Sciences*, Vol. 60, No. 11, pp. 119102: 1-119102: 3, November, 2017,

[20] H. Wang, Certificateless Ring Signature Scheme from Anonymous Subsets, *International Conference on Multimedia Information Networking and Security*, Nanjing, China, 2010. pp. 413-417.

[21] H. Wang, S. Han, A Provably Secure Threshold Ring Signature Scheme in Certificateless Cryptography, *International Conference of Information Science and Management Engineering*, Shanxi, China, 2010, pp. 105-108.

[22] 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.

[23] T. Yuen, J. Liu, M. Au, W. Susilo, J. Zhou, Efficient Linkable and/or Threshold Ring Signature without Random Oracles, *Computer Journal*, Vol. 56, No. 4, pp. 407-421, April, 2013.

[24] L. Zhang, F. Zhang, W. Wu, A Provably Secure Ring Signature Scheme in Certificateless Cryptography, *International Conference on Provable Security*, LNCS, Vol. 4784, Wollongong, Australia, 2007, pp. 103-121.

[25] Y. Zhang, J. Zeng, W. Li, H. Zhu, A Certificateless Ring Signature Scheme with High Efficiency in the Random Oracle Model, *Mathematical Problems in Engineering*, Vol. 2017, Article ID 7696858, June, 2017.

[26] The Certicom Corporation, SEC 2: Recommended Elliptic Curve Domain Parameters, www.secg.org/collateral/sec2final.pdf.

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 Mathematical Sciences, Guizhou Normal University, Guiyang, PR China. His recent research interests include cryptography and information safety.

Siwei Li received his B.S. from Guizhou University, Guiyang, PR China, in 2009; M.S. from Guizhou University, Guiyang, PR China, in 2012; She is now a lecturer in the Graduate School, Guizhou Normal University, Guiyang, PR China. His recent research interests include data management and security.

Huawei Huang received his Ph.D. from Xidian University, Xi'an, PR China, in 2008. He is currently an associate professor in the School of Mathematical Sciences, Guizhou Normal University, Guiyang, PR China. His recent research interests include algebra and cryptograph.

Yuhong Jiang received her B.S. from Yangtze Normal University, Chongqing, PR China, in 2017; She is now a graduate student in the School of Mathematical Sciences, Guizhou Normal University in China.. Her recent research interests include cryptography protocol and information safety.

Bingqin Ning received her B.S. from Xiangnan University, Chenzhou, PR China, in 2017; She is now a graduate student in the School of Mathematical Sciences, Guizhou Normal University in China. Her recent research interests include cryptography protocol and information safety.