IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 29, 2020, accepted March 7, 2020, date of publication March 17, 2020, date of current version March 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981360

Certificateless Linkable Ring Signature Scheme

LUNZHI DENG, HONGYU SHI, AND YAN GAO
School of Mathematical Sciences, Guizhou Normal University, Guiyang 550001, China

Corresponding author: Lunzhi Deng (denglunzhi@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61562012 and Grant 61962011, in part by the Innovation Group Major Research Projects of Department of Education of Guizhou Province under Grant KY[2016]026, and in part by the Guizhou Provincial Science and Technology Foundation under Grant [2019]1434.

ABSTRACT Ring signature is an anonymous signature that both authenticates the message and protects the identity information of the signer. It is suitable for scenarios such as anonymous network access, online auctions, etc. However, there is a problem in a regular ring signature scheme. The signer can generate multiple different signatures for the same message without being discovered by the verifier. This will of course cause some confusion. Linkable ring signature (LRS) resolves the problem. The verifier can determine if multiple signatures were generated by the same signer, but he cannot determine the actual signer's identity. In this paper, we first proposed the system model and security requirements for certificateless linked ring signature (CL-LRS). Then, we constructed a concrete CL-LRS scheme and gave the security proofs. Finally, we compared the efficiency of our scheme with several other LRS schemes. Our scheme does not use pairing operation and is more suitable for the computation-constrained environment.

INDEX TERMS Anonymous, certificateless cryptography, elliptic curve group, linkable, ring signature.

I. INTRODUCTION

With the continuous upgrade of network information technology and communication technology, e-commerce has entered people's lives, and people have become accustomed to conducting various business activities (online shopping, online transactions, electronic payment, etc.) through the network. According to the Blue Book of the Global E-commerce Industry in 2018, from the perspective of user scale, in 2018, the size of global online shopping consumers is expected to exceed 1.88 billion, a year-on-year increase of 8.9%, and the global online shopping penetration rate is about 25.3%. In the future, as the Internet penetration rate rises year by year, the growth rate of online shopping consumers will gradually slow down. It is estimated that the global online shopping consumer size will reach 2.38 billion by 2022, and the online shopping penetration rate will increase to more than 30%.

The rapid development of e-commerce is inseparable from the construction of infrastructure such as Internet software and hardware. In 2018, China's fixed data and Internet business revenues reached 207.2 billion yuan, an increase of 5.1% over the previous year, and its proportion in telecommunications business revenue increased from 15.6% to 15.9% in the previous year. An increase of 10.2% over the previous year, and its proportion in telecommunications business income increased from 43.5% in the previous year to 46.6%. According to statistics, as of the end of 2018, the total number of mobile Internet users in China reached 1.39 billion, an increase of 10.7% year-on-year, and 1.26 billion users used mobile phones to access the Internet.

As people do more business online, there have been many incidents that leaked personal privacy. Privacy protection has become an urgent issue. To protect the identity of the signer, Rivest *et al.* [21] put forward the concept of ring signature, as shown in Figure 1. The signer forms a group by selecting several users, including himself, and then yields a signature. It can convince the verifier that the signatue was generated by someone in the group. However, it is impossible to determine the signer's identity.

In ''regular'' ring signature schemes, the signer can generate multiple different signatures for the same message. However, it is impossible for the verifier to determine if the two signatures are yield by the same person. This could lead to abuse of signing rights by users. To resolve the problem, Liu *et al.* [17] introduced the notion of linkable ring signature (LRS), the signer in a signature is still anonymous and the verifier can determine whether the real signers in two signatures are same.

LRS is suitable for many real-world scenarios, such as online authentication, e-voting and e-cash. For an example,

The associate editor coordinating the review of this manuscript and approving it for publication was Parul Garg.

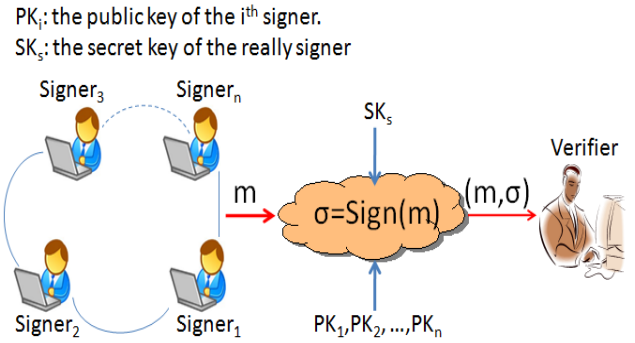

FIGURE 1. Ring signature.

a "regular" ring signature scheme is not suitable for e-voting since the same voter can yield multi different votes without being found. It is impossible for the checker to discern whether the two votes are yield by the same voter. LRS is a good option to solve the problem due to the checker can detect whether two votes are yielded by the same voter.

To avoid key escrow while eliminating certificate management, Al-Riyami and Paterson [3] introduced certificateless public key cryptography(CL-PKC). A user's private key consists of two parts: a partial private key and a secret value. They are independent of each other. In 2018, Cheng and Chen [7] redefined the formulations of CL-PKC. In their construction, the user first picks a secret value, computes the partial public key and sends it to KGC. Follow on, KGC yields the partial private key with the partial public key, identity information and master secret key, then sends it to the user. So the partial private key and the secret value are are no longer independent of each other.

A. RELATED WORK

In 2007, Chow and Yap [9] presented the security model of certificateless ring signature (CL-RS) and gave a concrete construction. Zhang *et al.* [30] put forward a CL-RS scheme, which requires a constant number pairing operations. In 2009, Chang *et al.* [6] redefined the security requirements, which captures the user partial key replacement attack, then presented a concrete CL-RS scheme. In 2010, Wang and Han [28] proposed a CL-RS scheme from anonymous subsets. In 2015, Deng [10] constructed a CL-RS scheme, which does not use pairing operation. In 2017, Zhang *et al.* [31] presented a new CL-RS scheme, which requires a constant number pairing operations.

In 2004, Liu *et al.* [17] put forward the first LRS scheme. In which, the signer is still anonymous, the verifier can determine whether the real signers in two signatures are same. After that, several practical LRS schemes [18], [25], [32] have been proposed. In these schemes, the length of the signature increases linearly with the number of members in the ring. In 2005, Tsang and Wei [26] proposed a LRS scheme, in which the length of the signature is constant. Fujisaki [12] and Yuen *et al.* [29] respectively proposed a LRS scheme in the standard model. In which, the length of signature increases linearly with $\sqrt{n}$, where n is the number of members in the ring. Jeong *et al.* [15] and Liu *et al.* [19] respectively designed a LRS scheme, in which the identity of the real signer is unconditionally anonymous. In 2018, Boyen and Haines [5] put forward a LRS scheme, which uses an n-time one-way private key update mechanism based on n-times multi-linear mapping. Baum *et al.* [4] proposed a LRS scheme, the security is based on difficult questions on lattices.

In 2006, Au *et al.* [1] and Chow *et al.* [8] respectively proposed an identity-based LRS (IB-LRS) scheme, in which the length of the signature is constant. However, Jeong *et al.* [16] indicated that the scheme [1] is vulnerable. In 2010, Tsang *et al.* [27] presented an IB-LRS scheme, which does not require pairing operation. In 2013, Au *et al.* [2] constructed an IB-LRS scheme, in which the length of the signature is constant. In these two schemes, the user's private key does not only be decided by the identity information of the user and master secret key, one user may correspond to more than one private keys, It does not meet the requirements of traditional identity-based cryptography. In 2019, Deng *et al.* [11] put forward a new IB-LRS scheme, which requires only seven pairing operations.

LRS achieves both anonymity and traceability, and is increasingly being used in cryptocurrencies. In 2014, a cryptocurrency called Monero was created that provides anonymity to users by using Ring Confidential (Ring CT) [20], which is virtually a variant of LRS. In 2017, Sun *et al.* [23] presented a new Ring CT scheme, which is based on the Pedersen commitment. In 2018, Torres *et al.* [24] proposed a lattice-based Ring CT scheme, which provides confidential transactions for cryptocurrencies.

B. MOTIVATION AND CONTRIBUTIONS

LRS implements authentication of messages, hides the identity of the signer, and prevents users from abusing signing right. It is suitable for scenarios such as online authentication, electronic voting and electronic cash. The linkable ring signature schemes currently known are either from traditional public key infrastructures or identity-based cryptography. Neither certificate management nor key escrow required in CL-PKC. It is meaningful to constructed a secure and efficient CL-LRS scheme. Contributions to this paper consist of the following four sections.

- We proposed the system model and the security requirements for a CL-LRS scheme: unforgeability, anonymity, linkability, unlinkability, non-slanderability,.
- We proposed a concrete CL-LRS scheme by following the methods in [7]. It avoids key escrow while not requiring certificate management. To the best of our knowledge, it is the first CL-LRS scheme.
- We gave the security proofs in random oracle model (ROM). Our scheme has all the security attributes.
- We made a performance comparison between the our scheme and several other LRS schemes. Our scheme is more efficient than the other LRS schemes since no pairing operation is required.

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 $q = \|\mathcal{G}\|/2$.
Z_q^*	A set consisting of positive integers less than q.
G	A additive group consisting of point on $\mathbb{E}$.
P	A generator of G with order q.
x	The master secret key of system, where $x \in Z_q^*$.
P_{pub}	The public key of system, where $P_{pub} = xP$.
$H_1 \sim H_4$	Four secure hash functions.
ID_i	The identity of i^{th} user.
d_i	The partial private key of i^{th} user.
t_i	The secret value of i^{th} user, where $T_i = t_iP$.
PK_i	The public key of i^{th} user, where $PK_i = (T_i, R_i)$ and $R_i = r_iP$.
W	A set consisting of n users, where $W = \{ID_1, \ldots, ID_n\}$.
U	A set consisting of the identities/public keys, where $U = W \bigcup \{PK_i : ID_i \in W\}$.
$event$	An event description.
m	A message.
σ	A certificateless linkable ring signature.

C. ROADMAP

The structure of this article is as follows. First, we introduced two mathematical tools in Section II. Second, we proposed the system model and security requirements in Section III and Section IV, respectively. Third, we proposed a concrete CL-LRS scheme in Section V and showed the security proofs in Section VI. Next, we made the performance comparisons for several schemes in Section VII. Lastly, we made some conclusions in Section VIII.

II. PRELIMINARIES

In this section, we introduce two mathematical tools. The notations used throughout the paper are listed in Table 1.

Elliptic Curve Group:

Let $\mathbb{E}/F_p$ denote an elliptic curve $\mathbb{E}$ over a prime finite field F_p, defined by an equation:

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

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

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

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

III. SYSTEM MODEL

A CL-LRS scheme consists of the following eight algorithms:

- Setup: Input a security parameter ν, key generation center (KGC) yields the system parameters (*params*) and the master secret key (*msk*).
- Set-SV: The user $ID_i \in \{0, 1\}^*$ randomly picks a value t_i.
- Generate-TPK: The user ID_i yields a partial public key T_i.
- Extract-PPK: Input a tuple (ID, T_i), KGC yields a partial private key d_i.
- Set-UPK: The user ID_i outputs his user public key PK_i.
- Sign: Input a tuple ($event$, m, U), the real signer ID_s yields a signature σ.
- Verify: Input a tuple (σ, $event$, m, U), the verifier outputs 1 if σ is valid. Otherwise, outputs 0.
- Link: Input two tuples ($event$, σ_1, m_1), ($event$, σ_2, m_2), anyone outputs either *link* or *unlink*.

IV. NOTIONS OF SECURITY

An adversary $\mathscr{A}$ ($\mathscr{A}_1$, $\mathscr{A}_2$) performs the following queries.

- Query-Hash: $\mathscr{A}$ inputs a value and obtains a corresponding value of the hash function.
- Query-UPK: $\mathscr{A}$ submits an identity ID_i and obtains a value PK_i.
- Replace-TPK: $\mathscr{A}$ submits a tuple (T_i', ID_i), the challenger $\mathscr{C}$ replaces T_i with T_i'.
- Query-SV: $\mathscr{A}$ submits an identity ID_i whose TPK was not replaced and obtains a value t_i.
- Query-PPK: $\mathscr{A}$ submits an identity ID_i and obtains a value d_i.
- Query-Sign: $\mathscr{A}$ submits a tuple ($event$, m, U, ID_s) and obtains a signature.

Definition 2: A CL-LRS scheme is unforgeable (UNF-CL-LRS) if the advantage of each adversary is negligible in the next games.

Game I: A Type I adversary $\mathscr{A}_1$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization: $\mathscr{C}$ yields the *params* and *msk* by running the Setup algorithm, then forwards *params* to the adversary $\mathscr{A}_1$.

Query: $\mathscr{A}_1$ performs various queries as defined above.

Forge: $\mathscr{A}_1$ yields a new tuple (σ, $event$, m, U). $\mathscr{A}_1$ wins if the following requirements are met.

1) The tuple (σ, $event$, m, U) is not obtained by performing Query-Sign.
2) $\mathscr{A}_1$ did not perform Query-PPK for anyone in W.
3) Verify(σ, $event$, m, U) = 1.

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

$$Adv_{\mathscr{A}_1}^{UNF-CL-LRS} = \Pr[\mathscr{A}_1 \ wins].$$

Game II: A Type II adversary $\mathscr{A}_2$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization: $\mathscr{C}$ yields the *params* and *msk* by running the Setup algorithm, then forwards them to the adversary $\mathscr{A}_2$.

Query: Same as those in Game I.

Forge: $\mathscr{A}_2$ yields a new tuple (σ, $event$, m, U). $\mathscr{A}_2$ wins if the following requirements are met.

1) The tuple (σ, $event$, m, U) is not obtained by performing Query-Sign.
2) $\mathscr{A}_2$ did not perform Query-SV for anyone in W.
3) $\mathscr{A}_2$ did not perform Replace-TPK for anyone in W.
4) Verify (σ, $event$, m, U) = 1

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

$$Adv_{\mathscr{A}_2}^{UNF-CL-LRS} = \Pr[\mathscr{A}_2 \ wins].$$

Definition 3: A CL-LRS scheme is anonymous (ANO-CL-LRS) if the advantage of each adversary is negligible in the next games.

Game III: A Type I adversary $\mathscr{A}_1$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization: Same as that in Game I.

Phase 1: $\mathscr{A}_1$ performs various queries.

Challenge: $\mathscr{A}_1$ inputs a tuple $(event, m, U, ID_0, ID_1)$, $\mathscr{C}$ selects at random a bit $\mu \in \{0, 1\}$, then provides $\mathscr{A}_1$ with $\sigma = \text{Sign}(event, m, U, d_\mu, t_\mu)$. And the following conditions must be met.

1) $ID_0, ID_1 \in W$.
2) $\mathscr{A}_1$ did not perform Query-PPK for ID_0 and ID_1.
3) $\mathscr{A}_1$ did not perform Query-Sign for $(event, *, *, ID_i)$ for $i = 0, 1$.

Phase 2: $\mathscr{A}_1$ performs various queries and follows the constraints below.

1) $\mathscr{A}_1$ cannot perform Query-PPK for ID_0 and ID_1.
2) $\mathscr{A}_1$ cannot perform Query-Sign for $(event, *, *, ID_i)$ for $i = 0, 1$.

Response: $\mathscr{A}_1$ outputs a bit $\mu' \in \{0, 1\}$. $\mathscr{A}_1$ wins if $\mu' = \mu$. The advantage of $\mathscr{A}_1$ is defined as:

$$Adv_{\mathscr{A}_1}^{ANO-CL-LRS} = |2\Pr[\mu' = \mu] - 1|.$$

Game IV: A Type II adversary $\mathscr{A}_2$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization: Same as that in Game II.

Phase 1: $\mathscr{A}_2$ performs various queries.

Challenge: $\mathscr{A}_2$ inputs a tuple $(event, m, U, ID_0, ID_1)$, $\mathscr{C}$ selects at random a bit $\mu \in \{0, 1\}$, then provides $\mathscr{A}_2$ with $\sigma = \text{Sign}(event, m, U, d_\mu, t_\mu)$. And the following conditions must be met.

1) $ID_0, ID_1 \in W$.
2) $\mathscr{A}_2$ did not perform Query-SV for ID_0 and ID_1.
3) $\mathscr{A}_2$ did not perform Replace-TPK for ID_0 and ID_1.
4) $\mathscr{A}_2$ did not perform Query-Sign for $(event, *, *, ID_i)$ for $i = 0, 1$.

Phase 2: $\mathscr{A}_2$ performs various queries and follows the constraints below.

1) $\mathscr{A}_2$ cannot perform Query-SV for ID_0 and ID_1.
2) $\mathscr{A}_2$ cannot perform Replace-TPK for ID_0 and ID_1.
3) $\mathscr{A}_2$ cannot perform Query-Sign for $(event, *, *, ID_i)$ for $i = 0, 1$.

Response: $\mathscr{A}_2$ outputs a bit $\mu' \in \{0, 1\}$. $\mathscr{A}_2$ wins if $\mu' = \mu$. The advantage of $\mathscr{A}_2$ is defined as:

$$Adv_{\mathscr{A}_2}^{ANO-CL-LRS} = |2\Pr[\mu' = \mu] - 1|.$$

Definition 4: A CL-LRS scheme is linkable (LINK-CL-LRS) if the advantage of each adversary is negligible in the next games.

Game V: A Type I adversary $\mathscr{A}_1$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization, Query: Same as those in Game I.

Unlink: $\mathscr{A}_1$ generates two tuples $(\sigma_1, event, m_1, U_1)$ and $(\sigma_2, event, m_2, U_2)$, where $U_k = W_k \bigcup \{PK_{ki} : ID_{ki} \in W_k\}$, $W_k = \{ID_{k1}, \ldots, ID_{kn}\}$ for $k = 1, 2$. $\mathscr{A}_1$ wins if the following requirements are met.

1) The two tuples are not obtained by performing Query-Sign.
2) $\mathscr{A}_1$ performed Query-PPK on at most one user in $W_1 \bigcup W_2$.
3) $\text{Verify}(\sigma_1, event, m_1, U_1) = 1$
4) $\text{Verify}(\sigma_2, event, m_2, U_2) = 1$.
5) $\text{Link}(\sigma_1, \sigma_2) = unlink$.

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

$$Adv_{\mathscr{A}_1}^{LINK-CL-LRS} = \Pr[\mathscr{A}_1 \ wins].$$

Game VI: A Type II adversary $\mathscr{A}_2$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization, Query: Same as those in Game II.

Unlink: $\mathscr{A}_2$ generates two tuples $(\sigma_1, event, m_1, U_1)$ and $(\sigma_2, event, m_2, U_2)$, where $U_k = W_k \bigcup \{PK_{ki} : ID_{ki} \in W_k\}$, $W_k = \{ID_{k1}, \ldots, ID_{kn}\}$ for $k = 1, 2$. $\mathscr{A}_2$ wins if the following requirements are met.

1) The two tuples are not obtained by performing Query-Sign.
2) $\mathscr{A}_2$ performed Query-SV (or Replace-TPK) on at most one user in $W_1 \bigcup W_2$.
3) $\text{Verify}(\sigma_1, event, m_1, U_1) = 1$
4) $\text{Verify}(\sigma_2, event, m_2, U_2) = 1$.
5) $\text{Link}(\sigma_1, \sigma_2) = unlink$.

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

$$Adv_{\mathscr{A}_2}^{LINK-CL-LRS} = \Pr[\mathscr{A}_2 \ wins].$$

Definition 5: A CL-LRS scheme is non-slanderable (NS-CL-LRS) if the advantage of each adversary is negligible in the next games.

Game VII: A Type I adversary $\mathscr{A}_1$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization: Same as that in Game I.

Phase 1: $\mathscr{A}_1$ performs various queries.

Challenge: $\mathscr{A}_1$ inputs a tuple $(event, m, U, ID_s)$, $\mathscr{C}$ yields a valid signature $\sigma = \text{Sign}(event, m, U, d_s, t_s)$ and sends it to $\mathscr{A}_1$.

Phase 2: $\mathscr{A}_1$ performs various queries.

Slander: $\mathscr{A}_1$ gives a tuple $(\sigma', event, m', U')$, where $U' = W' \bigcup \{PK_i : ID_i \in W'\}$. $\mathscr{A}_1$ wins if the following requirements are met.

1) The tuple is not obtained by performing Query-Sign.
2) $\mathscr{A}_1$ did not perform Query-PPK for the actual signer $ID_s \in W$.
3) $\text{Verify}(\sigma', event, m', U') = 1$.
4) $\text{Link}(\sigma, \sigma') = link$.

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

$$Adv_{\mathscr{A}_1}^{NS-CL-LRS} = \Pr[\mathscr{A}_1 \ wins].$$

Game VIII: A Type II adversary $\mathscr{A}_2$ and a challenger $\mathscr{C}$ paly the game as follows.

Initialization: Same as that in Game II.

Phase 1: $\mathscr{A}_2$ performs various queries.

Challenge: $\mathscr{A}_2$ inputs a tuple $(event, m, U, ID_s)$, $\mathscr{C}$ yields a valid signature $\sigma = \text{Sign}(event, m, U, d_s, t_s)$ and sends it to $\mathscr{A}_2$.

Phase 2: $\mathscr{A}_2$ performs various queries.

slander: $\mathscr{A}_2$ gives a tuple $(\sigma', event, m', U')$, where $U' = W' \bigcup \{PK_i : ID_i \in W'\}$. $\mathscr{A}_2$ wins if the following requirements are met.

1) The tuple is not obtained by performing Query-Sign.
2) $\mathscr{A}_2$ did not perform Query-SV for the actual signer $ID_s \in W$.
3) $\mathscr{A}_2$ did not perform Replace-TPK for the actual signer $ID_s \in W$.
4) $\text{Verify}(\sigma', event, m', U') = 1$.
5) $\text{Link}(\sigma, \sigma') = link$.

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

$$Adv_{\mathscr{A}_2}^{NS-CL-LRS} = \Pr[\mathscr{A}_2 \ wins].$$

V. NEW SCHEME

In this section, we propose a concrete CL-LRS scheme as follows.

- Setup: Inputs a security parameter v, KGC performs the following steps.
  1) Chooses a group G with prime order $q > 2^v$ and a generator P of G.
  2) Picks four secure hash functions $H_1, H_3, H_4 : \{0,1\}^* \to Z_q^*$, $H_2 : \{0,1\}^* \to G$.
  3) Picks a value $x \in Z_q^*$, sets $msk = \{x\}$ and computes $P_{pub} = xP$.
  4) Broadcasts $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$.
- Set-SV: The user ID_i randomly chooses $t_i \in Z_q^*$.
- Generate-TPK: The user ID_i computes $T_i = t_i P$.
- Extract-PPK: Inputs a tuple (ID_i, T_i), KGC randomly chooses $r_i \in Z_q^*$ and computes $R_i = r_i P$, $k_i = H_1(T_i, R_i, ID_i)$, $d_i = r_i + k_i x$, then sends $D_i = (R_i, d_i)$ to the user ID_i by an authenticated channel.
- Set-UPK: The user ID_i sets $PK_i = (T_i, R_i)$.
- Sign: Inputs a tuple $(event, m, U)$, the real signer $ID_s \in W$ does as follows.
  1) Computes $E = H_2(event)$, $h = H_3(event)$, $V = (d_s + ht_s)E$.
  2) Randomly chooses $z, c_i \in Z_q^*$ for $i = 1, 2, \ldots, s-1, s+1, \ldots, n$.
  3) Computes $A = zE + \sum_{i=1, i\neq s}^{n} c_i V$.
  4) Computes $k_i = H_1(T_i, R_i, ID_i)$ for $i = 1, 2, s-1, s+1, \ldots, n$.
  5) Computes $B = zP + \sum_{i=1, i\neq s}^{n} c_i(hT_i + R_i + k_i P_{pub})$.
  6) Computes $u = H_4(event, m, V, A, B, U)$.
  7) Computes $c_s = u - \sum_{i=1, i\neq s}^{n} c_i$, $y = z - c_s(d_s + ht_s)$
  8) Outputs the signature $\sigma = (c_1, \ldots, c_n, y, V)$.
- Verify: Inputs a tuple $(\sigma = (c_1, \ldots, c_n, y, V), event, m, U)$, the verifier does as follows.
  1) Computes $E = H_2(event)$, $h = H_3(event)$.
  2) Computes $A = yE + \sum_{i=1}^{n} c_i V$.
  3) Computes $k_i = H_1(T_i, R_i, ID_i)$ for $i = 1, 2, \ldots, n$.
  4) Computes $B = yP + \sum_{i=1}^{n} c_i(hT_i + R_i + k_i P_{pub})$.
  5) Computes $u = H_4(event, m, V, A, B, U)$.
  6) Checks whether $\sum_{i=1}^{n} c_i (\text{mod } q) = u$. Outputs 1 if the equality holds and outputs 0 otherwise.
- Link: Inputs two message-signature pairs $(event, m_1, \sigma_1 = (V_1, \cdot))$, $(event, m_2, \sigma_2 = (V_2, \cdot))$. The verifier first checks if two signatures are valid, and refuses to answer if one signature is invalid. Next, the verifier outputs *link* if $V_1 = V_2$ and outputs *unlink* otherwise.
- On correctness

$$\begin{aligned} &yE + \sum_{i=1}^{n} c_i V \\ &= (z - c_s(d_s + ht_s))E + \sum_{i=1}^{n} c_i V \\ &= zE + \sum_{i=1, i\neq s}^{n} c_i V \\ &= A \\ &yP + \sum_{i=1}^{n} c_i(hT_i + R_i + k_i P_{pub}) \\ &= (z - c_s(ht_s + d_s))P + \sum_{i=1}^{n} c_i(hT_i + R_i + k_i P_{pub}) \\ &= zP + \sum_{i=1, i\neq s}^{n} c_i(hT_i + R_i + k_i P_{pub}) \\ &= B \end{aligned}$$

VI. SECURITY OF SCHEME

In this section, we give the security proofs of the new scheme in ROM.

Theorem 1: If DL problem is hard, then the scheme is unforgeable against Type I adversary in ROM.

Proof: Suppose that the tuple (P, aP) is an instance of DL problem. $\mathscr{C}$ will compute the value a by acting as the challenger in Game I.

Initialization: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ by running the Setup algorithm, then forwards it to the $\mathscr{A}_1$.

Queries: Several lists are set to store the queries and answers. $\mathscr{A}_1$ will execute Query-UPK for an identity ID_i before that is used in any other queries.

- Query-UPK: $\mathscr{C}$ maintains a list L_U of tuple (ID_i, t_i, r_i). $\mathscr{A}_1$ inputs an identity ID_i, $\mathscr{C}$ does as follows: At the j^{th} query, selects at random $t_j \in Z_q^*$, sets $ID_j = ID^{\diamond}$ and $PK_j = PK^{\diamond} = (t_j P, aP)$. For $i \neq j$, $\mathscr{C}$ selects at random $t_i, r_i \in Z_q^*$, returns $PK_i = (t_i P, r_i P)$, and stores the query and answer in the list L_U.
- Query-H_1: $\mathscr{C}$ maintains a list L_1 of tuple (α_i, k_i). When $\mathscr{A}_1$ issues a query $H_1(\alpha_i)$, $\mathscr{C}$ randomly picks $k_i \in Z_q^*$, sets $H_1(\alpha_i) = k_i$ and adds (α_i, k_i) to the list L_1.

- Query-H_2: $\mathscr{C}$ maintains a list L_2 of tuple (β_i, E_i). When $\mathscr{A}_1$ issues a query $H_2(\beta_i)$, $\mathscr{C}$ randomly picks $E_i \in G_1$, sets $H_2(\beta_i) = E_i$ and adds (β_i, E_i) to the list L_2.
- Query-H_3: $\mathscr{C}$ maintains a list L_3 of tuple (γ_i, h_i). When $\mathscr{A}_1$ issues a query $H_3(\gamma_i)$, $\mathscr{C}$ randomly picks $h_i \in Z_q^*$, sets $H_3(\gamma_i) = h_i$ and adds (γ_i, h_i) to list L_3.
- Query-H_4: $\mathscr{C}$ maintains a list L_4 of tuple (δ_i, u_i). When $\mathscr{A}_1$ issues a query $H_4(\delta_i)$, $\mathscr{C}$ randomly picks $u_i \in Z_q^*$, sets $H_4(\delta_i) = u_i$ and adds (δ_i, u_i) to the list L_4.
- Replace-TPK: $\mathscr{C}$ maintains a list L_R of tuple (ID_i, T_i, T_i'). When $\mathscr{A}_1$ issues this request for an identity ID_i with a new T_i'. $\mathscr{C}$ replaces T_i with T_i' and adds (ID_i, T_i, T_i') to the list L_R.
- Query-SV: When $\mathscr{A}_1$ issues this query for an identity ID_i, $\mathscr{C}$ finds (ID_i, t_i, r_i) in the list L_U, responds with t_i and adds (ID_i, t_i) to the list L_E.
- Query-PPK: $\mathscr{C}$ maintains a list L_D of tuple (ID_i, d_i). When $\mathscr{A}_1$ issues this query for an identity ID_i. $\mathscr{C}$ fails if $ID_i = ID^\diamond$. Otherwise, $\mathscr{C}$ finds (ID_i, t_i, r_i) in the list L_U or (ID_i, T_i, T_i') in list L_R, gives the d_i by executing the Extract-PPK algorithm and adds (ID_i, d_i) to the list L_D.
- Query-Sign: $\mathscr{A}_1$ inputs a tuple $(event, m, U, ID_s)$, where $ID_s \in W$. $\mathscr{C}$ performs the following steps.
  If $ID_s \neq ID^\diamond$ and $ID_s \notin L_R$, $\mathscr{C}$ outputs a signature σ by executing the Sign algorithm. Otherwise, $\mathscr{C}$ maintains a list $L^\diamond$ of tuple $(ID_s, event, V)$ and does as follows.

  1) Retrievals the list $L^\diamond$, then uses the value V if there is a tuple $(ID_s, event, V)$ in it. Otherwise, chooses a new value $V \in G_1$ and adds $(ID_s, event, V)$ to the list $L^\diamond$.
  2) Computes $E = H_2(event)$, $h = H_3(event)$.
  3) Computes $k_i = H_1(R_i, ID_i)$ for $i = 1, 2, \ldots, n$.
  4) Selects at random $y, c_i \in Z_q^*$ for $i = 1, 2, \ldots, n$.
  5) Computes $A = yE + \sum_{i=1}^{n} c_i V$, $B = yP + \sum_{i=1}^{n} c_i(hT_i + R_i + k_i P_{pub})$.
  6) Computes $u = \sum_{i=1}^{n} c_i (\mathrm{mod}\ q)$.
  7) Adds $u = H_4(event, m, V, A, B, U)$ to the list L_4. Repeats the steps 4-7 if collision occurs.
  8) Outputs the signature $\sigma = (c_1, \ldots, c_n, y, V)$.

Forge: $\mathscr{A}_1$ outputs a signature $\sigma^* = (c_1^*, \ldots, c_n^*, y^*, V^*)$ on the tuple $(event^*, m^*, U^*)$, which meets the requirements in Game I.

Solve DL Problem: In order to yield the signature $\sigma^* = (c_1^*, \ldots, c_n^*, y^*, V^*)$ on the tuple $(event^*, m^*, U^*)$, $\mathscr{A}_1$ must perform the query $H_4(event^*, m^*, V^*, A^*, B^*, U^*)$. It is a reasonable assumption that is done at the l^{th} query of H_4 and $\mathscr{A}_1$ obtained a reply value u^*. Due to H_4 is a secure hash function and $u^* = c_1^* + \cdots + c_n^*$, there is at least a $s \in \{1, 2, \ldots, n\}$ such that c_s^* is determined after u^* obtained by $\mathscr{A}_1$. Afterwards, $\mathscr{C}$ rewinds with the same input tape for $\mathscr{A}_1$ and answer all queries consistently except $u^{*\prime}$ returned by the l^{th} H_4 query. So $\mathscr{A}_1$ generates another signature $\sigma^{*\prime} = (c_1^{*\prime}, \ldots, c_n^{*\prime}, y^{*\prime}, V^{*\prime})$, where $u^{*\prime} = c_1^{*\prime} + \cdots + c_n^{*\prime}$. Since $u^{*\prime} \neq u^*$ and $c_s^{*\prime}$ is determined after $u^{*\prime}$ is returned by $\mathscr{C}$, then $c_s^{*\prime} \neq c_s^*$ and $c_i^{*\prime} = c_i^*$ for $i \neq s$. It implies that $y^{*\prime} \neq y^*$. If $ID_s^* = ID^\diamond$, then $y^* = z^* - c_s^*(h^* t^* + k_s^* x + a)$, $y^{*\prime} = z^* - c_s^{*\prime}(h^* t^* + k_s^* x + a)$, and $PK_s^* = PK_j$, so $(t^* P, r^* P) = (t_j P, aP)$. $\mathscr{C}$ finds tuple $(ID_j, t_j, *)$ in the list L_U, computes $k_s^* = H_1(ID_s^*, aP)$, $h^* = H_2(event^*)$, then computes: $a = (c_s^* - c_s^{*\prime})^{-1}(y^{*\prime} - y^*) - h^* t^* - k_s^* x$, where $t^* = t_j$.

Probability: Let $q_{H_i}(i = 1, 2, 3, 4)$, q_U, q_D, q_E, q_R and q_S be the number of Query-$H_i(i = 1, 2, 3, 4)$, Query-UPK, Query-PPK, Query-SV, Replace-TPK and Query-Sign queries, respectively. We denote the three events as follows.

π_1: $\mathscr{A}_1$ did not performs the Query-PPK for $ID^\diamond$.

π_2: $ID^\diamond \in W^*$.

π_3: $ID^\diamond$ is the real signer.

Getting following results is not difficult.

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

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

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

By forking lemma [13], $\mathscr{A}_1$ can generate two signatures with probability $\frac{\varepsilon^2}{66C_{q_{H_4}}^n}$ if $\mathscr{A}_1$ can forge a valid signature with probability $\varepsilon \geq \frac{7C_{q_{H_4}}^n}{2^v}$. Hence, $\mathscr{C}$ can resolve the instance of DL problem with probability $\frac{\varepsilon^2}{66C_{q_{H_4}}^n} \cdot \frac{1}{q_U}$ if $\mathscr{A}_1$ wins with advantage $\varepsilon \geq \frac{7C_{q_{H_4}}^n}{2^v}$.

Theorem 2: If DL problem is hard, then the scheme is unforgeable against Type II adversary in ROM.

Proof: Suppose that the tuple (P, aP) is an instance of the DL problem. $\mathscr{C}$ will compute the value a by acting as the challenger in Game II.

Initialization: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ and $msk = \{x\}$ by running the Setup algorithm, then forwards them to $\mathscr{A}_2$.

Queries: Several lists are set to store the queries and answers. $\mathscr{A}_2$ will execute Query-UPK for an identity ID_i before that is used in any other queries.

- Query-UPK: $\mathscr{C}$ maintains a list L_U of tuple (ID_i, t_i, r_i). $\mathscr{A}_2$ inputs an identity ID_i, $\mathscr{C}$ does as follows:
  At the j^{th} query, selects at random $r_j \in Z_q^*$, sets $ID_j = ID^\diamond$ and $PK_j = PK^\diamond = (aP, r_j P)$. For $i \neq j$, $\mathscr{C}$ selects at random $t_i, r_i \in Z_q^*$, $PK_i = (t_i P, r_i P)$, and stores the query and answer in the list L_U.
- Query-$H_i(i = 1, \ldots, 4)$: Same as that in Theorem 1.
- Replace-TPK: Same as that in the Theorem 1.
- Query-SV: When $\mathscr{A}_2$ issues this query for an identity ID_i. $\mathscr{C}$ fails if $ID_i = ID^\diamond$. Otherwise, $\mathscr{C}$ finds (ID_i, t_i, r_i)

in the list L_U, responds with t_i and adds (ID_i, t_i) to the list L_E.

- Query-PPK: $\mathscr{C}$ maintains a list L_D of tuple (ID_i, d_i). When $\mathscr{A}_2$ issues this query for an identity ID_i. $\mathscr{C}$ finds (ID_i, t_i, r_i) in the list L_U or (ID_i, T_i, T_i') in the list L_R, gives the d_i by executing the Extract-PPK algorithm and adds (ID_i, d_i) to the list L_D.
- Query-Sign: Same as that in Theorem 1.

Forge: $\mathscr{A}_2$ outputs a signature $\sigma^* = (c_1^*, \ldots, c_n^*, y^*, V^*)$ on the tuple $(event^*, m^*, U^*)$, which meets the requirements in Game II.

Probability: In order to yield the signature $\sigma^* = (c_1^*, \ldots, c_n^*, y^*, V^*)$ on the tuple $(event^*, m^*, U^*)$, $\mathscr{A}_2$ must perform the query $H_4(event^*, m^*, V^*, A^*, B^*, U^*)$. It is a reasonable assumption that is done at the l^{th} query of H_4 and $\mathscr{A}_2$ obtained a reply value u^*. Since H_4 is a secure hush function and $u^* = c_1^* + \cdots + c_n^*$, there is at least a $s \in \{1, 2, \ldots, n\}$ such that c_s^* is determined after u^* obtained by $\mathscr{A}_2$. Afterwards, $\mathscr{C}$ rewinds with the same input tape for $\mathscr{A}_2$ and answer all queries consistently except $u^{*\prime}$ returned by the l^{th} H_4 query. So $\mathscr{A}_2$ generates another signature $\sigma^{*\prime} = (c_1^{*\prime}, \ldots, c_n^{*\prime}, y^{*\prime}, V^{*\prime})$, where $u^{*\prime} = c_1^{*\prime} + \cdots + c_n^{*\prime}$. Since $u^{*\prime} \neq u^*$ and $c_s^{*\prime}$ is determined after $u^{*\prime}$ is returned by $\mathscr{C}$, then $c_s^{*\prime} \neq c_s^*$ and $c_i^{*\prime} = c_i^*$ for $i \neq s$. It implies that $y^{*\prime} \neq y^*$. If $ID_s^* = ID^\diamond$, then $y^* = z^* - c_s^*(h^*a + k_s^*x + r^*)$, $y^{*\prime} = z^* - c_s^{*\prime}(h^*a + k_s^*x + r^*)$, and $PK_s^* = PK_j$, namely $(t^*P, r^*P) = (aP, r_jP)$. $\mathscr{C}$ finds tuple $(ID_j, t_j, *)$ in the list L_U, computes $k_s^* = H_1(ID_s^*, r_jP)$ and $h^* = H_2(event^*)$, then computes $a = h^{*-1}[(c_s^* - c_s^{*\prime})^{-1}(y^{*\prime} - y^*) - r^* - k_s^*x]$, where $r^* = r_j$.

Probability: Let $q_{H_i}(i = 1, 2, 3, 4)$, q_U, q_D, q_E, q_R and q_S be the number of Query-$H_i(i = 1, 2, 3, 4)$, Query-UPK, Query-PPK, Query-SV, Replace-TPK and Query-Sign, respectively.

It is a reasonable assumption that $L_E \cap L_R = \emptyset$ and we denote the three events as follows.

π_1: $\mathscr{A}_2$ performed neither Replace-TPK nor Query-SV for $ID^\diamond$.

π_2: $ID^\diamond \in W^*$.

π_3: $ID^\diamond$ is the actual signer.

Getting following results is not difficult.

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

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

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

By forking lemma [13], $\mathscr{A}_2$ can generate two signatures with probability $\frac{\varepsilon^2}{66C_{q_{H_4}}^n}$ if $\mathscr{A}_2$ can forge a valid signature with probability $\varepsilon \geq \frac{7C_{q_{H_4}}^n}{2^v}$. Hence, $\mathscr{C}$ can resolve the instance of DL problem with probability $\frac{\varepsilon^2}{66C_{q_{H_4}}^n} \cdot \frac{1}{q_U}$ if $\mathscr{A}_2$ wins with advantage $\varepsilon \geq \frac{7C_{q_{H_4}}^n}{2^v}$.

Theorem 3: The scheme is anonymous against Type I adversary in ROM.

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ by running the Setup algorithm, then forwards it to the $\mathscr{A}_1$.

Phase 1: $\mathscr{A}_1$ executes various queries as those in Theorem 1.

Challenge: $\mathscr{A}_1$ inputs a tuple $(event, m, U, ID_0, ID_1)$ meeting the requirements in Game III. $\mathscr{C}$ selects at random a bit $\mu \in \{0, 1\}$, generates a signature $\sigma = \text{Sign}(event, m, U, d_\mu, t_\mu)$ and sends it to the $\mathscr{A}_1$.

Phase 2: $\mathscr{A}_1$ executes various queries as those in Phase 1 and complies with the constrains in Game III.

Response: $\mathscr{A}_1$ returns a bit $\mu' \in \{0, 1\}$.

Due to $\mathscr{A}_1$ does not know the values d_0 and d_1, $\mathscr{A}_1$ cannot verify the following equations: $V = (d_0 + ht_0)E$ and $V = (d_1 + ht_1)E$.

For a signature σ generated by sign algorithm, since H_2, H_3 is two secure hash function, then $E = H_2(event)$ and $h = H_3(event)$ are the distributed uniformly over G and Z_q^*, respectively. So $V = (d_s + ht_s)E$ is also the distributed uniformly over G. If $ID_i \in W$ is not the real signer, z and c_i are independently selected and are evenly distributed over Z_q^*, then $A = zE + \sum_{i=1, i\neq s}^n c_i V$ and $B = zP + \sum_{i=1, i\neq s}^n c_i(hT_i + R_i + k_i P_{pub})$ are distributed uniformly over G. Since u is the output of the secure hash function H_4, then u is distributed uniformly over Z_q^*, it follows that $c_s = u - \sum_{i=1, i\neq s}^n c_i$ and $y = z - c_s(d_s + ht_s)$ are also distributed uniformly over Z_q^*. In conclusion, all the parameters mentioned above are evenly distributed. So it is conclude that the adversary $\mathscr{A}_1$ has no advantage in recognizing the real signer over random guessing.

Theorem 4: The scheme is anonymous against Type II adversary in ROM.

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ and $msk = \{x\}$ by running the Setup algorithm, then forwards them to the $\mathscr{A}_2$.

Phase 1: $\mathscr{A}_2$ executes various queries as those in Theorem 2.

Challenge: $\mathscr{A}_2$ inputs a tuple $(event, m, U, ID_0, ID_1)$ meeting the requirements in Game IV. $\mathscr{C}$ selects at random a bit $\mu \in \{0, 1\}$, generates a signature $\sigma = \text{Sign}(event, m, U, d_\mu, t_\mu)$ and sends it to the $\mathscr{A}_2$.

Phase 2: $\mathscr{A}_2$ executes various queries as those in Phase 1 and complies with the constrains in Game IV.

Response: $\mathscr{A}_2$ returns a bit $\mu' \in \{0, 1\}$.

Due to $\mathscr{A}_2$ does not know the values t_0 and t_1, $\mathscr{A}_1$ cannot verify the following equations: $V = (d_0 + ht_0)E$ and $V = (d_1 + ht_1)E$.

The next analysis is the same as in Theorem 3.

Lemma 1: In ROM, $\mathscr{A}_1$ must obtain the partial private key d_s of the real signer before generating a valid signature $(\sigma = (c_1, \ldots, c_n, y, V), event, m, U)$. (where $ID_s \in W$).

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ by running the Setup algorithm, then forwards it to the $\mathscr{A}_1$.

Firstly, $\mathscr{A}_1$ executes various queries as those in Theorem 1.

Next, $\mathscr{A}_1$ produces a valid signature

$$(\sigma = (c_1, \ldots, c_n, y, V), event, m, U),$$

where $V = (\lambda + ht_s)E$ for some $\lambda \in Z_q^*$ for the first run. $\mathscr{A}_1$ can get the secret value of anyone in W by performing Query-SV.

Then, $\mathscr{C}$ gets another signature

$$(\sigma' = (c_1', \ldots, c_n', y', V), event, m, U)$$

by rewinding $\mathscr{A}_1$ with a different answer for the Query-H_4. Where $V, A, B \in G$ and the values $\lambda, t_s \in Z_q^*$ are same during both signatures. $\mathscr{C}$ outputs two different values u, u' for the query $H_4(event, m, V, A, B, U)$ and gives the same answers for all other queries in both signatures.

Since $u = c_1 + \cdots + c_n$ and $u' = c_1' + \cdots + c_n'$, then there exists a $c_s(c_s')(1 \le s \le n)$ that is calculated after $u(u')$ is returned. That implies that $c_s' \ne c_s$ and $c_i' = c_i$ for $i \ne s$. It follows that $y \ne y'$ where $y = z - c_s(ht_s + \lambda)$, $y' = z - c_s'(ht_s + \lambda)$. So $\mathscr{C}$ can compute $\lambda = (c_s - c_s')^{-1}(y' - y) - ht_s$. By Theorem 1, $\mathscr{A}_1$ can not forge a valid signature. The signature must be generated by using the values d_s and t_s. Since $V = (\lambda + ht_s)E = (d_s + ht_s)E$, then $\lambda = d_s$. That implies that $\mathscr{A}_1$ must obtain the partial private key d_s of the real signer.

Lemma 2: In ROM, $\mathscr{A}_2$ must obtain the secret value t_s of the real signer before generating a valid signature $(\sigma = (c_1, \ldots, c_n, y, V), event, m, U)$ (where $ID_s \in W$).

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ and $msk = \{x\}$ by running the Setup algorithm, then forwards them to the $\mathscr{A}_2$.

Firstly, $\mathscr{A}_2$ executes various queries as those in Theorem 2.

Next, $\mathscr{A}_2$ produces a valid signature

$$(\sigma = (c_1, \ldots, c_n, y, V), event, m, U),$$

where $V = (d_s + h\lambda)E$ for some $\lambda \in Z_q^*$ for the first run. $\mathscr{A}_2$ can get the partial private key of anyone in W by performing Query-PPK.

Then, $\mathscr{C}$ gets another signature

$$(\sigma' = (c_1', \ldots, c_n', y', V), event, m, U)$$

by rewinding $\mathscr{A}_2$ with a different value for the Query-H_4, where $V, A, B \in G$ and the values $d_s, \lambda \in Z_q^*$ are same during both signatures. $\mathscr{C}$ outputs two different values u, u' for the query $H_4(event, m, V, A, B, U)$ and gives the same answers for all other queries in both signatures.

Since $u = c_1 + \cdots + c_n$ and $u' = c_1' + \cdots + c_n'$, then there exists a $c_s(c_s')(1 \le s \le n)$ which is determined after $u(u')$ is returned. That implies that $c_s' \ne c_s$ and $c_i' = c_i$ for $i \ne s$. It follows that $y \ne y'$ where $y = z - c_s(h\lambda + d_s)$, $y' = z - c_s'(h\lambda + d_s)$. So $\mathscr{C}$ can compute $\lambda = h^{-1}[(c_s - c_s')^{-1}(y' - y) - d_s]$. By Theorem 2, $\mathscr{A}_2$ can not forge a valid signature. The signature must be generated by using the values d_s and t_s. Since $V = (d_s + h\lambda)E = (d_s + ht_s)E$, then $\lambda = t_s$. That implies that $\mathscr{A}_2$ must obtain the secret value t_s of the real signer.

Theorem 5: The scheme is linkable against the Type I adversary in ROM.

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ by running the Setup algorithm, then forwards it to the $\mathscr{A}_1$.

Firstly, $\mathscr{A}_1$ executes various queries as those in Theorem 1.

Next, $\mathscr{A}_1$ produces two valid signatures $(\sigma_1, event, m_1, U_1)$ and $(\sigma_2, event, m_2, U_2)$ that fulfill the requirements as defined in Game V. Where $U_k = W_k \bigcup \{PK_{ki} : ID_{ki} \in W_k\}$, $W_k = \{ID_{k1}, \ldots, ID_{kn}\}$ for $k = 1, 2$,

Since $\sigma_1 = (\cdot, V_1)$ and $\sigma_2 = (\cdot, V_2)$ are unlinkable, then $V_1 \ne V_2$, where $V_1 = (ht_{1s} + d_{1s})E$, $V_2 = (ht_{2s} + d_{2s})E$. From Lemma 1, $\mathscr{A}_1$ have obtained the values d_{1s} and d_{2s}. It contradicts with the fact that $\mathscr{A}_1$ gets the partial private key of at most one user in $W_1 \bigcup W_2$.

Therefore, $\mathscr{A}_1$' advantage in Game V is negligible.

Theorem 6: The scheme is linkable against the Type II adversary in ROM.

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ and $msk = \{x\}$ by running the Setup algorithm, then forwards them to the $\mathscr{A}_2$.

First of all, $\mathscr{A}_2$ executes various queries as those in Theorem 2.

Next, $\mathscr{A}_2$ produces two valid signatures $(\sigma_1, event, m_1, U_1)$ and $(\sigma_2, event, m_2, U_2)$ that fulfill the requirements as defined in Game VI. Where $U_k = W_k \bigcup \{PK_{ki} : ID_{ki} \in W_k\}$, $W_k = \{ID_{k1}, \ldots, ID_{kn}\}$ for $k = 1, 2$.

Since $\sigma_1 = (\cdot, V_1)$ and $\sigma_2 = (\cdot, V_2)$ are unlinkable, then $V_1 \ne V_2$, where $V_1 = (ht_{1s} + d_{1s})E$, $V_2 = (ht_{2s} + d_{2s})E$. From Lemma 2, $\mathscr{A}_2$ have obtained the values t_{1s} and t_{2s}. It contradicts with the fact that $\mathscr{A}_2$ gets the secret value of at most one user in $W_1 \bigcup W_2$.

Therefore, $\mathscr{A}_2$' advantage in Game VI is negligible.

Theorem 7: The scheme is non-slanderable against the Type I adversary in ROM.

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ by running the Setup algorithm, then forwards it to the $\mathscr{A}_1$.

First of all, $\mathscr{A}_1$ executes various queries as those in Theorem 1.

Next, $\mathscr{A}_1$ submits a tuple $(event, m, U, ID_s)$, where $ID_s \in W$, $\mathscr{C}$ provides $\mathscr{A}_1$ with

$$\sigma = (\cdot, V) = \mathrm{Sign}(event, m, U, d_s, t_s).$$

Once again, $\mathscr{A}_1$ executes various queries.

Finally, $\mathscr{A}_1$ outputs another signature

$$(\sigma' = (\cdot, V'), event, m', U')$$

that meets the requirements in Game VII, where $U' = W' \bigcup \{PK_i : ID_i \in W'\}$.

From Lemma 1, $\mathscr{A}_1$ must obtain the partial private key d_s' of the identity $ID_s' \in W'$, where $E = H_2(event)$, $h = H_3(event)$, $V = (ht_s + d_s)E$ and $V' = (ht_s' + d_s')E$. $\mathscr{A}_1$ cannot compute the value $ht_s + d_s$ from the value V (DL problem). Since the two

TABLE 2. Cryptographic operation time (in milliseconds).

T_{bp}	T_{hp}	T_{sm-G_1}	T_{exp-G_2}	T_{sm-G}
32.713	33.582	13.405	2.249	3.335

signatures are linkable, then $V = V'$. It follows that $d'_s = d_s$ and $t'_s = t_s$. In other words, $\mathscr{A}_1$ must get d_s. That contradicts with that $\mathscr{A}_1$ does not get the partial private key d_s of the identity $ID_s \in W$.

Therefore, $\mathscr{A}_1$' advantage in Game VII is negligible.

Theorem 8: The scheme is non-slanderable against the Type II adversary in ROM.

Proof: $\mathscr{C}$ obtains the $params = \{G, q, P, P_{pub} = xP, H_1 \sim H_4\}$ and $msk = \{x\}$ by running the Setup algorithm, then forwards them to the $\mathscr{A}_2$

Firstly, $\mathscr{A}_2$ executes various queries as those as those in Theorem 2.

Next, $\mathscr{A}_2$ submits a tuple $(event, m, U, ID_s)$, where $U = W \bigcup \{PK_i : ID_i \in W\}$ and $ID_s \in W$, $\mathscr{C}$ provides $\mathscr{A}_2$ with $\sigma = (\cdot, V) = \text{Sign}(event, m, U, d_s, t_s)$.

Once again, $\mathscr{A}_2$ executes various queries.

Finally, $\mathscr{A}_2$ outputs another signature

$$(\sigma' = (\cdot, V'), event, m', U')$$

that meets the requirements in the Game VIII, where $U' = W' \bigcup \{PK_i : ID_i \in W'\}$.

From Lemma 2, $\mathscr{A}_2$ must obtain the secret value t'_s of the identity $ID'_s \in W'$, where $E = H_2(event)$, $h = H_3(event)$, $V = (ht_s + d_s)E$ and $V' = (ht'_s + d'_s)E$. $\mathscr{A}_2$ cannot compute the value $ht_s + d_s$ from the value V (DL problem). Since the two signatures are linkable, then $V = V'$. It follows that $d'_s = d_s$ and $t'_s = t_s$. In other words, $\mathscr{A}_2$ must get the value t_s. That contradicts with that $\mathscr{A}_2$ does not get the secret value t_s of the identity $ID_s \in W$.

Therefore, $\mathscr{A}_2$' advantage in Game V is negligible.

VII. EFFICIENCY AND COMPARISON

In this section, we compare the performance of the four schemes. For convenience, we define several notations as follows.

T_{bp}: A bilinear pairing operation.

T_{hp}: A hash-to-point operation.

T_{sm-G_1}: A scale multiplication operation in G_1.

T_{exp-G_2}: An exponentiation operation in G_2.

T_{sm-G}: A scale multiplication operation in G.

In order to be fair and reasonable, we will use third-party data to analyze the efficiency of several schemes. By executing the cryptographic operations in a mobile phone (Samsung Galaxy S5 with 2G bytes memory, a Quad-core 2.45G processor and the Google Android 4.4.2 operating system), He *et al.* [14] obtained the running time, as shown in Table 2. In their experiments, to realize the 1024-bits RSA level security, they used an Ate pairing $e : G_1 \times G_1 \to G_2$, where G_1 with order q is an additive group defined on a super singular elliptic curve $y^2 = x^3 + x$ over a finite field F_p, and the sizes of p and q are 160 bits and 512 bits, respectively. Additional, they used an additive group G with order q, which is defined on a non-singular elliptic curve over a prime field F_p, where both sizes of p and q are 160 bits.

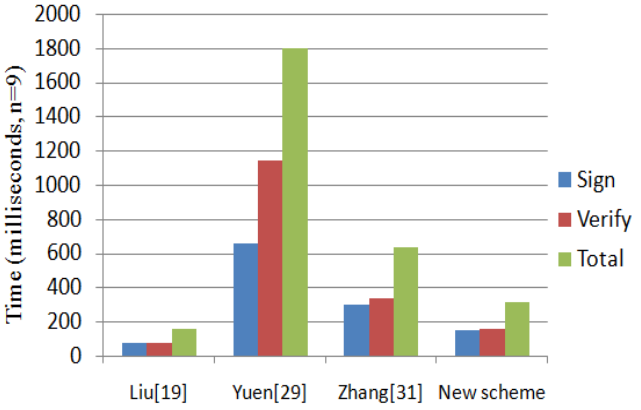

FIGURE 2. Computation cost.

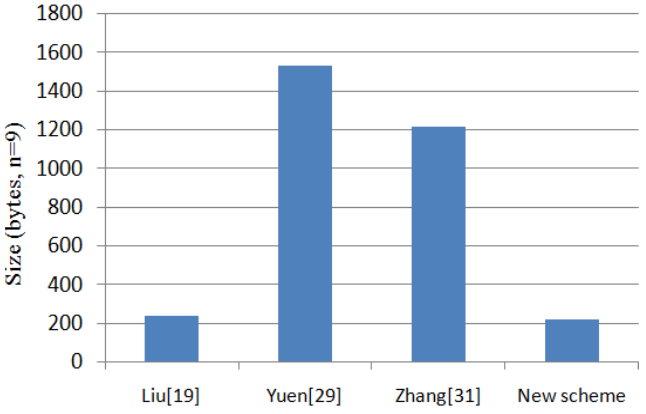

FIGURE 3. Storage expenses.

First of all, we analyze the computation cost, as shown in Table 3 and Figure 2. Suppose that each ring contains n users. For convenience, we let $n = 9$. Liu *et al.*'s scheme [19] needs 2 hash-to-point operations and $2n + 8$ scale multiplication operations in G, the computation time is $2 \times 33.582 + (2 \times 9 + 8) \times 3.335 = 153.874$ ms. Yuen *et al.*'s scheme [29] needs $n+10\sqrt{n}+12$ scale multiplication operations in G_1 and $8\sqrt{n}+10$ bilinear pairing operations, the computation time is $(9+10\sqrt{9}+12) \times 13.405 + (8\sqrt{9}+10) \times 32.713 = 1795.899$ ms. Zhang *et al.*'s scheme [31] needs $4n + 4$ scale multiplication operations in G_1 and 3 bilinear pairing operations, the computation time is $(4 \times 9 + 4) \times 13.405 + 3 \times 32.713 = 634.339$ ms. The new scheme needs 2 hash-to-point operations and $8n + 1$ scale multiplication operations in G, the computation time is $2 \times 33.582 + (8 \times 9 + 1) \times 3.335 = 310.619$ ms.

Follow on, we evaluate the size of signatures, as shown in Table 3 and Figure 3. It is assumed that each ring contains $n(= 9)$ users. In the scheme [19], a signature contains $n + 3$

TABLE 3. Comparison of four schemes.

Scheme	Liu [19]	Yuen [29]	Zhang [31]	New scheme
Sign	$(n+4)T_{sm-G}+T_{hp}$	$(n+10\sqrt{n}+10)T_{sm-G_1}$	$(2n+4)T_{sm-G_1}$	$(4n-1)T_{sm-G}+T_{hp}$
Verify	$(n+4)T_{sm-G}+T_{hp}$	$(8\sqrt{n}+10)T_{bp}+2T_{sm-G_1}$	$3T_{bp}+2nT_{sm-G_1}$	$(4n+2)T_{sm-G}+T_{hp}$
Time	$6.67n+93.844$	$13.405n+395.754\sqrt{n}+487.99$	$53.62n+151.759$	$26.68n+70.499$
(n=9)	(153.874)	(1795.899)	(634.339)	(310.619)
Length	$(n+3)\cdot 20$	$(6+6\sqrt{n})\cdot 64$	$(2n+1)\cdot 64$	$(n+2)\cdot 20$
(n=9)	(240)	(1536)	(1216)	(220)
Linkable	Yes	Yes	No	Yes
Public key	PKI	PKI	Certificateless	Certificateless

points from an additive group G, the size is $[(9+3)\times 160]/8 = 240$ bytes. In the scheme [29], a signature contains $6\sqrt{n}+6$ points from $E/F_p : y^2 = x^3+1$, the size is $[(6\sqrt{9}+6)\times 512]/8 = 1536$ bytes. In the scheme [31], a signature contains $2n+1$ points from $E/F_p : y^2 = x^3+1$, the size is $[(2\times 9+1)\times 512]/8 = 1216$ bytes. In the new scheme, a signature contains $n+2$ points from an additive group G, the size is $[(9+2)\times 160]/8 = 220$ bytes.

VIII. CONCLUSION

In a "regular" ring signature scheme, the signer can generate multiple different signatures for the same message without being discovered by the verifier. As a result, users may abuse their signing rights. LRS solves this problem, it not only achieves the anonymity of the signer, but also prevents the abuse of signing rights. It is suitable for e-commerce to protect the privacy of users. All LRS schemes currently known are constructed from public key infrastructure or identity-based cryptography. CL-PKC removes the certificate while avoiding key escrow. In this paper, we introduced the system model and security requirements for CL-LRS scheme and presented a concrete construction, then showed the security proofs in ROM. As far as we know, the scheme is the first CL-LRS scheme. Since no pairing operation is required, our scheme is very effective.

REFERENCES

[1] M. H. Au, J. K. Liu, W. Susilo, and T. Yuen, "Constant-size ID-based linkable and revocable-iff-linked ring signature," in *Proc. 7th Int. Conf. Cryptol. India*, in Lecture Notes in Computer Science, vol. 4329, 2006, pp. 364–378.

[2] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction," *Theor. Comput. Sci.*, vol. 469, pp. 1–14, Jan. 2013.

[3] S. S. Al-Riyami and K. G. Paterson, "Certificateless public cryptography," in *Advances in Cryptology*, in Lecture Notes in Computer Science, vol. 2894, 2003, pp. 452–473.

[4] C. Baum, H. Lin, and S. Oechsner, "Towards practical lattice-based one-time linkable ring signatures," in *Proc. Int. Conf. Inf. Commun. Secur.*, in Lecture Notes in Computer Science, vol. 11149, 2018, pp. 303–322.

[5] X. Boyen and T. Haines, "Forward-secure linkable ring signatures," in *Proc. Australas. Conf. Inf. Secur. Privacy*, in Lecture Notes in Computer Science, vol. 10946, 2018, pp. 245–264.

[6] S. Chang, D. S. Wong, Y. Mu, and Z. Zhang, "Certificateless threshold ring signature," *Inf. Sci.*, vol. 179, no. 20, pp. 3685–3696, Sep. 2009.

[7] Z. Cheng and L. Chen, "Certificateless public key signature schemes from standard algorithms," in *Proc. Int. Conf. Inf. Secur. Pract. Exper.*, in Lecture Notes in Computer Science, vol. 11125, 2018, pp. 179–197.

[8] S. S. M. Chow, W. Susilo, and T. H. Yuen, "Escrowed linkability of ring signatures and its applications," in *Proc. 1st Int. Conf. Cryptol. Vietnam*, in Lecture Notes in Computer Science, vol. 4341, 2006, pp. 175–192.

[9] S. Chow and W. Yap, "Certificateless ring signature," Cryptol. ePrint Arch., Tech. Rep. 2007/236. [Online]. Available: http://eprint.iacr.org/2007/236

[10] L. Deng, "Certificateless ring signature based on RSA problem and DL problem," *RAIRO-Theor. Inform. Appl.*, vol. 49, no. 4, pp. 307–318, Oct. 2015.

[11] L. Deng, Y. Jiang, and B. Ning, "Identity-based linkable ring signature scheme," *IEEE Access*, vol. 7, pp. 153969–153976, 2019.

[12] E. Fujisaki, "Sub-linear size traceable ring signatures without random oracles," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E95-A, no. 1, pp. 151–166, 2012.

[13] J. Herranz and G. Sáez, "Forking lemmas for ring signature schemes," in *Proc. Int. Conf. Cryptol. India*, in Lecture Notes in Computer Science, vol. 2904, 2003, pp. 266–279.

[14] D. He, H. Wang, L. Wang, J. Shen, and X. Yang, "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Comput.*, vol. 21, no. 22, pp. 6801–6810, Nov. 2017.

[15] I. R. Jeong, J. O. Kwon, and D. H. Lee, "Ring signature with weak linkability and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1145–1148, Aug. 2008.

[16] I. R. Jeong, J. O. Kwon, and D. H. Lee, "Analysis of revocable-iff-linked ring signature scheme," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E92-A, no. 1, pp. 322–325, 2009.

[17] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Proc. 9th Australas. Conf. Inf. Secur. Privacy*, in Lecture Notes in Computer Science, vol. 3108, 2004, pp. 325–335.

[18] J. K. Liu and D. S. Wong, "Linkable ring signatures: Security models and new schemes," in *Proc. Int. Conf. Comput. Sci. Appl.*, in Lecture Notes in Computer Science, vol. 3481, 2005, pp. 614–623.

[19] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 157–165, Jan. 2014.

[20] S. Noether, "Ring signature confidential transactions for Monero," Cryptol. ePrint Arch., Tech. Rep. 2015/1098, 2015. [Online]. Available: http://eprint.iacr.org/

[21] R. Rivest, A. Shamir, and Y. Taumn, "How to leak a secret," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Sciences), vol. 2248. Australia, 2001, pp. 552–565.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology. CRYPTO* (Lecture Notes in Computer Science), vol. 196. USA, 1984, pp. 47–53.

[23] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero," in *Proc. Eur. Symp. Res. Comput. Secur.*, in Lecture Notes in Computer Science, vol. 10493, 2017, pp. 456–474.

[24] W. A. A. Torres, R. Steinfeld, A. Sakzad, J. K. Liu, V. Kuchta, N. Bhattacharjee, M. H. Au, and J. Cheng, "Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice RingCT v1.0)," in *Proc. Australas. Conf. Inf. Secur. Privacy*, Lecture Notes in Computer Science, vol. 10946, 2018, pp. 558–576.

[25] P. P. Tsang, V. K. Wei, T. K. Chan, M. H. Au, J. K. Liu, and D. S. Wong, "Separable linkable threshold ring signatures," in *Proc. Int. Conf. Cryptol. India*, in Lecture Notes in Computer Science, vol. 3348, 2004, pp. 384–398.

[26] P. Tsang and V. Wei, "Short linkable ring signatures for e-voting, e-cash and attestation," in *Proc. Int. Conf. Inf. Secur. Practice Exper.*, in Lecture Notes in Computer Science, vol. 3439, 2005, pp. 48–60.
[27] P. P. Tsang, M. H. Au, J. K. Liu, W. Susilo, and D. S. Wong, "A suite of non-pairing ID-based threshold ring signature schemes with different levels of anonymity," in *Proc. Int. Conf. Provable Secur.*, in Lecture Notes in Computer Science, vol. 6402, 2010, pp. 166–183.
[28] H. Wang and S. Han, "A provably secure threshold ring signature scheme in certificateless cryptography," in *Proc. Int. Conf. Inf. Sci. Manage. Eng.*, Aug. 2010, pp. 105–108.
[29] 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.*, vol. 56, no. 4, pp. 407–421, Apr. 2013.
[30] L. Zhang, F. Zhuang, and W. Wu, "A provably secure ring signature scheme in certificateless cryptography," in *Proc. Int. Conf. Provable Secur.*, in Lecture Notes in Computer Science, vol. 4784, 2007, pp. 103–121.
[31] Y. Zhang, J. Zeng, W. Li, and H. Zhu, "A certificateless ring signature scheme with high efficiency in the random oracle model," *Math. Problems Eng.*, vol. 2017, Jun. 2017, Art. no. 7696858.
[32] D. Zheng, X. Li, K. Chen, and J. Li, "Linkable ring signatures from linear feedback shift register," in *Proc. Int. Conf. Embedded Ubiquitous Comput.*, in Lecture Notes in Computer Science, vol. 4809, 2007, pp. 716–727.

LUNZHI DENG received the B.S. and M.S. degrees from Guizhou Normal University, Guiyang, China, in 2002 and 2008, respectively, and the Ph.D. degree from Xiamen University, Xiamen, China, in 2012. He is currently a Professor with the School of Mathematical Sciences, Guizhou Normal University. His recent research interests include cryptography and information safety.

HONGYU SHI received the B.S. degree from Guizhou Normal University, Guiyang, China, in 2018. She is currently a Graduate Student with Guizhou Normal University. Her recent research interests include cryptography protocol and information safety.

YAN GAO received the B.S. degree from Guizhou Normal University, Guiyang, China, in 2019. She is currently a Graduate Student with Guizhou Normal University. Her recent research interests include cryptography protocol and information safety.