Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8021267, 14 pages
https://doi.org/10.1155/2022/8021267

Research Article

Revocable One-Time Ring Signature from Pairings

Xu Han,[1,2] **Dawei Zhang**,[1,2] **Zongmin Huang**,[1,2] **Shuang Yao**,[1,2] **and Zuodong Wu**[1,2]

[1]*School of Computer and Information Technology, Beijing Jiaotong University, Beijing, Beijing 100044, China*
[2]*Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing, Beijing 100044, China*

Correspondence should be addressed to Dawei Zhang; dwzhang@bjtu.edu.cn

Received 5 April 2022; Accepted 22 July 2022; Published 18 August 2022

Academic Editor: Yin Zhang

Ring signature is an anonymous signature that allows a person to sign a message on behalf of a self-formed group while concealing the identification of the signer. However, due to its anonymity and unlinkability, malicious or irresponsible signers can easily attack the signature without any responsibility in some scenarios. In this paper, we propose a novel revocable one-time ring signature (roRS) scheme from bilinear pairings, which introduces linkability and mandatory revocability into ring signature. In particular, linkability can resist the double-signing attack and mandatory revocability guarantees that a revocation authority can identify the actual signer when a suspicious signer appears in any situation. The computational complexity of pairing computations is constant, and the time of the revocation phase is more efficient than previous schemes. Furthermore, our scheme is provable secure in the random oracle model, using DL, CDH, and DBDH assumptions.

1. Introduction

Ring signature, initially proposed in 2001 by Rivest [1], is a variant of digital signature, which can prove that one among a set of spontaneous parties has already signed a message, without revealing the actual signer. And these spontaneous parties compose a particular set called a "ring." More specifically, a ring member can sign the signature without reveal any identity information, namely, a verifier who uses ring members' public keys only know whether the signature is true or not and cannot find out the actual signer, and the verifier has no clue who the signer is. As shown in Figure 1, first step, the actual signer uses private key sk_j and randomly chooses $r_j \in \mathbb{Z}_p^*$ to generate L_j by using the commit function $C(sk_j, r_j)$; then, the signer uses L_j to compute the (j+1)-th challenge c_{j+1} by hash function H; signer randomly picks a response z_{j+1} and the (j+1)-th user's public key pk_{j+1} to reconstruct the L_{j+1} by the verify function Ver and generates c_{j+2} by hash function H; then, the ring is formed sequentially; finally, the signer uses sk_j, c_j, r_j to compute z_j by the response function Z. In the whole process of generating a ring, we only need the actual signer's private key sk_j and a set of users' public keys which contains pk_j. In the view of the actual signer, users except the actual signer can be seen as decoys. When the verifier does the verifications, he/she does not know any information about the knowledge of the actual signer. As for security, ring signature not only provides regular properties, such as correctness and unforgeability which any signature schemes must possess, but also has the special feature, anonymity. Correctness requires a ring member who represents a ring to sign a message and unforgeability demands that an efficient adversary cannot forge a signature on behalf of a ring which the adversary knows nothing about one secret key of ring members. As for anonymity, it allows that ring signature schemes cannot leak any information about the identity of the actual signer, that is, no one can tell which key was used to produce a signature.

As an extension of ring signature, one-time ring signature can be known as linkable ring signature; the slight difference between these two kinds of ring signature is that

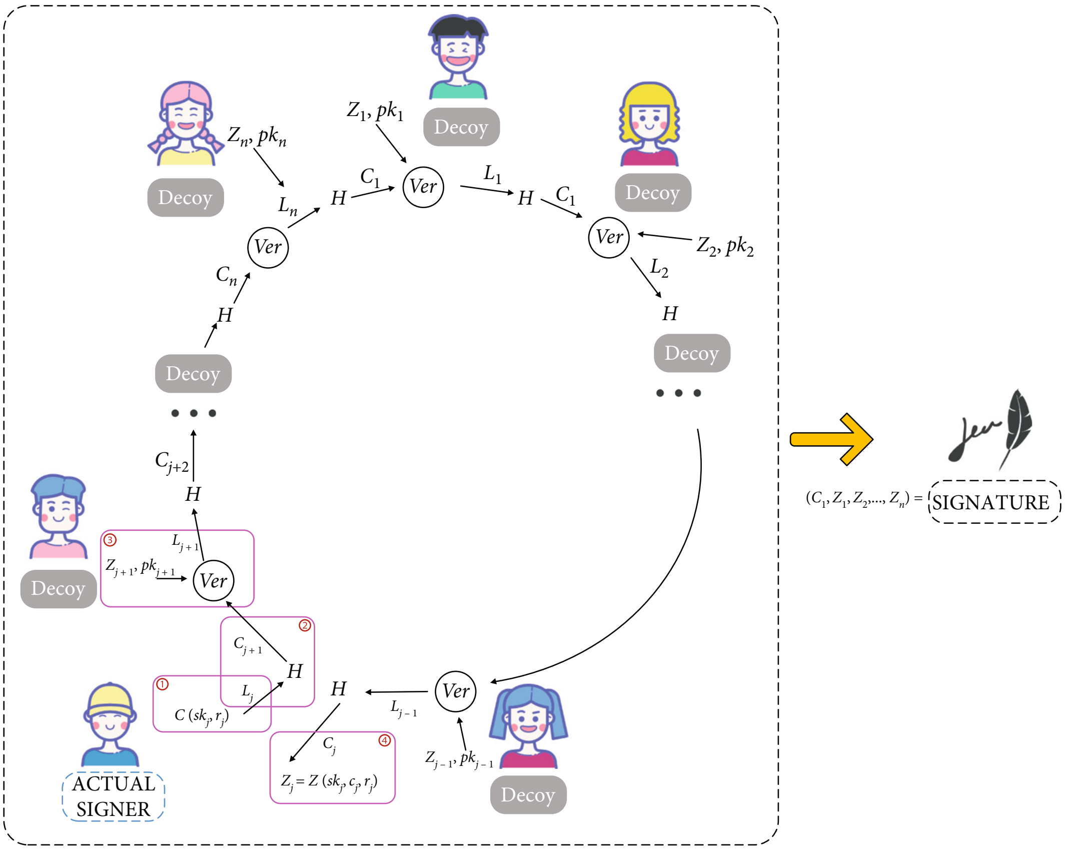

Figure 1: Ring signature.

signers in one-time ring signature use one-time key images to sign a signature, while signers in linkable ring signature take static ones. So we just need to introduce linkable ring signature in this section. Liu et al. [2] first put forward the concept of linkable ring signature (lRS). Beside the regular properties of ring signature, lRS provides two more special properties: non-slanderability and linkability. Non-slanderability guarantees that a ring member should not be entrapped that he has signed twice. Linkability requires that two signatures with the same ring on random messages must be linked if signed by the identical signer; thus, it can defeat the double-signing (double-spending) attack. This property is suitable in some practical applications; one scenario is on detecting double-voting in e-voting [3] systems. At the beginning of e-voting systems, we use ring signature schemes to implement the systems for its spontaneity, and there is no registration phase. The only requirement is that everyone has a public key pair which is considered as a well-known assumption in a ring. However, using classic ring signature as e-voting has a main problem. Anyone can vote more than once without being detected as ring signature schemes are unlinkable and anonymous. Thus, using lRS can solve this problem as double voting (double signing) can be detected easily, and anyone only can vote once in the system. Beside e-voting systems, lRS can also apply in other actual scenarios, such as ad hoc network authentication [4], blockchain-based applications [5, 6], and cryptocurrencies (Monero [7]). But in some actual transactions based on ring signature, when a ring signer has committed an offence, such as money laundering, online extortion, and terrorist financing, the authority needs to find out who is the actual signer among the ring members. Since lRS cannot let the actual signer be identified, the revocability of ring signature becomes necessary. Revocability requires that the authority can revoke the anonymity of ring signers when a suspicious signer does a transaction.

In order to solve the above problem, we propose a novel revocable one-time ring signature (roRS) scheme. Our scheme can be applied in some blockchain transactions. For example, by using the functionality of ring signature, Monero [7] protects the privacy of the signer's identity and provides autonomous mixing in transactions, but the unconditional anonymity of the ring signature makes difficult to regulate transactions for authorities. As for our scheme, a verifier can prevent the user's double-spending behavior according to the linkability during the transactions, and a revocation authority can recover the public key of the transaction user by using the revocability of our scheme, thus restoring the transaction user's identity.

1.1. Related Work. Rivest et al. [1] in 2001 proposed the first ring signature scheme, using trapdoor permutation based on the discrete logarithm problem assumption. Hereafter, many of schemes [8–11] followed this idea, but using different techniques has come out. For instance, Boneh et al. [12] first proposed a ring signature using bilinear pairings based on co-computational Diffie-Hellman (co-CDH) assumption. Cayrel et al. [13] presented a lattice-based ring signature scheme with modifying Melchor's code-based method [14] to make the short integer solution (SIS) problem as a security assumption. As for proving the membership problem in ring signature, most of these schemes use non-interactive witness indistinguishable (NIWI) proofs [15] or dynamic accumulator [16] to be more efficient, and readers who want to learn more refer to [17–21].

Then, Liu et al. [2] first proposed a linkable ring signature (lRS) scheme in 2004. This scheme inherits the anonymity of ring signature and provides a new property called linkability to resist double-spending attempts in real transactions, and it is proven to be secured in the random oracle model. Tsang et al. [22] constructed the first separable linkable ring signature scheme with introducing the security notions of accusatory linkability and non-slanderability. Liu and Wong [23] enhanced the security model for adapting to new attacking scenarios, and proposed two polynomial-structured lRS schemes based on zero knowledge proof. In 2007, Zheng [24] designed an lRS scheme based on linear feedback shift register under discrete logarithm assumption. In 2014, Liu et al. [25] put forward the first unconditional anonymous lRS scheme and provide mandatory linkability. Recently, Noether [26] proposed a dual lRS scheme which key images are tied to both output one-time public keys in a dual, and this can be considered using in non-interactive refund transactions in Monero. Tang et al. [27] presented an identity-based lRS scheme by employing trapdoor generation and rejection sampling as the basic building tool under the SIS problem on NTRU lattice. Hu et al. [28] designed a lattice-based lRS scheme under the well-studied standard lattice assumptions (SIS and LWE) in the standard model.

Revocable ring signature, also called traceable ring signature, is presented to reduce and even revoke the anonymity of the signers mainly. In 2007, Liu et al. [29] first proposed a revocable ring signature that authorities can mandatory revoke the anonymity of the actual signer when authorities need in some scenarios, but this scheme cannot provide linkability against the double-signing attack. Fujisaki et al. [30] put forward a traceable ring signature which only can trace a signer who was double-signing, that is, the traceability is not mandatory. The similar constructions can be found in [17–21]. In [31], Fujisaki presented the first secure traceable ring signature scheme without random oracles in the common reference string model, and the signature size grows linearly with $\sqrt{n}$ where n is the number of users in the ring. Au et al. [32] adapted traceable ring signature to the identify-based setting with constant signature size and enhanced privacy. Recently, Feng et al. [33] designed a logarithmic-size traceable ring signature scheme from lattices which proved to be secure in the quantum random oracle model.

1.2. Motivation and Contributions. In this section, to be more concrete, we summarize that the contribution of our paper is as follows:

(i) We present a novel revocable one-time ring signature scheme and define a perfect security model which provides the security properties: unforgeability, anonymity, linkability, non-slanderability, and revocability. And revocability of our scheme is mandatory

(ii) We show that our scheme is provable secure in the random oracle model under the assumptions that discrete logarithm (DL) problem, computational Diffie-Hellman (CDH) problem, and decisional bilinear Diffie-Hellman (DBDH) problem are intractable

(iii) We compare the efficiency of our scheme and previous schemes. Our scheme requires 4 times pairing computations which is independent of the size in the ring. Besides, the computational complexity in revocation part is more efficient than previous ones, and it only requires one scalar multiplication computation and one additional computation

2. Preliminaries

In this section, we introduce bilinear pairing and complex assumptions. They are utilized in the construction and provable security for our scheme. The notations used throughout the paper are described in Table 1.

2.1. Bilinear Pairing. Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be cyclic groups of a large prime order p. We write $\mathbb{G}_1$ additively and $\mathbb{G}_T$ multiplicatively. We assume that the discrete logarithm problems in $\mathbb{G}_1$ and $\mathbb{G}_T$ are intractable.

Let G be a bilinear group generator that, on input of a security parameter κ, outputs a description of bilinear groups $(\mathbb{G}_1,\mathbb{G}_T,e,P)$ such that $\mathbb{G}_1$ and $\mathbb{G}_T$ are cyclic groups of prime order p, P is a generator of $\mathbb{G}_1$, and a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$ satisfies the following properties:

(i) Bilinear: $\forall P \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$: $e(aP, bP) = e(P, P)^{ab}$

(ii) Non-degenerate: There exists $\forall P \in \mathbb{G}_1$, such that $e(P, P) \neq 1$

(iii) Computability: The map $e(P, P)$ is efficiently computability for any $P \in \mathbb{G}_1$

2.2. Complexity Assumptions

Definition 1 Discrete logarithm (DL) assumption. We say that the DL assumption holds if for any polynomial-time adversary $\mathscr{A}$, the following advantage ε^{DL} is negligible function in κ:

$$\varepsilon^{DL} \coloneqq \boldsymbol{Pr}\left[\mathscr{A}_{DL}(P, aP) = a : \begin{array}{c} (p, \mathbb{G}_1) \longleftarrow G(\kappa) ; \\ P \longleftarrow \mathbb{G}_1 ; \\ a \longleftarrow \mathbb{Z}_p^* ; \end{array}\right] = \operatorname{negl}(\kappa). \tag{1}$$

Table 1: The symbol description.

Symbol	Description
κ	A security parameter
p	A large prime number
$\mathbb{Z}_p^*$	The set consisting of positive integers less than p
$\mathbb{G}_1$,P	The additive group with p order and a generator
$\mathbb{G}_T$	The multiplicative group with prime number p order
e	A bilinear pairing, where $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$
pk_{revoke}	A revocation authority's public key, where $pk_{revoke} = \hat{y} = sk_{revoke}P = \hat{x}P$
H	A cryptographic hash function, where $H : \{0,1\}^* \longrightarrow \mathbb{Z}_p^*$
H_1	A deterministic hash function, where $H_1 : E(\mathbb{G}_1) \longrightarrow E(\mathbb{G}_1)$
H_2	A cryptographic hash function, where $H_2 : \mathbb{G}_1 \longrightarrow \mathbb{Z}_p^*$

Table 2: Comparison of ring signature schemes.

Scheme	Signature size	Sign	Verify	Assumption	Security model
Zhang et al. [35]	$n\lvert\mathbb{G}_1\rvert$	$(2n-1)\mathcal{S}m$	$(n+1)\mathcal{P}air$	q_s-CAA	ROM
Schäge et al. [36]	$(n+1)\lvert\mathbb{G}_1\rvert$	$(n+2)Sm + nadd$	$(n+2)Pair + nadd$	CDH	StanM
Liu et al. [25]	$(2n+3)\lvert\mathbf{G}_T\rvert$	$(3n+4)Sm + (4n+3)add$	$(2n+5)Pair + (3n+1)add$	CDH	StanM
roRS	$n\lvert G_1\rvert + n\left\lvert Z_p^*\right\rvert$	$(n+1)Sm + Pair$	$4Pair + nSm$	DL, DBD, CDH	ROM

Table 3: The notions in comparison.

Notion	Description
$\left\lvert\mathbb{Z}_p^*\right\rvert$	The length of the elements in $\mathbb{Z}_p^*$
$\lvert\mathbb{G}_1\rvert$	The size of the underlying group in $\mathbb{G}_1$
$\lvert\mathbb{G}_2\rvert$	The size of the underlying group in $\mathbb{G}_2$
$\lvert\mathbb{G}_T\rvert$	The size of the underlying group in $\mathbb{G}_T$
add	The time required for an addition computation
δm	The time required for a scalar multiplication computation
$\mathcal{P}air$	The time required for a pairing computation
ROM	The abbreviation for random Oracle model
StanM	The abbreviation for standard model

Table 4: Comparison in [29, 30] and roRS.

Scheme	Signature size	Revoke	Assumption	Linkability	Mandatory Revocability
Liu et al. [29]	$(2n+2)\left\lvert Z_p^*\right\rvert$	$nPair$	DBDH	✘	√
Fujisaki et al. [30]	$(2n+1)\left\lvert Z_p^*\right\rvert^{``}$	$2nadd + 2n\mathcal{S}m$	Dl, DDH	√	✘
roRS	$n\lvert G_1\rvert + n\left\lvert Z_p^*\right\rvert$	$\mathcal{S}m + add$	Dl, DBDH, CDH	√	√

Definition 2 Computational Diffie-Hellman (CDH) assumption. Let G be a group generator that, on input of a security parameter κ, outputs a cyclic group. We say that the CDH assumption holds if for any polynomial-time adversary $\mathscr{A}$, the following advantage ε^{CDH} is negligible function in κ:

$$\varepsilon^{CDH} \coloneqq \boldsymbol{Pr}\left[\mathscr{A}_{CDH}(P, aP, bP) = abP : \begin{array}{c} (p, \mathbb{G}_1) \longleftarrow G(\kappa) \,; \\ P \longleftarrow \mathbb{G}_1 \,; \\ a, b \longleftarrow \mathbb{Z}_p^* \,; \end{array}\right] = \mathrm{negl}(\kappa). \tag{2}$$

Definition 3 Decisional bilinear Diffie-Hellman (DBDH) assumption. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$ be a bilinear pairing, $P \in \mathbb{G}_1$. For $Z \in \mathbb{G}_T$, given the tuples (P, aP, bP, cP, Z), we say that the DBDH assumption holds if for any polynomial-time adversary $\mathscr{A}$, the following advantage ε^{DBDH} is negligible function in κ:

$$\begin{aligned} \varepsilon^{DBDH} \coloneqq & \left|\Pr\left[\mathscr{A}_{DBDH}\left(P, aP, bP, cP, e(P, P)^{abc}\right) = 1\right]\right| \\ & - \Pr[\mathscr{A}_{DBDH}(P, aP, bP, cP, Z) = 1] = \mathrm{negl}(\kappa). \end{aligned} \tag{3}$$

3. Security Model

In this section, we give the security model and the security notions of our revocable one-time ring signature.

3.1. Definition of Revocable One-Time Ring Signature

3.1.1. Revocable One-time Ring Signature. Revocable one-time ring signature (roRS) scheme is the tuples (Setup, KeyGen, Sign, Verify, Link, and Revoke).

(i) $pp \longleftarrow Setup(\kappa)$: The setup algorithm is a probabilistic polynomial time algorithm which takes as input a security parameter $\kappa \in N$ and outputs a set of public parameters pp

(ii) $(sk_i, pk_i) \longleftarrow KeyGen(pp)$: The key generation algorithm is a probabilistic polynomial time algorithm which takes as input public parameters pp and outputs a private/public key pair (sk_i, pk_i). Respectively, we denote SK and PK as the domain of possible private keys and possible public keys

(iii) $\sigma \longleftarrow \mathrm{Sign}(I, sk, Y, pk_{revoke}, M)$: The signing algorithm is a probabilistic polynomial time algorithm which takes as input a key image I, a private key sk, a message M, a revocation authority's public key $pk_{revoke} \in PK$, and a list Y of public keys in PK which includes the one corresponding to sk and produces a signature σ

(iv) $accept/reject \longleftarrow Verify(I, Y, M, pk_{revoke}, \sigma)$: The signature verification algorithm is a probabilistic polynomial time algorithm which takes as input the key image I, a set Y of public keys in PK, a revocation authority's public key $pk_{revoke} \in PK$, a message M, and a signature σ and returns accept or reject

(v) $linked/unlinked \longleftarrow Link(I_1, I_2, Y_1, Y_2, M_1, M_2, \sigma_1, \sigma_2)$: The linking algorithm which takes as input key images I_1, I_2, a set Y_1 of public keys in PK, and a set Y_2 of public keys in PK, messages M_1 and M_2, and signatures σ_1 and σ_2, such that $Verify(I_1, Y_1, M_1, \sigma_1) = accept$ and $Verify(I_2, Y_2, M_2, \sigma_2) = accept$ and returns linked or unlinked

(vi) $pk \longleftarrow Revoke(Y, \sigma, sk_{revoke})$: The revoking algorithm is a probabilistic polynomial time algorithm which takes as input a set Y of public keys in PK, a valid signature σ, and a secret key sk_{revoke} of the revocation authority and returns a public key pk in Y

3.1.2. Correctness. A revocable one-time ring signature scheme should satisfy the following:

(i) Verification correctness: A signature signed by honest signers is verified to be valid as follows:

$$\Pr\left[Verify(I, Y, M, pk_{revoke}, \sigma^*) = accept : \begin{array}{c} pp \longleftarrow Setup(\kappa) \,; \\ (sk_i, pk_i) \longleftarrow KeyGen(pp) \,; \\ \sigma \longleftarrow \mathrm{Sign}(I, sk, Y, pk_{revoke}, M) \,; \end{array}\right] = 1. \tag{4}$$

(ii) Linking correctness: Two signatures with the same event description generated by the same secret key of the identical signer must be linkable

(iii) Revocation Correctness: The revocation authority can reveal an honest signer's public key with overwhelming probability

3.2. Notions of Security. Security of our roRS scheme has five aspects: unforgeability, anonymity, linkability, non-slanderability, and revocability.

Formally, we capture attack behaviors as adversarial queries to oracles implemented by a challenger S. We provide adversary $\mathscr{A}$ the following oracles.

(i) JO (joining oracle). O_{join}: $pk_i \longleftarrow JO(\perp)$. $\mathscr{A}$ queries this oracle for adding a new user to the system. S keeps track of this type of queries by maintaining a list T_{join}, which is initially empty. Upon receiving a fresh query, S responds as below: picks random public parameters pp, runs $(sk_i, pk_i) \longleftarrow KeyGen(pp)$ to obtain (sk_i, pk_i). S records (sk_i, pk_i) in T_{join}, and then returns $pk_i (pk_i \in PK)$ to $\mathscr{A}$. This type of oracle captures $\mathscr{A}$ can observe the public keys of honest users in the system

(ii) CO (corruption oracle). $O_{corrupt}$: $sk_i \longleftarrow CO(pk_i)$. $\mathscr{A}$ queries this oracle with a public key $pk_i \in PK$ in

T_{join}. S keeps track of this type of queries by maintaining a list $T_{corrupt}$, which is initially empty. Upon receiving a fresh query, S records pk_i in $T_{corrupt}$. S returns the associated $sk_i \in SK$ to $\mathscr{A}$ and moves this entry to $T_{corrupt}$. This oracle captures $\mathscr{A}$ can corrupt some honest users and return private key sk_i

(iii) SO (signing oracle). O_{sign}: $\sigma' \longleftarrow SO(I, Y, pk_{revoke}, M)$. $\mathscr{A}$ queries this oracle with (I, Y, pk_{revoke}, M) (a key image I, a set Y of public keys, a revocation authority's public key $pk_{revoke} \in PK$, and a message M) and subjects to the restriction that $pk_s \in T_{join}(s = 1, \cdots, n)$. S keeps track of this type of queries by maintaining a list T_{sign}, which is initially empty. Upon receiving a fresh query, S responds as below: S runs $\sigma' \longleftarrow \text{Sign}(I, sk, pk_{revoke}, Y, M)$, records σ' on the list T_{sign}, and then sends σ' to $\mathscr{A}$. This oracle captures $\mathscr{A}$ can generate a signature itself

3.2.1. Unforgeability. We define unforgeability via the following security experiment $Expt_{\mathscr{A}}^{unforge}$ between $\mathscr{A}$ and S:

$$\mathscr{A}dv_{\mathscr{A}}^{unforge}(\kappa) = \Pr\left[\begin{array}{ccc} Verify(I, Y, pk_{revoke}, M, \sigma^*) = accept \wedge & & pp \longleftarrow Setup(\kappa)\,; \\ & : & \\ pk_i \in T_{join} \wedge \sigma^* \notin T_{\text{sign}} & & (I, Y, M, \sigma^*) \longleftarrow \mathscr{A}^{JO,CO,SO}(pp)\,; \end{array}\right]. \tag{5}$$

Here, $n \in \mathbb{N}$, $Y = \{pk_1, pk_2, \cdots, pk_n\} \in PK$, and I is a key image. pk_{revoke} is a revocation authority's public key, and all public keys are in T_{join}. No public keys are in $T_{corrupt}$. σ^* is not in T_{sign}. $\mathscr{A}dv_{\mathscr{A}}^{unforge}(\kappa)$ is the successful probability of adversary $\mathscr{A}$ who wins the unforgeability security experiment.

The process of unforgeability security experiment $Expt_{\mathscr{A}}^{unforge}$ is briefly described as follow:

(1) $\mathscr{A}$ runs $Setup(\kappa)$ with security parameter κ and sends the public parameter pp to $\mathscr{A}$

(2) $\mathscr{A}$ is allowed to make queries to O_{join}, $O_{corrupt}$, and O_{sign} according to any adaptive strategy

(3) $\mathscr{A}$ outputs a forgery signature σ^*

The conditions that $\mathscr{A}$ wins the experiment are as follows:

(1) All public keys are outputs of JO

(2) $Verify(I, Y, pk_{revoke}, M, \sigma^*)$=accept, and σ is not the output of SO

(3) No public keys have been queried to CO

Our roRS satisfies unforgeability if no PPT adversary has non-negligible advantage in the above experiment.

Definition 4 Unforgeability. A revocable one-time ring signature scheme is unforgeable, if for every probabilistic polynomial-time adversary $\mathscr{A}$, the advantage $\mathscr{A}dv_{\mathscr{A}}^{unforge}(\kappa)$ is negligible in κ.

3.2.2. Anonymity. We define anonymity via the following security experiment $Expt_{\mathscr{A}}^{anonymous}$ between $\mathscr{A}$ and S:

$$\mathscr{A}dv_{\mathscr{A}}^{anonymous}(\kappa) = \left| \Pr\left[s = s' : \begin{array}{c} pp \longleftarrow Setup(\kappa)\,; \\ (I, Y, M) \longleftarrow \mathscr{A}^{JO}(pp)\,; \\ s \overset{R}{\longleftarrow} \{1, \cdots, n\}\,; \\ \sigma_s \longleftarrow \text{Sign}(I, sk, Y, pk_{revoke}, M)\,; \\ s' \longleftarrow \mathscr{A}(\sigma_s)\,; \end{array} \right] - \frac{1}{n} \right|. \tag{6}$$

Here, $n \in \mathbb{N}$, $Y = \{pk_1, pk_2, \cdots, pk_n\} \in PK$, and I is a key image. pk_{revoke} is a revocation authority's public key, and pk_i is chosen by $\mathscr{A}$ in T_{join}. sk_s is a corresponding private key of pk_s. $\mathscr{A}dv_{\mathscr{A}}^{anonymous}(\kappa)$ is the successful probability of adversary $\mathscr{A}$ who wins the anonymity security experiment.

The process of anonymity security experiment $Expt_{\mathscr{A}}^{anonymous}$ is briefly described as follow:

(1) $\mathscr{A}$ runs $Setup(\kappa)$ with security parameter κ and sends the public parameter pp to $\mathscr{A}$

(2) $\mathscr{A}$ is allowed to make queries to O_{join} according to any adaptive strategy

(3) $\mathscr{A}$ gives S a set Y of public keys in PK such that all of the public keys in Y are query outputs of JO, a key image I and a message M. Parse the set Y as $\{pk_1, pk_2, \cdots, pk_n\}$. S randomly picks $s \in \{1, \cdots, n\}$ and computes $\sigma_s = \text{Sign}(I, Y, sk_s, M)$, where sk_s is a corresponding private key of pk_s. σ_s is given to $\mathscr{A}$

(4) $\mathscr{A}$ outputs a guess $s' \in \{1, \cdots, n\}$

So our roRS satisfies anonymity if no PPT adversary has non-negligible advantage in the above experiment.

Definition 5 Anonymity. A revocable one-time ring signature scheme is anonymous, if for every probabilistic polynomial-time adversary $\mathscr{A}$, the advantage $Adv^{anonymous}\mathscr{A}(\kappa)$ is negligible in κ.

3.2.3. Linkability. We define linkability via the following security experiment $Expt_{\mathscr{A}}^{linkable}$ between $\mathscr{A}$ and S:

$$\mathscr{A}dv_{\mathscr{A}}^{linkable}(\kappa) = \Pr\left[\begin{array}{c} Verify(I, Y, pk_{revoke}, M, \sigma^*) = accept\wedge \\ pk_i \in T_{join}\wedge\sigma_i \notin T_{\text{sign}}\wedge \\ Link(\sigma_1,\sigma_2) = unlinked, i = 1,2 \end{array} : \begin{array}{c} pp \longleftarrow Setup(\kappa)\,; \\ (I, Y, M, \sigma^*) \longleftarrow \mathscr{A}^{JO,CO,SO}(pp)\,; \\ \end{array}\right]. \tag{7}$$

Here, $n_i \in \mathbb{N}$, $i = 1, 2$; $Y_i = \{pk_1, pk_2, \cdots, pk_{n_i}\} \in PK$, a message M and signature σ_i, $i = 1, 2$; I is a key image, and pk_{revoke} is a revocation authority's public key. All pk_i chosen by $\mathscr{A}$ are in T_{join}. σ_i are not in T_{sign}, CO has been queried less than 2 times. $\mathscr{A}dv_{\mathscr{A}}^{linkable}(\kappa)$ is the successful probability of adversary $\mathscr{A}$ who wins the linkability security experiment.

The process of linkability security experiment $Expt_{\mathscr{A}}^{linkable}$ is briefly described as follows:

(1) $\mathscr{A}$ runs $Setup(\kappa)$ with security parameter κ and sends the public parameter pp to $\mathscr{A}$

(2) $\mathscr{A}$ is allowed to make queries to O_{join}, $O_{corrupt}$, and O_{sign} according to any adaptive strategy

(3) $\mathscr{A}$ outputs a forgery signature σ

The conditions that $\mathscr{A}$ wins the experiment are as follows:

(1) All public keys are outputs of JO

(2) $Verify(I, Y, pk_{revoke}, M, \sigma)$=accept, and σ is not the output of SO

(3) $\mathscr{A}$ can only at most queried 1 time and at most have one user's private key

(4) $Link(\sigma_1, \sigma_2)$= unlinked

So our roRS satisfies linkability if no PPT adversary has non-negligible advantage in the above experiment.

Definition 6 Linkability. A revocable one-time ring signature scheme is linkable, if for every probabilistic polynomial-time adversary $\mathscr{A}$, the advantage $\mathscr{A}dv_{\mathscr{A}}^{linkable}(\kappa)$ is negligible in κ.

3.2.4. Non-slanderability. We define non-slanderability via the following security experiment $Expt_{\mathscr{A}}^{\text{non}-slanderous}$ between $\mathscr{A}$ and S:

$$\mathscr{A}dv_{\mathscr{A}}^{\text{non}-slanderous}(\kappa) = \Pr\left[\begin{array}{c} Verify(I, Y^*, pk_{revoke}, M^*, \sigma^*) = accept\wedge \\ \sigma^* \neq \sigma'\wedge\sigma^* \notin T_{\text{sign}}\wedge \\ pk_s \notin T_{corrupt}\wedge pk_s \notin T_{\text{sign}}\wedge \\ Link\left(\sigma^*, \sigma'\right) = linked \end{array} : \begin{array}{c} pp \longleftarrow Setup(\kappa)\,; \\ (I, Y, M, pk_s) \longleftarrow \mathscr{A}^{JO,CO,SO}(pp)\,; \\ \sigma' \longleftarrow \text{Sign}(sk_s, Y, pk_{revoke}, M)\,; \\ (I, Y^*, M^*, \sigma^*) \longleftarrow \mathscr{A}^{JO,CO,SO}\left(\sigma'\right)\,; \end{array}\right]. \tag{8}$$

Here, $Y, Y^* \in PK$, message M, M^*, and signature σ', σ^*; pk_{revoke} is a revocation authority's public key, and I is a key image. $\mathscr{A}$ is subject to the restriction that pk_s chosen by $\mathscr{A}$ is not allowed to be either in $T_{corrupt}$ or T_{sign}. All of the public keys in Y^* and Y are in T_{join}. $\sigma^* \neq \sigma'$ and σ^* are not in T_{sign}. $\mathscr{A}dv_{\mathscr{A}}^{\text{non}-slanderous}(\kappa)$ is the successful probability of adversary $\mathscr{A}$ who wins the non-slanderability security experiment.

The process of non-slanderability security experiment $Expt_{\mathscr{A}}^{\text{non}-slanderous}$ is briefly described as follows:

(1) $\mathscr{A}$ runs $Setup(\kappa)$ with security parameter κ and sends the public parameter pp to $\mathscr{A}$

(2) $\mathscr{A}$ is allowed to make queries to O_{join}, $O_{corrupt}$, and $O_{\boldsymbol{sign}}$ according to any adaptive strategy

(3) $\mathscr{A}$ outputs a forgery signature σ^*

The conditions that $\mathscr{A}$ wins the experiment are as follows:

(1) $Verify(I, Y^*, pk_{revoke}, M^*, \sigma^*)$=accept

(2) σ^* is not an output of SO

(3) pk_s has not been queried to CO

(4) All public keys are in Y^*; Y are query outputs of JO

(5) $Link(\sigma', \sigma^*)$=linked

So our roRS satisfies non-slanderability if no PPT adversary has non-negligible advantage in the above experiment.

Definition 7 Non-slanderability. A revocable one-time ring signature scheme is non-slanderous, if for every probabilistic polynomial-time adversary $\mathscr{A}$, the advantage $\mathscr{A}dv_{\mathscr{A}}^{\text{non}-slanderous}(\kappa)$ is negligible in κ.

3.2.5. Revocability. We define revocability via the following security experiment $Expt_{\mathscr{A}}^{revocable}$ between $\mathscr{A}$ and S:

$$\mathscr{A}dv_{\mathscr{A}}^{revocable}(\kappa) = \Pr\left[\begin{array}{ccc} Verify(I, Y, pk_{revoke}, M, \sigma) = accept\wedge & & pp \longleftarrow Setup(\kappa)\,; \\ pk_i \in T_{join}\wedge\sigma \notin T_{\boldsymbol{sign}}\wedge & : & sk_s \longleftarrow \mathscr{A}^{CO}(pp)\,; \\ pk_j = Revoke(Y, \sigma, sk_{revoke}), j \neq s & & (I, Y, M, \sigma) \longleftarrow \mathscr{A}^{JO,CO,SO}(pp)\,; \end{array}\right]. \tag{9}$$

Here, $n \in \mathbb{N}$, $Y = \{pk_1, pk_2,\cdots,pk_n\} \in PK$, signature σ, and pk_{revoke} is a revocation authority's public key, and I is a key image. All pk_i chosen by $\mathscr{A}$ are in T_{join}. σ are not in $T_{\boldsymbol{sign}}$, CO has been queried less than 2 times, that is, $\mathscr{A}$ can only obtain at most one private key denotes as sk_s, and pk_{revoke} is the corresponding public key of sk_{revoke}. $\mathscr{A}dv^{revocable}\mathscr{A}(\kappa)$ is the successful probability of adversary $\mathscr{A}$ who wins the revocability security experiment.

The process of revocability security experiment $Expt_{\mathscr{A}}^{revocable}$ is briefly described as follows:

(1) $\mathscr{A}$ runs $Setup(\kappa)$ with security parameter κ and sends the public parameter pp to $\mathscr{A}$

(2) $\mathscr{A}$ is allowed to make queries to O_{join}, $O_{corrupt}$, and $O_{\boldsymbol{sign}}$ according to any adaptive strategy

(3) $\mathscr{A}$ outputs a forgery signature σ

The conditions that $\mathscr{A}$ wins the experiment are as follows:

(1) $Verify(I, Y, pk_{revoke}, M, \sigma)$=accept

(2) σ^* is not an output of SO

(3) All public keys are query outputs of JO

(4) CO has been queried less than 2 times, that is, $\mathscr{A}$ can only obtain at most one private key denotes as sk_s

(5) $y_j = Revoke(Y, \sigma, sk_{revoke})$ for $j \neq s$

So our roRS satisfies revocability if no PPT adversary has non-negligible advantage in the above experiment.

Definition 8 Revocability. A revocable one-time ring signature scheme is revocable, if for every probabilistic polynomial-time adversary $\mathscr{A}$, the advantage $\mathscr{A}dv_{\mathscr{A}}^{revocable}(\kappa)$ is negligible in κ.

4. Construction

In this section, we propose our new revocable one-time ring signature (roRS) scheme. Our scheme is constructed as follows:

Setup: Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be two groups with prime order $p > 2^\kappa$; $\mathbb{G}_1$ is an additive group and $\mathbb{G}_T$ is a multiplicative group. P is the generator of $\mathbb{G}_1$ and a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$. Let $H : \{0, 1\}^* \longrightarrow \mathbb{Z}_p^*$ and $H_2 : \mathbb{G}_1 \longrightarrow \mathbb{Z}_p^*$ be two cryptographic hash functions and $H_1 : E(\mathbb{G}_1) \longrightarrow E(\mathbb{G}_1)$ be a deterministic hash function. And the public parameters $params = (\mathbb{G}_1, \mathbb{G}_T, e, P, H, H_1, H_2, p)$..

KeyGen: A ring user i randomly picks $x_i \in \mathbb{Z}_p^*$ and computes $pk_i = x_iP \in \mathbb{G}_1$. So the user i has secret key and public key pair $(sk_i, pk_i) = (x_i, x_iP)$. The secret key sk_{revoke} of the revocation authority is $\hat{x}$, and the corresponding public key pk_{revoke} is $\hat{y} = \hat{x}P$.

Sign: Let $Y = \{pk_1, pk_2, \cdots, pk_n\}$ be a set of users' public keys in the ring. So a ring user $s(1 \leq s \leq n)$ has his own key pair $(sk_s, pk_s) = (x_s, x_sP)$. Additionally, the user has given a message $M \in \{0, 1\}^*$; then, the user s with the knowledge of x_s computes a signature of knowledge as follows:

(1) Compute the key image I: First, use a hash function with pk_s to make one signer only have the corresponding one key image.

$K = H_1(pk_s)$; then, compute the key image I with the signer's private key x_s and K,$I = x_sK$.

(2) Randomly choose $\omega \in \mathbb{Z}_p^*$, and compute: First make proof of knowledge for private key x_s, $B_1 = \omega P$ and then construct B_2 for the revoking phase and B_3 for the verification phase:

$$\begin{aligned} B_2 &= \omega\hat{y} + pk_s, \\ B_3 &= e(\omega pk_s, K). \end{aligned} \tag{10}$$

(3) Randomly choose $c_i \in \mathbb{Z}_p^*$, where $i = \{1, 2, \cdots, n\}$; and randomly choose $z_i \in \mathbb{G}_1$, where $i = \{1, 2, \cdots, s-1, s+1, \cdots, n\}(i \neq s)$. Then, set the following transformations for all users in the ring:

$$C_i = \begin{cases} c_iP, & if\, i = s \\ c_iP + pk_sH_2(z_i), & if\, i \neq s \end{cases} \text{ and } Z_i = \begin{cases} c_iK, & if\, i = s \\ c_iK + IH_2(z_i), & if\, i \neq s \end{cases}. \tag{11}$$

(4) Get the non-interactive challenge:

$$z = H(M, C_1, \cdots, C_n, Z_1, \cdots, Z_n). \tag{12}$$

(5) Randomly choose $\Phi_i \in \mathbb{G}_1$, where $i = \{1, 2, \cdots, s-1, s+1, \cdots, n\}$ and $i \neq s$. Then when $i \neq s$, make a hash function h_i to be a random number for verifiers. Compute $h_i = H(\Phi_i, Y, I, M, B_3)$, where $i \neq s$

(6) Then, when $i = s$, use notions above and compute:

Compute Φ_s as a random number:

$$\Phi_s = zP - \sum_{i=1, i\neq s}^{n} (h_ipk_i + \Phi_i) \in \mathbb{G}_1, \tag{13}$$

then make a hash function h_s as a random number for $i = s$:

$$h_s = H(\Phi_s, Y, I, M, B_3) \in \mathbb{Z}_p^*, \tag{14}$$

use private key x_s to make the response V when $i = s$:

$$V = (z + h_sx_s) \cdot B_1 \in \mathbb{G}_1. \tag{15}$$

(7) To close the ring, set $l_i = \begin{cases} c_s - x_sH_2(\Phi_s), & if\, i = s \\ c_i, & if\, i \neq s \end{cases}$

(8) Output the signature $\sigma = (l_1, \cdots, l_n, \Phi_1, \cdots, \Phi_n, V, I, B_1, B_2)$

Verify: Given the tuples (σ, Y, M), the verifier carries out the following steps:

(1) Compute $C_i' = l_iP + pk_sH_2(\Phi_i)$ and $Z_i' = l_iH_1(pk_s) + IH_2(\Phi_i)$, $B_3 = e(B_1, I)$, and $h_i = H(\Phi_i, Y, I, M, B_3) \in \mathbb{Z}_p^*$, where $i = \{1, 2, \cdots, n\}$

(2) Check whether $\sum_{i=1}^{n}(h_ipk_i + \Phi_i) = H(M, C_1', \cdots, C_n', Z_1', \cdots, Z_n') \cdot P$ and $e(P, V) = e(\sum_{i=1}^{n}[h_ipk_i + \Phi_i], B_1)$. If two equalities hold, accept the signature; otherwise, reject it

Link: On receive the tuples:

$$\begin{aligned}&(\sigma_1 = (\cdot, I_1), Y_1, M_1),\\ &(\sigma_2 = (\cdot, I_2), Y_2, M_2).\end{aligned} \tag{16}$$

The verifier checks if both σ_1 and σ_2 are two valid signatures. If yes, then outputs link if $I_1 = I_2$. Otherwise, reject.

Revoke: On receive the tuples (Y, σ, sk_{revoke}). The revocation authority first checks whether σ is a valid signature. If it holds, continue; otherwise, reject. In order to revoke the anonymity of the actual signer, the revocation authority makes as follows:

If there exists $y_s \in Y$, such that $y_s = B_2 - \hat{x}B_1$, where y_s is the public key of the actual signer.

5. Correctness Analysis

5.1. Verification Correctness. On correctness, using the bilinearity and nondegeneracy of the pairing e, a signature is correctly verified by the Verify algorithm as follows:

$$\begin{gathered} B_3 = e(B_1, I) = e(\omega P, x_sK) = e(\omega x_sP, K) = e(\omega pk_s, K), \\ e\left(\sum_{i=1}^{n}[h_i \cdot pk_i + \Phi_i], B_1\right) = e\left(\sum_{i=1, i\neq s}^{n}[h_i \cdot pk_i + \Phi_i] + h_s \cdot pk_s + \Phi_s, B_1\right) = e(z \cdot P + h_s \cdot pk_s, B_1) = e(P, V). \end{gathered} \tag{17}$$

5.1.1. Linking Correctness. On linking correctness, the signer computes the key image as follows:

$$I = x_iH_1(pk_i). \tag{18}$$

Therefore, the user can only compute the key image once with the same event description.

5.1.2. Revoking Correctness. On revoking correctness, the revocation authority can successfully identify the actual signer's public key as follows:

$$y_s = B_2 - \hat{x}B_1 = \omega\hat{y} + pk_s - \hat{x}\omega P = pk_s, \tag{19}$$

where $\hat{x}$ is the revocation authority's private key and y_s is the actual signer's public key.

6. Security Analysis

In this section, the security proofs of the proposed scheme are given.

Theorem 9 Unforgeability. *Our proposed scheme is unforgeable in the random oracle model with the CDH assumption.*

Proof. Suppose there exists a PPT adversary $\mathscr{A}$ with advantage ε, which means that $\mathscr{A}$ can forge a valid signature with probability ε. Then, we use a simulator S to solve the CDH problem. Let (P, aP, bP) be a given instance, where a, b is randomly picked in $\mathbb{Z}_p^*$ and $P \in \mathbb{G}_1$. Through the adversary $\mathscr{A}$, we use the simulator S which outputs the CDH solution abP. S randomly selects $j \in \{1, \cdots, n\}$ as the actual signer in the simulation. So the simulator S simulates the oracles by interacting with the adversary $\mathscr{A}$ in the nature way as follows:

(1) Setup: S sets $pk_j = aP$ and $B_1 = bP$. S randomly chooses $u \in \mathbb{Z}_p^*$ and sets $\hat{y} = uP$. $\mathscr{A}$ is given the public parameters $(\mathbb{G}_1, \mathbb{G}_T, p, e, P, H, H_1, H_2)$, and the public keys $Y = \{pk_1, pk_2, \cdots, pk_n\}$

(2) Queries: $\mathscr{A}$ queries the oracles H, H_1, H_2, JO, CO, and SO in this phase, and S sets several lists to record the queries and answers. These lists are initially empty

(i) H-queries: S maintains and checks a corresponding list L_H as $\mathscr{A}$ queries hash values. If an entry for the query is found in L_H, S returns the same answer to $\mathscr{A}$. Otherwise, S generates a random value as an answer to $\mathscr{A}$, and then the query and the answer are added in L_H

(ii) H_1-queries: S maintains a list L_{H_1}. When $\mathscr{A}$ issues a query $H_1(\alpha_i)$, S randomly chooses $f_1 \in \mathbb{G}_1$ and sets $H_1(\alpha_i) = f_1$ as the answer. The query and the answer then are stored in list L_{H_1}

(iii) H_2-queries: S maintains a list L_{H_2}. When $\mathscr{A}$ issues a query $H_1(\beta_i)$, S randomly chooses $f_2 \in \mathbb{Z}_p^*$ and sets $H_2(\beta_i) = f_2$ as the answer. The query and the answer then are stored in list L_{H_2}

(iv) JO-queries: When $\mathscr{A}$ queries JO at the jth query, S returns the corresponding public key $pk_j = aP$ to $\mathscr{A}$. When $\mathscr{A}$ queries JO at the ith query($i \neq j$), S randomly chooses $d_i \in \mathbb{Z}_p^*$ and returns d_iP as the corresponding public key. The query and the answer then are stored in list L_{JO}

(v) CO-queries: When $\mathscr{A}$ queries CO with the public key pk_i which is an output of JO, S first checks if $i = j$. If yes, S fails and stops. Otherwise, S returns the corresponding d_i as the corresponding secret key. The query and the answer then are stored in list L_{CO}

(vi) SO-queries: When $\mathscr{A}$ queries SO with a tuple $(event, M, Y, pk_j)$. If $pk_i \neq pk_j$, S outputs a signature σ by Sign algorithm. Otherwise, S maintains a list L_{sign} with a tuple $(event, pk_j, I)$ and performs as follows:

(1) S retrieves the list $L_{\boldsymbol{sign}}$, if (pk_j, I) is found, and then takes the value I. If not, randomly selects a new value $I \in \mathbb{G}_1$ and adds (pk_j, I) to the list L_{sign}

(2) S computes $B_1 = bP, B_2 = b(uP) + aP, B_3 = e(B_1, I)$

(3) S chooses a random $j \in \{1, \cdots, n\}$

(4) S chooses $l_i \in \mathbb{Z}_p^*$ and $\Phi_i \in_R \mathbb{G}_1$ randomly, where

$i = \{1, 2, \cdots, j-1, j+1, \cdots, n\}$ and $i \neq j$; and S computes $C_i = l_iP + f_2(aP)$ and $Z_i = l_if_1 + f_2I$ and then selects $h_i \in \mathbb{Z}_p^*$ in L_H such that $H(\Phi_i, M, Y, I, B_3) = h_i$

(5) S randomly picks $z, h_j \in \mathbb{Z}_p^*$ and computes

$$\Phi_j = z \cdot P - h_j \cdot pk_j + \left[\sum_{i=1, i\neq j}^{n} h_i pk_i + \Phi_i\right], \tag{20}$$

and stores the value $h_j = H(\Phi_j, M, Y, I, B_3)$ to L_H. If there has a collision with hash values in L_H, do Step (5) again until no collision happen

(6) S computes $V = zB_1$ and then outputs $\sigma = (l_1, \cdots, l_n, \Phi_1, \cdots, \Phi_n, V, I, B_1, B_2)$ as a response to $\mathscr{A}$

Since each response is independently and uniformly distributed, all responses in the simulation are as in the real attack. Besides, all responses to SO are valid, and the output $\sigma = (l_1, \cdots, l_n, \Phi_1, \cdots, \Phi_n, V, I, B_1, B_2)$ in signing query can be verified with the Verify algorithm. Therefore, from $\mathscr{A}$'s view, the simulation is indistinguishable from the real attack. Now, $\mathscr{A}$ has the tuple $(\sigma = (l_1, \cdots, l_n, \Phi_1, \cdots, \Phi_n, V, I, B_1, B_2), h_1, \cdots, h_n)$, then by using Forking Lemma for ring signature [34], S rewinds the same random tape to let $\mathscr{A}$ obtain another tuple $(\sigma' = (l_1, \cdots, l_n, \Phi_1, \cdots, \Phi_n, V', I, B_1, B_2), h_1', \cdots, h_n')$ such that $h_j \neq h_j'$ for all $i \in \{1, \cdots, n\}$ and $i \neq j$. Then, there exists $V - V' = (h_j - h_j') \cdot a \cdot B_1 = (h_j - h_j') \cdot a \cdot (bP)$, that is,

$$\frac{V - V'}{\left(h_j - h_j'\right)} = abP. \tag{21}$$

Therefore, S can compute abP as a solution to solve CDH problem. Due to the Forking Lemma, the probability of successful rewind simulation is at least $\varepsilon/4$. Then, S can solve the CDH problem with probability $\varepsilon/4$ at least. □

Theorem 10 Anonymity. *Our proposed scheme is anonymous in the random oracle model with the DBDH assumption.*

Proof. Suppose there exists a PPT adversary $\mathscr{A}$ with advantage ε. Then, we use a simulator S to solve the DBDH problem. Let (aP, bP, cP, Z) be a given instance, where a, b, c is random picked in Z_p^*, $Z \in \mathbb{G}_T$, and $P \in \mathbb{G}_1$. Through the adversary $\mathscr{A}$, the simulator S's objective is to determine whether $Z = e(P, P)^{abc}$.

So the simulator S simulates the oracles by interacting with the adversary $\mathscr{A}$ as follows:

(1) Setup: The challenge signature is created using the randomly picked public key in Y. S randomly chooses $u \in \mathbb{Z}_p^*$ and sets $\hat{y} = uP$. In addition, S sets $pk_j = aP$ and $K = bP$. $\mathscr{A}$ is given the system parameters P and the public keys $Y = \{pk_1, pk_2, \cdots, pk_n\}$ of signers

(2) Queries: $\mathscr{A}$ does the same queries with the oracles (H, H_1, H_2, JO and CO) as Theorem 9

(3) Then, $\mathscr{A}$ queries SO, and S performs the steps as follows:

(1) S retrieves the list $L_{\boldsymbol{sign}}$, if (pk_j, I) is found, and then takes the value I. If not, randomly selects a new value $I \in \mathbb{G}_1$ and adds (pk_j, I) to the list $L_{\boldsymbol{sign}}$

(2) S sets $B_1 = cP$ and $B_3 = Z \in \mathbb{G}_T$ and computes $B_2 = c(uP) + bP$

(3) S chooses a random $j \in \{1, \cdots, n\}$

(4) S chooses $l_i \in \mathbb{Z}_p^*$ and $\Phi_i \in_R \mathbb{G}_1$ randomly, where $i = \{1, 2, \cdots, j-1, j+1, \cdots, n\}$ and $i \neq j$; and S computes $C_i = l_i P + f_2(aP)$ and $Z_i = l_i f_1 + f_2 I$ and then selects $h_i \in \mathbb{Z}_p^*$ in L_H such that

$$H(\Phi_i, M, Y, I, Z) = h_i \tag{22}$$

(5) S randomly picks $z, h_j \in \mathbb{Z}_p^*$ and computes

$$\Phi_j = z \cdot P - h_j \cdot pk_j + \left[\sum_{i=1, i \neq j}^{n} h_i pk_i + \Phi_i \right], \tag{23}$$

and stores the value $h_j = H(\Phi_j, M, Y, I, Z)$ to L_H. If there has a collision with hash values in L_H, do this step again until no collision happens.

(6) S computes $V = zB_1$ and then outputs$\sigma = (l_1, \cdots, l_n, \Phi_1, \cdots, \Phi_n, V, I, B_1, B_2)$ as a response to $\mathscr{A}$

S gives σ to $\mathscr{A}$, and $\mathscr{A}$ can query the random oracles adaptively and returns a bit $\eta \in \{0, 1\}$. Suppose $\mathscr{A}$ guesses that the signer's index is $j \in [1, n]$. If $\mathscr{A}$ cannot identify a signer, S returns 0. If $j = s$, it returns 1; if $j = 0$, it returns 0; otherwise, it returns 1/0 with equal probability. If $Z = e(P, P)^{bcd}$, then $B_3 = e(\omega pk_j, K) = e(c(aP), bP) = e(P, P)^{abc} = Z$. And when S returns 0, from $\mathscr{A}$'s view, all signers has equal probability to sign the signature. Suppose $\mathscr{A}$ has advantage ε in the simulation, then we set $\Pr[Z = e(P, P)^{abc}] = 1/2 + \varepsilon$. The probability of the right choice is computed as

$$\boldsymbol{Pr}\left[Z = e(P, P)^{bcd}\right] \geq \frac{1}{2}\left(\boldsymbol{Pr}\left[Z = e(P, P)^{bcd} | S \longleftarrow 1\right] + \boldsymbol{Pr}\left[Z = e(P, P)^{bcd} | S \longleftarrow 0\right]\right) \geq \frac{1}{2}\left[\left(\frac{1}{2} + \frac{1}{2n} + \frac{\varepsilon}{2}\right) + \left(\frac{1}{2} - \frac{1}{2n}\right)\right] = \frac{1}{2} + \frac{\varepsilon}{4}. \tag{24}$$

Therefore, S can determine whether $Z = e(P, P)^{abc}$ with the probability than 1/2 if $\mathscr{A}$ can win, contradiction occurs. □

Theorem 11 Linkability. *Our proposed scheme is linkable in the random oracle model with the discrete logarithm assumption.*

Proof. In order to prove linkability of our roRS scheme, we perform the same setting of oracle queries as the proof in Theorem 9, and we allow S to give $\mathscr{A}$ the public parameters, where S has at most one private key sk_s, and this private key can correspond to two different keys in ring group Y for $i = 1, 2$. (When $\mathscr{A}$ queries the CO, $\mathscr{A}$ can only get one private key. $\mathscr{A}$ can be allowed to get only one private key.)

So through the queries, $\mathscr{A}$ can output two valid signatures:

$$\begin{aligned} \sigma^{(1)} &= \left(\cdot, \Phi_1^{(1)}, \cdots, \Phi_n^{(1)}, V^{(1)}, I^{(1)}\right), \\ \sigma^{(2)} &= \left(\cdot, \Phi_1^{(2)}, \cdots, \Phi_n^{(2)}, V^{(2)}, I^{(2)}\right), \end{aligned} \tag{25}$$

and the key image of $\sigma^{(1)}$ is $I_s^{(1)} = x_s H_1(pk_s)$, and the key image of $\sigma^{(2)}$ is $I_s^{(2)} = x'_s H_1(pk_s)$.

For $\sigma^{(1)}$, S rewinds the same tape with a different value to obtain another valid signature $\tilde{\sigma}^{(1)}$. Then, we obtain

$$\tilde{\sigma}^{(1)} = \left(\cdot, \tilde{\Phi}_1^{(1)}, \cdots, \tilde{\Phi}_n^{(1)}, \tilde{V}^{(1)}, I^{(1)}\right). \tag{26}$$

If $\varPhi_s^{(1)} = \tilde{\varPhi}_s^{(1)}$, abort. If $\varPhi_s^{(1)} \neq \tilde{\varPhi}_s^{(1)}$, we have

$$\begin{aligned} \tilde{V}^{(1)} - V^{(1)} &= \left(z + \tilde{h}_s^{(1)} x_s\right)P - \left(z + h_s^{(1)} x_s\right)P, \\ x_s &= \log_P\left(\frac{\tilde{V}^{(1)} - V^{(1)}}{\tilde{h}_s^{(1)} - h_s^{(1)}}\right), \end{aligned} \tag{27}$$

where $I^{(1)} = x_s H_1(pk_s)$ and $y_s = x_s P = pk_s$.

For $\sigma^{(2)}$, S rewinds the same tape with a different value to obtain another valid signature $\tilde{\sigma}^{(2)}$. Then, we obtain

$$\tilde{\sigma}^{(2)} = \left(\cdot\,,\tilde{\varPhi}_1^{(2)}, \cdots, \tilde{\varPhi}_n^{(2)}, \tilde{V}^{(2)}, I^{(2)}, B_1, B_2\right). \tag{28}$$

If $\varPhi_s^{(2)} = \tilde{\varPhi}_s^{(2)}$, abort. If $\varPhi_s^{(2)} \neq \tilde{\varPhi}_s^{(2)}$, we have

$$\begin{aligned} \tilde{V}^{(2)} - V^{(2)} &= \left(z + \tilde{h}_s^{(2)} x_s'\right)P - \left(z + h_s^{(2)} x_s'\right)P, \\ x_s' &= \log_P\left(\frac{\tilde{V}^{(2)} - V^{(2)}}{\tilde{h}_s^{(2)} - h_s^{(2)}}\right). \end{aligned} \tag{29}$$

Therefore, $x_s = x_s'$ and $I^{(1)} = I^{(2)}$. Two signatures $(\sigma^{(1)}, \sigma^{(2)})$ are linked. S can break DLP, and the advantage of $\mathscr{A}$ is negligible since the rewind simulation is successful. □

Theorem 12 Non-slanderability. *Our proposed scheme is non-slanderable in the random oracle model with the discrete logarithm assumption.*

Proof. In order to prove non-slanderability of our roRS scheme, we perform the same setting of oracle queries as the proof in Theorem 9; the adversary $\mathscr{A}$ can query CO to get any public key, but $\mathscr{A}$ cannot be allowed to query the signer's public key pk_s. But $\mathscr{A}$ can give simulator S the tuples $(pk_s, Y, M, pk_{revoke})$. S uses the tuple to generate a valid signature $\sigma = (\cdot\,, I)$ which I is the key image computed using x_s. ($\mathscr{A}$ can keep querying oracles with the restriction of submitting pk_s to CO.)

Suppose $\mathscr{A}$ generates another valid signature $\sigma'' = (\cdot\,, I'')$ which I'' is computed by x_s'' and σ'' is not an output of SO. Additionally, σ and σ'' are linked. Therefore, $I = I''$ which means $I = x_s H_1(pk_s) = I'' = x_s'' H_1(pk_s)$. So $x_s = x_s''$, it can imply that $\mathscr{A}$ knows the secret key x_s corresponding to pk_s. This is opposite to our assumption that $\mathscr{A}$ cannot query C O to get the secret key of pk_s. □

Theorem 13 Revocability. *Our proposed scheme is revocable in the random oracle model if the construction is unforgeable.*

Proof. In order to prove revocability of our roRS scheme, we perform the same setting of oracle queries as the proof in Theorem 9, and we allow S give $\mathscr{A}$ the public parameters, where $\mathscr{A}$ can get one private key denoted as $sk_s = x_s$ corresponding to $pk_s = y_s$ in Y from CO. Due to the fact that $\{pk_1, \cdots, pk_{s-1}, pk_{s+1}, \cdots, pk_n\}$ is randomly and independently distributed, $\mathscr{A}$ cannot tell out the corresponding secret keys according to our assumption. Then, we suppose that $\mathscr{A}$ can generate a valid signature $\sigma = (l_1, \cdots, l_n, \varPhi_1, \cdots, \varPhi_n, V, I, B_1, B_2)$ successfully for contradiction. The valid signature must be generated by $sk_s = x_s$ because our proposed scheme is unforgeable. There exists a case that can break revocability in our scheme. We consider one case that can break revocability of our scheme. $\mathscr{A}$ randomly selects $j \in \{1, \cdots, s-1, s+1, \cdots, n\} (j \neq s)$ and performs the Sign algorithm; but on behalf of closing the ring, $\mathscr{A}$ must know the secret key x_j $(x_j \neq x_s)$. Due to our assumption that $\mathscr{A}$ can only get one private key, contradiction occurs. □

7. Efficiency Analysis

In this section, performance analysis is shown in Table 2. The efficiency computational cost and signature size between our proposed scheme and several signature schemes such as revocable ring signature [29], traceable ring signature [30] and so on. Then, we have several computational notions need to define as follows in Table 3.

We do the comparison about computation cost on the size of signature schemes and the timings of signing and verifying with [25, 29, 30, 35, 36], where n is the number users included in the group.

From Table 2, we compare our scheme with [25, 35, 36] in signature size, signing, verifying, assumption and security model. We can see that roRS and [25, 35, 36] have the same level of computational complexity in terms of signature size and signature signing. As for verifying, we only need the constant pairing computations which show more efficient than [25, 35, 36] constructed from bilinear pairings similarly. And our security model is built in the random oracle model that is different from [25, 36] built in the standard model.

In Table 4, in relation to the revocation phase, we only need one scalar multiplication computation and one addition computation compared with n pairing computations in [29] and $2n$ addition computations and $2n$ scalar multiplication computations in [30]. As for the functionality of roRS and [29, 30], [29] cannot achieve the linkability, and [30] cannot provide mandatory revocability; nevertheless, we have accomplished the linkability and the mandatory revocation simultaneously in the random oracle.

8. Conclusion

In this literature, we proposed a novel revocable one-time ring signature scheme based on bilinear pairings, which is provable secure under the DL, DBDH, and CDH assumptions in the random oracle model. In our scheme, we have simultaneously introduced linkability and mandatory revocability to distinguish from other ring signature schemes. In particular, linkability can prevent the double-signing attack, and mandatory revocability guarantees that a revocation authority can identify the actual signer when the actual signer commits a crime in transactions. The scheme about revocation phase requires only one scalar multiplication computation and one additional computation, and the pairing computations in the timings of verifying phase is constant. And constructing provable security revocable one-time ring signature schemes from lattices to resist quantum attackers is an interesting problem that we leave open for further research.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (Grant no. U21A20463). We thank Zhang et al. [37] who have provided the inspiration for our scheme.

References

[1] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology-ASIACRYPT 2001*, C. Boyd, Ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[2] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 157–165, 2014.

[3] P. P. Tsang and V. K. Wei, "Short Linkable Ring Signatures for E-Voting, E-Cash and Attestation," in *Information Security Practice and Experience*, R. H. Deng, F. Bao, H. H. Pang, and J. Zhou, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[4] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup, "Anonymous Identification in Ad Hoc Groups," in *Advances in Cryptology-EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., pp. 609–626, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[5] K. Kajita, K. Ogawa, and E. Fujisaki, "A constant-size signature scheme with a tighter reduction from the cdh assumption," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 1, pp. 141–149, 2020.

[6] W. Wang, J. Song, G. Xu, Y. Li, H. Wang, and C. Su, "Contractward: automated vulnerability detection models for ethereum smart contracts," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1133–1144, 2021.

[7] S. Noether, A. Mackenzie, and T. M. Research Lab, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, 2016.

[8] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n signatures from a variety of keys," in *Advances in Cryptology-ASIACRYPT 2002*, YC. Zheng, Ed., pp. 415–432, 2002.

[9] A. Bender, J. Katz, and R. Morselli, "Ring signatures: stronger definitions, and constructions without random oracles," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds., pp. 60–79, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[10] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret: theory and applications of ring signatures," in *Theoretical Computer Science*, pp. 164–186, Springer, 2006.

[11] H. Shacham and B. Waters, "Efficient ring signatures without random oracles," in *International Workshop on Public Key Cryptography*, pp. 166–180, Springer, 2007.

[12] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in Cryptology — EUROCRYPT 2003*, E. Biham, Ed., pp. 416–432, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[13] P.-L. Cayrel, R. Lindner, M. Rückert, and R. Silva, "A lattice-based threshold ring signature scheme," in *Progress in Cryptology-LATINCRYPT 2010*, M. Abdalla and P. S. L. M. Barreto, Eds., pp. 255–272, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[14] J. K. Liu, T. H. Yuen, and J. Zhou, "Forward secure ring signature without random oracles," in *Information and Communications Security*, S. Qing, W. Susilo, G. Wang, and D. Liu, Eds., pp. 1–14, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[15] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in *Advances in Cryptology-CRYPTO '94*, Y. G. Desmedt, Ed., pp. 174–187, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[16] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Advances in Cryptology-CRYPTO 2002*, M. Yung, Ed., pp. 61–76, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[17] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Constant-size id-based linkable and revocable-iff-linked ring signature," in *Progress in Cryptology-INDOCRYPT 2006*, R. Barua and T. Lange, Eds., pp. 364–378, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[18] I. R. Jeong, J. O. Kwon, and D. H. Lee, "Ring signature with weak linkability and its applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1145–1148, 2008.

[19] J. K. Liu and D. S. Wong, "Linkable ring signatures: security models and new schemes," in *Computational Science and Its Applications-ICCSA 2005*, O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganá, H. P. Lee, Y. S. Mun, D. Taniar, and C. J. K. Tan, Eds., pp. 614–623, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[20] M.-J. Qin, Y.-L. Zhao, and Z.-J. Ma, "Practical constant-size ring signature," *Journal of Computer Science and Technology*, vol. 33, no. 3, pp. 533–541, 2018.

[21] T. H. Yuen, J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Efficient linkable and/or threshold ring signature without random oracles," *The Computer Journal*, vol. 56, no. 4, pp. 407–421, 2013.

[22] 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 *Progress in Cryptology-INDOCRYPT 2004*, A. Canteaut and K. Viswanathan, Eds., pp. 384–398, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[23] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," in *Information Security and Privacy*, H. Wang, J. Pieprzyk, and V.

Varadharajan, Eds., pp. 325–335, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[24] D. Zheng, X. Li, K. Chen, and J. Li, "Linkable ring signatures from linear feedback shift register," in *Emerging Directions in Embedded and Ubiquitous Computing*, M. K. Denko, C. S. Shih, K. C. Li, S. L. Tsao, Q. A. Zeng, S. H. Park, Y. B. Ko, S. H. Hung, and J. H. Park, Eds., pp. 716–727, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[25] D. Y. Liu, J. K. Liu, Y. Mu, W. Susilo, and D. S. Wong, "Revocable ring signature," *Journal of Computer Science and Technology*, vol. 22, no. 6, pp. 785–794, 2007.

[26] C. A. Melchor, P.-L. Cayrel, P. Gaborit, and F. Laguillaumie, "A new efficient threshold ring signature scheme based on coding theory," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4833–4842, 2011.

[27] Y. Tang, F. Xia, Q. Ye, M. Wang, R. Mu, and X. Zhang, "Identity-based linkable ring signature on ntru lattice," *Security and Communication Networks*, vol. 2021, Article ID 9992414, 17 pages, 2021.

[28] M. Hu and Z. Liu, *Lattice-Based Linkable Ring Signature in the Standard Model*, Cryptology ePrint Archive, 2022.

[29] L. Li, J. Liu, L. Cheng et al., "Creditcoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[30] E. Fujisaki and K. Suzuki, "Traceable ring signature," in *International Workshop on Public Key Cryptography*, pp. 181–200, Springer, 2007.

[31] E. Fujisaki, "Sub-linear size traceable ring signatures without random oracles," in *Topics in Cryptology-CT-RSA 2011*, A. Kiayias, Ed., pp. 393–415, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[32] 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," *Theoretical Computer Science*, vol. 469, pp. 1–14, 2013.

[33] H. Feng, J. Liu, D. Li, Y.-N. Li, and Q. Wu, "Traceable ring signatures: general framework and post-quantum security," *Designs, Codes and Cryptography*, vol. 89, no. 6, pp. 1111–1145, 2021.

[34] J. Herranz and G. Sáez, "Forking lemmas for ring signature schemes," in *Progress in Cryptology - INDOCRYPT 2003*, T. Johansson and S. Maitra, Eds., pp. 266–279, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[35] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *International workshop on public key cryptography*, pp. 277–290, Springer, 2004.

[36] S. Schäge and J. Schwenk, "A cdh-based ring signature scheme with short signatures and public keys," in *Financial Cryptography and Data Security*, R. Sion, Ed., pp. 129–142, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[37] X. Zhang, J. K. Liu, R. Steinfeld, V. Kuchta, and J. Yu, "Revocable and linkable ring signature," in *Information Security and Cryptology*, Z. Liu and M. Yung, Eds., pp. 3–27, Springer International Publishing, Cham, 2020.