

CERTIFICATELESS RING SIGNATURE BASED ON RSA PROBLEM AND DL PROBLEM

LUNZHI DENG¹

Abstract. Certificateless public key cryptography solves the certificate management problem in the traditional public key cryptography and the key escrow problem in identity-based cryptography. RSA is a key cryptography technique and provides various interfaces for the applied software in real-life scenarios. To the best of our knowledge, all of the known certificateless ring signature schemes employed bilinear pairings. But the computation cost of the pairings is much higher than that of the exponentiation in a RSA group. In this paper, we present the first certificateless ring signature scheme without pairing and prove the security in the random oracle model. The security of the scheme is closely related to the RSA problem and the discrete logarithm (DL) problem.

Mathematics Subject Classification. 94A60.

1. INTRODUCTION

To solve the certificate management problem in traditional public key infrastructure (PKI), Shamir [12] introduced identity-based public key cryptography, which needs a trusted private key generator (PKG) to generate a private key for a user according to his identity. Therefore the key escrow problem arises.

To solve the two problems, Al-Riyami *et al.* [1] introduced certificateless public key cryptography, which needs a semi-trusted key generation center (KGC) to create user's partial private key with respect to user's identity. A user's full private key including two parts: partial private key and the secret value selected by himself.

Keywords and phrases. Certificateless cryptography, ring signature, RSA, DL problem.

¹ School of Mathematical Sciences, Guizhou Normal University, Guiyang 550001, P.R. China.
denglunzhi@163.com

1.1. RELATED WORK

In 2001, Rivest *et al.* [11] introduced ring signature. In this setting, a signer first chooses several members to form a group without the agreement of the other members, then generates a signature which can convince an arbitrary verifier that the message was signed by someone in the group, but no one can identify the real signer among group members.

Many practical ring signature schemes have been proposed since ring signature was formalized, such as ring signature without random oracles [2], linkable ring signature scheme [14], identity-based ring signature [5], and ring signature with constant-size signature [10].

Shamir [12] proposed the first identity-based signature scheme from the RSA primitive. Herranz [8] constructed an identity-based ring signatures from RSA. Zhang and Mao [15] presented an efficient RSA-based certificateless signature scheme. Dong and Lu [7] proposed an improved RSA-based certificateless signature scheme.

There are only several work published on certificateless ring signature (CLRS) schemes [3,4,13,16]. Chow *et al.* [4] presented a security model and a CLRS scheme, it does not capture the type of attacks where the adversary has compromised some members' partial keys and some others' secret keys. Their scheme requires n pairing operations and $3n + 1$ exponentiation operations. Zhang *et al.* [16] constructed another CLRS scheme, which requires 5 pairing operations and $4n + 3$ exponentiation operations. Chang *et al.* [3] proposed formal security definition and a concrete CLRS scheme, which requires 4 pairing operations and $4n + 4$ exponentiation operations.

1.2. OUR CONTRIBUTIONS

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

- The scheme is security under the strong security model. Namely, in the scheme, the super Type I/II adversary can obtain the valid signatures for the replaced public key, without additional submission.
- All other CLRS schemes need pairing operations. Our scheme is first CLRS scheme from RSA without pairing operations.

2. PRELIMINARIES

Definition 2.1. Let $N = pq$, where p and q are two k -bit prime numbers. Let d be a random prime number, greater than 2^l for some fixed parameter l , such that $\gcd(d, \varphi(N)) = 1$. Given $Y \in Z_N^*$, RSA problem is to find $X \in Z_N^*$ such that $X^d = Y \pmod N$.

Definition 2.2. Let $\mathcal{G} = (E, +)$, where E is an elliptic curve over a finite field F_p , $P \in E$ is a point having prime order $d = |E|/2$. Let $G = \langle P \rangle \leq \mathcal{G}$, given $xP \in G$, the discrete logarithm (DL) problem is to compute x .

2.1. MODEL OF CERTIFICATELESS RING SIGNATURE

A CLRS scheme consists of the following six algorithms:

- **Setup:** Given a security parameter k , key generate center (KGC) generates the system parameters $params$ and the master secret key msk .
- **Partial-Private-Key-Extract:** Given the user's identity $ID_i \in \{0, 1\}^*$, KGC generates the partial private key D_i .
- **Secret-Value-Set:** The user ID_i selects a secret value t_i .
- **User-Public-Key-Generate:** The user ID_i sets his public key P_i .
- **Sign:** Given a message M , signer chooses $n - 1$ other users to form group W including himself, then gives a signature σ on M on the behalf of the group W .
- **Verify:** On receiving the signature (σ, M, W) , anyone can verify it. Then outputs 1 or 0, depending on whether σ is a valid ring signature on message M .

Definition 2.3. 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. Now we illustrate the first game performed between a challenger \mathcal{C} and a Type I adversary \mathcal{A}_1 for a CLRS.

Initialization. \mathcal{C} runs the setup algorithm to generate msk and $params$. \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 queries:** \mathcal{A}_1 can ask for the values of the hash functions for any input.
- **User public key queries:** \mathcal{A}_1 requests the public key of a user ID_i , \mathcal{C} returns the corresponding public key P_i .
- **Partial private key queries:** \mathcal{A}_1 requests the partial private key of a user ID_i , \mathcal{C} responds with the partial private key D_i .
- **User public key replacements:** \mathcal{A}_1 supplies a new public key value P'_i with respect to a user ID_i . \mathcal{C} replaces the current public key with the value P'_i .
- **Secret value queries:** \mathcal{A}_1 requests the secret value of a user ID_i whose public key was not replaced, \mathcal{C} returns the secret value t_i . If a user's public key was replaced, \mathcal{A}_1 can not request the corresponding secret value.
- **Sign queries:** \mathcal{A}_1 supplies a group of n users' identities W , and a message M , \mathcal{C} outputs a signature.

Forge. \mathcal{A}_1 outputs a new triple (σ, M, W) . The adversary wins if the result of $\text{Verify}(\sigma, M, W)$ is the symbol 1 and the following conditions hold:

1. \mathcal{A}_1 can not query the partial private of anyone in W .
2. The forged signature (σ, M, W) is not from signing query.

The advantage of \mathcal{A}_1 is defined as: $\text{Adv}_{\mathcal{A}_1}^{\text{UNF-CLRS}} = \text{Pr}[\mathcal{A}_1 \text{ win}]$.

Game II. A Type II adversary \mathcal{A}_2 plays the second game with a challenger \mathcal{C} as follows.

Initialization. \mathcal{A}_2 runs the setup algorithm to obtain msk and params . \mathcal{A}_2 gives params and msk to \mathcal{C} .

Query. \mathcal{A}_2 makes a polynomially bounded number of queries as those in Game I. Obviously, \mathcal{A}_2 can compute the partial private key of any user by itself with the master secret key.

Forge. \mathcal{A}_2 outputs a new triple (σ, M, W) . The adversary wins if the result of $\text{Verify}(\sigma, M, W)$ is the symbol 1 and the following conditions hold:

1. \mathcal{A}_2 can not query the secret value of anyone in W .
2. \mathcal{A}_2 can not replace the user public key of anyone in W .
3. The forged signature (σ, M, W) is not from signing query.

The advantage of \mathcal{A}_2 is defined as: $\text{Adv}_{\mathcal{A}_2}^{\text{UNF-CLRS}} = \text{Pr}[\mathcal{A}_2 \text{ win}]$.

Remark 2.4. For a forge signature (σ, M, W) , \mathcal{A}_1 can know the all users' secret values in W , however, he does not know the partial private key of anyone in W . On the other hand, \mathcal{A}_2 can know the all users' partial private keys in W , however, he does not know the secret value of anyone in W .

Definition 2.5. 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 , then gives params and msk to \mathcal{A} .

Phase 1. \mathcal{A} may adaptively make a polynomially bounded number of queries as those in Game I.

Challenge. \mathcal{A} outputs a group of n users' identities W , two different members $ID_0, ID_1 \in W$ and a message M . \mathcal{C} randomly chooses a bit $\mu \in \{0, 1\}$ and provides \mathcal{A} with $\sigma = \text{Sign}(M, W, D_\mu, t_\mu)$.

Phase 2. \mathcal{A} continues to probe \mathcal{C} with the same type of queries made in Phase 1.

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

The advantage of \mathcal{A} is defined as: $Adv_{\mathcal{A}}^{ANO-CLRS} = |2Pr[\mu' = \mu] - 1|$.

3. OUR SCHEME

- Setup: Given the security parameter k , KGC generates two random k -bit prime numbers p and q , then computes $N = pq$. For some fixed parameter l (for example $l = 200$), chooses a prime number d satisfying $2^l < d < 2^{l+1}$ and $\text{gcd}(d, \varphi(N)) = 1$. Then it chooses group G of prime order d as defined in Definition 2.2, a generator P of G and computes $e = d^{-1} \pmod{\varphi(N)}$. Furthermore, KGC chooses three cryptographic hash functions: $H_1 : \{0, 1\}^* \rightarrow Z_N^*$, $H_2, H_3 : \{0, 1\}^* \rightarrow Z_d^*$. Finally, KGC outputs the set of public parameters: $params = \{N, d, G, P, H_1, H_2, H_3\}$. The master secret key is $msk = (p, q, e)$.
- Partial private key extract: For an identity $ID_i \in \{0, 1\}^*$, KGC computes $D_i = Q_i^e, Q_i = H_1(ID_i)$ and sends D_i to the user ID_i via a secure channel.
- Secret value set: the user ID_i randomly chooses $t_i \in Z_d^*$.
- User public key generate: the user ID_i sets his public key as $P_i = t_i P$.
- Sign: Let $R = W \cup \{P_i : ID_i \in W\}$, $W = \{ID_1, \dots, ID_n\}$ is the set of n identities and $ID_s \in W$ is the actual signer, the following steps are carried out:

1. Randomly chooses $A_i \in Z_N^*, c_i \in Z_d^*$, computes

$$h_i = H_2(M, R, P_i, ID_i, A_i, c_i), i = 1, 2, \dots, s - 1, s + 1, \dots, n.$$

2. Randomly chooses $r_1 \in Z_d^*, A \in Z_N^*$, computes

$$A_s = A^d Q_s^{r_1} \prod_{i=1, i \neq s}^n (A_i^{-1} Q_i^{-h_i}).$$

3. Randomly chooses $r_2 \in Z_d^*$, computes

$$c = H_3 \left(M, R, r_2 P + \sum_{i=1, i \neq s}^n c_i P_i, \bigcup_{i=1}^n \{A_i\} \right).$$

4. Computes $c_s = c - \sum_{i=1, i \neq s}^n c_i \pmod{d}$, $h_s = H_2(M, R, P_s, ID_s, A_s, c_s)$.
5. Computes $z = r_2 - c_s t_s \pmod{d}$, $V = AD_s^{r_1 + h_s}$.
6. The signature is $\sigma = \{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}$.

- Verify: A verifier can check whether a signature

$$\sigma = \left\{ z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\} \right\}$$

on the message M is given by someone in the W as follows:

1. Checks if $\sum_{i=1}^n c_i = H_3(M, R, zP + \sum_{i=1}^n c_i P_i, \bigcup_{i=1}^n \{A_i\})$. Proceeds if the equality holds, rejects otherwise.
 2. Computes $h_i = H_2(M, R, P_i, ID_i, A_i, c_i)$, $i = 1, 2, \dots, n$.
 3. Checks whether $V^d = \prod_{i=1}^n (A_i Q_i^{h_i})$. If the equality holds, outputs 1. Otherwise, outputs 0.
- On correctness, we have

$$\begin{aligned} \sum_{i=1}^n c_i &= c = H_3 \left(M, R, r_2 P + \sum_{i=1, i \neq s}^n c_i P_i, \bigcup_{i=1}^n \{A_i\} \right) \\ zP + \sum_{i=1}^n c_i P_i &= (r_2 - c_s t_s) P + \sum_{i=1}^n c_i P_i = r_2 P + \sum_{i=1, i \neq s}^n c_i P_i \\ \prod_{i=1}^n (A_i Q_i^{h_i}) &= A^d Q_s^{r_1 + h_s} = (A D_s^{r_1 + h_s})^d = V^d \end{aligned}$$

4. SECURITY OF PROPOSED SCHEME

Theorem 4.1. *The scheme is unforgeable against the type I adversary \mathcal{A}_1 in randomly oracle model if the RSA problem is hard.*

Proof. Suppose that the challenger \mathcal{C} receives a random instance (N, d, Y) of the RSA problem and has to find an element $X \in Z_N^*$ such that $X^d = Y$. \mathcal{C} will run \mathcal{A}_1 as a subroutine and act as \mathcal{A}_1 's challenger in the Game I. \square

Initialization. At the beginning of the game, \mathcal{C} runs the setup algorithm with the parameter k , gives \mathcal{A}_1 the system parameters: $params = \{N, d, G, P, H_1, H_2, H_3\}$.

Query. Without loss of generality, it is assumed that all the queries are distinct and \mathcal{A}_1 will ask for $H_1(ID_i)$ before ID_i is used in any other queries. \mathcal{A}_1 will set several lists to store the queries and answers, all of the lists are initially empty.

- H_1 queries: \mathcal{C} maintains the list L_1 of tuple (ID_i, B_i) . When \mathcal{A}_1 makes a query $H_1(ID_i)$, \mathcal{C} responds as follows:
At the j th H_1 query, \mathcal{C} sets $H_1(ID^*) = Y$. For $i \neq j$, \mathcal{C} randomly picks $B_i \in Z_N^*$ and sets $H_1(ID_i) = B_i^d$, the query and answer will be stored in the list L_1 .
- H_2 queries: \mathcal{C} maintains the list L_2 of tuple (α_i, h_i) . When \mathcal{A}_1 makes a query $H_2(\alpha_i)$, \mathcal{C} randomly picks $h_i \in Z_d^*$, sets $H_2(\alpha_i) = h_i$ and adds (α_i, h_i) to list L_2 .

- H_3 queries: \mathcal{C} maintains the list L_3 of tuple (β_i, c_i) . When \mathcal{A}_1 makes a query $H_3(\beta_i)$. \mathcal{C} randomly picks $c_i \in Z_d^*$, sets $H_3(\beta_i) = c_i$ and adds (β_i, c_i) to list L_3 .
- User public key queries: \mathcal{C} maintains the list L_U of tuple (ID_i, t_i) . When \mathcal{A}_1 makes a user public key query for ID_i , \mathcal{C} randomly picks $t_i \in Z_d^*$, returns $P_i = t_i P$ and adds (ID_i, t_i) to list L_U .
- Partial private key extraction queries: \mathcal{C} maintains the list L_D of tuple (ID_i, D_i) . When \mathcal{A}_1 makes partial private key query for ID_i . If $ID_i = ID^*$, \mathcal{C} fails and stops. Otherwise \mathcal{C} finds the tuple (ID_i, B_i) in list L_1 , responds with $D_i = B_i$ and adds (ID_i, D_i) to list L_D .
- User public key replacement requests: \mathcal{C} maintains the list L_R of tuple (ID_i, P_i, P'_i) . When \mathcal{A}_1 makes a user public key replacement request for ID_i with a new valid public key value P'_i . \mathcal{C} replaces P_i with P'_i and adds (ID_i, P_i, P'_i) to list L_R .
- Secret value queries: \mathcal{C} maintains the list L_E of tuple (ID_i, t_i) . When \mathcal{A}_1 makes secret value query for ID_i . \mathcal{C} checks list L_U , if the tuple (ID_i, t_i) is found in a list L_U , \mathcal{C} responds with t_i . Otherwise \mathcal{C} randomly picks $t_i \in Z_d^*$, responds with t_i and adds (ID_i, t_i) to list L_E .
- Sign queries: when \mathcal{A}_1 supplies a message M and a set $R = W \cup \{P_i : ID_i \in W\}$, where $W = \{ID_1, \dots, ID_n\}$ is the set of n users' identities. \mathcal{C} outputs a signature as follow:

If there exists a user $ID_s \in W$ such that $ID_s \neq ID^*$ 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. Randomly chooses $A_i \in Z_N^*, c_i \in Z_d^*$, computes

$$h_i = H_2(M, R, P_i, ID_i, A_i, c_i), i = 1, 2, s - 1, s + 1, \dots, n.$$

2. Randomly chooses $z, c_s \in Z_d^*$, computes $T = zP + \sum_{i=1}^n c_i P_i$.
3. Randomly chooses $A \in Z_N^*, h_s \in Z_d^*$, computes

$$A_s = A^d Q_s^{-h_s} \prod_{i=1, i \neq s}^n (A_i^{-1} Q_i^{-h_i}), V = A.$$

4. Stores the relations

$$\sum_{i=1}^n c_i = H_3 \left(M, R, T, \bigcup_{i=1}^n \{A_i\} \right), h_s = H_2(M, R, P_s, ID_s, A_s, c_s).$$

If collision occurs, repeats the steps (1)–(4).

5. Outputs the signature $\sigma = \{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}$

Forge. \mathcal{A}_1 outputs a forged signature $\sigma = \{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}$ on message M^* , which is signed by someone in the W and fulfills the following conditions:

1. \mathcal{A}_1 can not query the partial private key of anyone in W .
2. The forged signature $(\sigma = \{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}, M^*, W)$ is not from signing query.

Solve RSA problem. By the forking lemma for ring signature scheme [9], if \mathcal{A}_1 can give a valid forged signature with probability $\varepsilon \geq \frac{7C_{qH_2}^n}{2^k}$ within time T in the above interaction, then there exists another algorithm \mathcal{A}'_1 , which can output two signed messages with at least $\frac{\varepsilon^2}{66C_{qH_2}^n}$ probability within time $2T$. Therefore we get two signatures: $\{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}$ and $\{z, V', \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}$. To do so we keeps all the random tapes in two invocations of \mathcal{A}'_1 the same except the λ th result returned by H_2 query of the forged messages, so we have $h_\lambda \neq h'_\lambda$ and $h_i = h'_i$ for $i \neq \lambda$. If ID^* is the actual signer and $s = \lambda$, we can solve the RSA problem as follows: the relation becomes $(VV^{-1})^d = Y^{h'_s - h_s} \pmod N$. Since $h_s, h'_s \in Z_d^*$, we have that $|h'_s - h_s| < d$. By the element d is a prime number, then $\gcd(d, h'_s - h_s) = 1$. This means that there exist two integers a and b such that $ad + b(h'_s - h_s) = 1$. Finally, the value $X = (VV^{-1})^b Y^a \pmod N$ is the solution of the given instance of the RSA problem. In effect, we have $X^d = (VV^{-1})^{db} Y^{da} = Y^{b(h'_s - h_s)} Y^{da} = Y^{ad + b(h'_s - h_s)} = Y$.

Probability. Let $q_{H_i} (i = 1, 2)$, q_D , q_U and q_S be the numbers of $H_i (i = 1, 2)$ queries, partial private key queries, user public key queries and signing queries, respectively.

The probability that \mathcal{C} does not fail during the queries is $\frac{q_{H_1} - q_D}{q_{H_1}}$. The probability that ID^* belongs to the group W is $\frac{n}{q_{H_1} - q_D}$. The probability that ID^* is the actual signer is $\frac{1}{n}$. The probability that $s = \lambda$ is $\frac{1}{n}$. So the combined probability is $\frac{q_{H_1} - q_D}{q_{H_1}} \cdot \frac{n}{q_{H_1} - q_D} \cdot \frac{1}{n} \cdot \frac{1}{n} = \frac{1}{n \cdot q_{H_1}}$.

Therefore, if the adversary \mathcal{A}_1 can win the EUF-CLRS Game I with advantage ε and within time T , then \mathcal{C} can solve the RSA problem with the probability $\frac{\varepsilon^2}{66C_{qH_2}^n} \cdot \frac{1}{n \cdot q_{H_1}}$ within time $2T + (q_{H_1} + q_U + (2n + 2)q_S)T_e$ (where T_e denotes the time for a exponentiation in G).

Theorem 4.2. *The scheme is unforgeable against the type II adversary \mathcal{A}_2 in randomly oracle model if the DL problem is hard.*

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

Initialization. At the beginning of the game, \mathcal{A}_2 runs the setup algorithm with the parameter k , gives \mathcal{C} the system parameters $params = \{N, d, G, P, H_1, H_2, H_3\}$ and master secret key $msk = (p, q, e)$

Query. Without loss of generality, it is assumed that all the queries are distinct and \mathcal{A}_2 will query the user public key for identity ID_i before ID_i is used in any other queries. \mathcal{A}_2 will set 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) . When \mathcal{A}_2 makes a user public key query for ID_i , \mathcal{C} responds as follows:
At the j th query, \mathcal{C} set $ID_j = ID^*$ and $P^* = xP$. For $i \neq j$, \mathcal{C} randomly picks $t_i \in Z_d^*$ and returns $P_i = t_iP$, the tuple (ID_i, t_i) will be stored in the list L_U .
- H_1 queries: \mathcal{C} maintains the list L_1 of tuple (ID_i, B_i) . When \mathcal{A}_2 makes a query $H_1(ID_i)$, \mathcal{C} randomly picks $B_i \in Z_N^*$, sets $H_1(ID_i) = B_i^d$ and adds (ID_i, B_i) to list L_1 .
- H_2 and H_3 queries: Same as those in the proof of Theorem 4.1.
- Partial private key queries: \mathcal{A}_2 can compute the partial private keys of any identities by itself with the master secret key.
- User public key replacement requests: Same as those in the proof of Theorem 4.1.
- Secret value queries: \mathcal{C} maintains the list L_E of tuple (ID_i, t_i) . When \mathcal{A}_2 makes a secret value query for identity ID_i . If $ID_i = ID^*$, \mathcal{C} fails and stops. Otherwise, \mathcal{C} finds the tuple (ID_i, t_i) in list L_U , responds with t_i and adds (ID_i, t_i) to list L_E .
- Sign queries: Same as those in the proof of Theorem 4.1.

Forge. \mathcal{A}_2 outputs a forged signature $\sigma = \{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}$ on message M^* , which is signed by someone in the W , and fulfills the following conditions:

1. \mathcal{A}_2 cannot replace any user public key and query the secret value of anyone in W .
2. The forged signature $(M^*, W, \sigma = \{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\})$ is not from signing query.

Solve DL problem. By the forking lemma for ring signature scheme [9], if \mathcal{A}_2 can give a valid forged signature with probability $\varepsilon \geq \frac{7C^n}{2^k q_{H_3}}$ within time T in the above interaction, then there exists another algorithm \mathcal{A}'_2 , which can output two signed messages with at least $\frac{\varepsilon^2}{66C^n q_{H_3}}$ probability within time $2T$. Therefore we get two signatures: $\{z, V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c_i\}\}$ and $\{z', V, \bigcup_{i=1}^n \{A_i\}, \bigcup_{i=1}^n \{c'_i\}\}$. To do so we keeps all the random tapes in two invocations of \mathcal{A}'_2 the same except the result returned by H_3 query of the forged message, so we have $c_s \neq c'_s$ and $c_i = c'_i$ for $i \neq s$. If ID^* is the actual signer, we can solve the DL problem as follow: $x = \frac{z-z'}{c'_s-c_s}$.

Probability. Let $q_{H_i} (i = 1, 2)$, q_E , q_R , q_U and q_S be the numbers of $H_i (i = 1, 2)$ queries, secret value queries, user public replacement requests, user public key queries and signing queries, respectively.

For simplified, it is assumed that $L_E \cap L_R = \phi$. The probability that ID^* 's secret value was not queried and ID^* 's user public key was not replaced by \mathcal{A}_2 is $\frac{q_U - q_E - q_R}{q_U}$. The probability that ID^* belongs to the group W is $\frac{n}{q_U - q_E - q_R}$. The probability that ID^* is the actual signer is $\frac{1}{n}$. So the combined probability is $\frac{q_U - q_E - q_R}{q_U} \cdot \frac{n}{q_U - q_E - q_R} \cdot \frac{1}{n} = \frac{1}{q_U}$.

Therefore, if the adversary \mathcal{A}_2 can win the EUF-CLRS Game II with advantage ε and within time T , then \mathcal{C} can solve the DL problem with the probability $\frac{\varepsilon^2}{66C^n q_{H_3}} \frac{1}{q_U}$

within time $2T + (q_{H_1} + qu + (2n + 2)q_S)T_e$ (where T_e denotes the time for a exponentiation in G).

Theorem 4.3. *The scheme is anonymous.*

Proof. The adversary \mathcal{A} runs the setup program with the parameter k to generate a master secret key $msk = (p, q, e)$ and the public system parameters $params = \{N, d, G, P, H_1, H_2, H_3\}$, then gives $params$ and msk to the challenger \mathcal{C} . \square

First of all, \mathcal{A} makes a polynomially bounded number of queries as those in the proof of Theorem 4.1.

Secondly, \mathcal{A} outputs a group of n users W , two different members $ID_0, ID_1 \in W$ and a messages M . \mathcal{C} randomly chooses a bit $\mu \in \{0, 1\}$ and provide \mathcal{A} with $\sigma = \text{Sign}(M, W, D_\mu, t_\mu)$.

Once again, \mathcal{A} makes a polynomially bounded number of queries as those in the proof of Theorem 4.1.

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

For a signature (σ, M, W) generated by signing algorithm, if $ID_i \in W$ is not the actual signer, then c_i, A_i are chosen independently and distributed uniformly over Z_d^* and Z_N^* , respectively. If ID_s is the actual signer, since r_1, A are chosen uniformly at random from Z_d^* and Z_N^* , respectively, then $A_s = A^d Q_s^{r_1} \prod_{i=1, i \neq s}^n (A_i^{-1} Q_i^{-h_i})$ is distributed uniformly. By r_2 is chosen uniformly at random from Z_d^* and c is the output of the random oracle, then $c_s = c - \sum_{i=1, i \neq s}^n c_i \pmod{d}$ is distributed uniformly. By h_s is the output of the random oracle, then h_s is distributed uniformly. Moreover, z and V are also distributed uniformly over Z_d^* and Z_N^* , respectively. It is easy to see that all the mentioned parameters are uniformly distributed. Therefore, $\Pr[\mu' = \mu] = \frac{1}{2}$. In other words, the advantage of \mathcal{A} in the Game III is negligible.

5. EFFICIENCY

In this section, we compare the performance of our scheme with several ring signature schemes, we define some notations 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 .

Through a PIV 3 GHZ processor with 512 M bytes memory and the Windows XP operating system, Cao *et al.* [6] obtained the running time for cryptographic operations. To achieve the 1024-bit RSA level security, they used a supersingular elliptic curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2. 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, they used the ECC group on Koblitz elliptic curve $y^2 = x^3 + ax^2 + b$, which is defined on $F_{2^{163}}$ with $a = 1$ and b is a 163-bit random prime. The running times are listed in Table 1.

TABLE 1. Cryptographic operation time (in milliseconds).

P	M_P	M_E	M_N
20.01	6.38	0.83	11.20

TABLE 2. Comparison of several ring signature schemes.

Scheme	Sign	Verify	Execution time/($n = 10$)
Herranz 1 [8]	$2nM_N$	$(n + 1)M_N$	$33.60n + 11.20/347.20$
Herranz 2 [8]	$2nM_N$	$2nM_N$	$44.80n/448.00$
Chow [4]	$P + 2nM_P$	$nP + (n + 1)M_P$	$39.15n + 26.39/417.89$
Zhang [16]	$2P + (3n + 3)M_P$	$3P + 2nM_P$	$31.90n + 119.19/438.19$
Chang [3]	$2P + (2n + 2)M_P$	$2P + (2n + 1)M_P$	$25.52n + 99.18/354.38$
Our scheme	$nM_E + (n + 2)M_N$	$(n + 1)M_E + (n + 1)M_N$	$24.06n + 34.43/275.03$

To evaluate the computation efficiency of different schemes, we use the simple method from [6]. For example, Chow *et al.*'s [4] scheme requires $3n + 1$ pairing-based scalar multiplication operations and $n + 1$ pairing operations. So the resulting computation time is $6.38 \times (3n + 1) + 20.01 \times (n + 1) = 39.15n + 26.39$. In order to facilitate the comparison, we let $n = 10$, then the computation time is $39.15 \times 10 + 26.39 = 417.89$. Based on the above parameter settings and ways, the detailed comparison results of several different ring signature schemes are illustrated in Table 2.

6. CONCLUSION

It is well known that RSA is a classic cryptographical system and it is widely used in many industrial application. However, all of the existing CLRS schemes are based on bilinear pairings. But the computation cost of the pairings is much higher than that of the exponentiation in a RSA group. So it is quite significant to construct CLRS from RSA. In this paper, we proposed a new CLRS based on RSA problem and DL problem and proved the security in the random oracle model. Our scheme needs not pairings and it is more efficient than previous ones. To the best of our knowledge, our scheme is the first CLRS scheme without pairings. Due to the good properties of our scheme, it should be useful for practical applications.

Acknowledgements. The author is grateful to Jiwen Zeng, Chunming Tang and Huawei Huang for their helpful suggestions. This research is supported by the National Natural Science Foundation of China under Grants Nos. 61562012, 11261060, 11271003, 61462016, the Natural Science Foundation of Guangdong Province to Develop Major Infrastructure

Projects under Grant No. 2015A030308016, the Basic Research Major Projects of Department of Education of Guangdong Province under Grant No. 2014KZDXM044, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20134410110003, and Construction project of innovation team in universities in Guangdong Province under Grant No. 2015KCXTD014.

To revise the manuscript and respond to the reviewers' comments, three teachers Jiwen Zeng, Chunming Tang and Huawei Huang, give me lots of help and suggestions.

REFERENCES

- [1] S.S. Al-Riyami and K.G. Paterson, Certificateless public cryptography. In: Advances in Cryptology-Asiacrypt, edited by C.S. Laih. Vol. 2894 of *Lect. Notes Comput. Sci.* (2003) 452–473.
- [2] A. Bender, J. Katz and R. Morselli, Ring signatures: stronger definitions, and constructions without random Oracles. *J. Cryptology* **22** (2009) 114–138.
- [3] S. Chang, D.S. Wong, Y. Mu and Z.F. Zhang, Certificateless threshold ring signature. *Inf. Sci.* **179** (2009) 3685–3696.
- [4] S.S.M. Chow and W.S. Yap, Certificateless ring signature. *Cryptology ePrintArchive: Report 2007/236*. Available at <http://eprint.iacr.org/2007/236>
- [5] S.S.M. Chow, S.M. Yiu and L.C.K. Hui, Efficient identity based ring signature. ACNS'05, Vol. 3531 of *Lect. Notes Comput. Sci.* 499–512.
- [6] X. Cao and W. Kou, A pairing-free identity-based authenticated key agreement scheme with minimal message exchanges. *Inf. Sci.* **180** (2010) 2895–2903.
- [7] X.D. Dong and H.M. Lu, An Improved RSA-Based Certificateless Signature Scheme. *Appl. Mech. Mater.* **687** (2014) 2165–2168.
- [8] J. Herranz, Identity-based ring signatures from RSA. *Theoret. Comput. Sci.* **389** (2007) 100–117.
- [9] J. Herranz and G. Saez, New identity-based ring signature schemes. *ICICS*. Vol. 3269 of *Lect. Notes Comput. Sci.* (2004) 27–39.
- [10] L. Nguyen, Accumulators from bilinear pairings and applications. *CT-RSA* (2005) 275–292.
- [11] R.L. Rivest, A. Shamir and Y. Tauman, How to leak a secret. *ASIACRYPT'01*. Vol. 2248 of *Lect. Notes Comput. Sci.* (2001) 552–565.
- [12] A. Shamir, Identity-based cryptosystem and signature scheme. In: Advances in Cryptology-Crypto. Vol. 196 of *Lect. Notes Comput. Sci.* (1984) 47–53.
- [13] H.Q. Wang, Certificateless ring signature scheme from anonymous subsets. *International Conference on Multimedia Information Networking and Security* (2010) 413–417.
- [14] T.H. Yuen, J.K. Liu, M.H. Au, W. Susilo and J. Zhou, Efficient linkable and/or threshold ring signature without random oracles. *Comput. J.* **56** (2013) 407–421.
- [15] J. Zhang and J. Mao, An efficient RSA-based certificateless signature scheme. *J. Systems Software* **85** (2012) 638–642.
- [16] L. Zhang, F. Zhuang and W. Wu, A provably secure ring signature scheme in certificateless cryptography. In: *Proc. of the Prousec*. Vol. 4784 of *Lect. Notes Comput. Sci.* (2007) 103–121.

Communicated by D. Augot.

Received October 31, 2015. Accepted May 9, 2016.