Revocable One-Time Ring Signature from Pairings

Xu Han, Dawei Zhang, Zongmin Huang, Shuang Yao, and Zuodong Wu

1School of Computer and Information Technology, Beijing Jiaotong University, Beijing, Beijing 100044, China
2Beijing 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

Copyright © 2022 Xu Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 $s_k$ and randomly chooses $r_j \in \mathbb{Z}_p^*$ to generate $L_j$ by using the commit function $C(s_k, r_j)$; then, the signer uses $L_j$ to compute the $(j+1)$-th challenge $\epsilon_{j+1}$ by hash function $H$; signer randomly picks a response $z_{j+1}$ and the $(j+1)$-th user’s public key $p_{k_{j+1}}$ to reconstruct the $L_{j+1}$ by the verify function $Ver$ and generates $\epsilon_{j+2}$ by hash function $H$; then, the ring is formed sequentially; finally, the signer uses $s_k, \epsilon_j, r_j$ 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 $s_k$ and a set of users’ public keys which contains $p_{k_j}$. In the view of the actual signer, users except the actual signer can be seen as decoys. When the verifier does the verification, 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
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 IRS 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, IRS 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 IRS 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.

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 (IRS) 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 secure 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 IRS schemes based on zero knowledge proof. In 2007, Zheng [24] designed an IRS scheme based on linear feedback shift register under discrete logarithm assumption. In 2014, Liu et al. [25] put forward the first unconditional anonymous IRS scheme and provide mandatory linkability. Recently, Noether [26] proposed a dual IRS 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 IRS 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 IRS 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 identity-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 $G_1$ and $G_T$ be cyclic groups of a large prime order $p$. We write $G_1$ additively and $G_T$ multiplicatively. We assume that the discrete logarithm problems in $G_1$ and $G_T$ are intractable.

Let $G$ be a bilinear group generator that, on input of a security parameter $\kappa$, outputs a description of bilinear groups $(G_1, G_T, e, P)$ such that $G_1$ and $G_T$ are cyclic groups of prime order $p$, $P$ is a generator of $G_1$, and a map $e: G_1 \times G_1 \rightarrow G_T$ satisfies the following properties:

(i) Bilinear: \[ \forall P \in G_1 \ and \ a, b \in \mathbb{Z}_p^*: \ e(aP, bP) = e(P, P)^{ab} \]

(ii) Non-degenerate: There exists $\forall P \in G_1$, such that $e(P, P) \neq 1$

(iii) Computability: The map $e(P, P)$ is efficiently computable for any $P \in G_1$

2.2. Complexity Assumptions

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

$$e^{DL} := \Pr \left[ \begin{array}{c} (p, G_1) \leftarrow G(\kappa) ; \\ A_{DL}(P, aP) = a : P \leftarrow G_1 ; \\ a \leftarrow \mathbb{Z}_p^* ; \end{array} \right] = \text{negl}(\kappa). \tag{1}$$

$$\text{Pr}$$
Table 1: The symbol description.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>κ</td>
<td>A security parameter</td>
</tr>
<tr>
<td>p</td>
<td>A large prime number</td>
</tr>
<tr>
<td>ℤ_p</td>
<td>The set consisting of positive integers less than p</td>
</tr>
<tr>
<td>G_1, P</td>
<td>The additive group with p order and a generator</td>
</tr>
<tr>
<td>G_T</td>
<td>The multiplicative group with prime number p order</td>
</tr>
<tr>
<td>e</td>
<td>A bilinear pairing, where e: G_1 × G_1 → G_T</td>
</tr>
<tr>
<td>pk_revoke</td>
<td>A revocation authority’s public key, where pk_revoke = y = sk_revoke P = x P</td>
</tr>
<tr>
<td>H</td>
<td>A cryptographic hash function, where H: {0, 1}^* → ℤ_p^*</td>
</tr>
<tr>
<td>H_1</td>
<td>A deterministic hash function, where H_1: E(G_1) → E(G_1)</td>
</tr>
<tr>
<td>H_2</td>
<td>A cryptographic hash function, where H_2: G_1 → ℤ_p^*</td>
</tr>
</tbody>
</table>

Table 2: Comparison of ring signature schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Signature size</th>
<th>Sign</th>
<th>Verify</th>
<th>Assumption</th>
<th>Security model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang et al. [35]</td>
<td>n</td>
<td>G_1</td>
<td></td>
<td>(2n - 1)δm</td>
<td>(n + 1)</td>
</tr>
<tr>
<td>Schäge et al. [36]</td>
<td>(n + 1)</td>
<td>G_1</td>
<td></td>
<td>(n + 2)</td>
<td>Sm + nadd</td>
</tr>
<tr>
<td>Liu et al. [25]</td>
<td>(2n + 3)</td>
<td>G_T</td>
<td></td>
<td>(3n + 4)</td>
<td>Sm + (4n + 3)</td>
</tr>
<tr>
<td>roRS</td>
<td>n</td>
<td>G_1</td>
<td>+ n</td>
<td>Z_p^*</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The notions in comparison.

<table>
<thead>
<tr>
<th>Notion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ℤ_p</td>
<td>The length of the elements in ℤ_p^*</td>
</tr>
<tr>
<td></td>
<td>G_1</td>
</tr>
<tr>
<td></td>
<td>G_2</td>
</tr>
<tr>
<td></td>
<td>G_T</td>
</tr>
<tr>
<td>add</td>
<td>The time required for an addition computation</td>
</tr>
<tr>
<td>δm</td>
<td>The time required for a scalar multiplication computation</td>
</tr>
<tr>
<td></td>
<td>Pair</td>
</tr>
<tr>
<td>ROM</td>
<td>The abbreviation for random Oracle model</td>
</tr>
<tr>
<td>StanM</td>
<td>The abbreviation for standard model</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Signature size</th>
<th>Revoke</th>
<th>Assumption</th>
<th>Linkability</th>
<th>Mandatory Revocability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu et al. [29]</td>
<td>(2n + 2)</td>
<td>Z_p^*</td>
<td>n</td>
<td>Pair</td>
<td>DBDH</td>
</tr>
<tr>
<td>Fujisaki et al. [30]</td>
<td>(2n + 1)</td>
<td>Z_p^*</td>
<td>2nadd + 2nδm</td>
<td>Dl, DDH</td>
<td>√</td>
</tr>
<tr>
<td>roRS</td>
<td>n</td>
<td>G_1</td>
<td>+ n</td>
<td>Z_p^*</td>
<td>δm + add</td>
</tr>
</tbody>
</table>
Definition 2 Computational Diffie-Hellman (CDH) assumption. Let $G$ be a group generator that, on input of a security parameter $\kappa$, outputs a cyclic group. We say that the CDH assumption holds if for any polynomial-time adversary $\mathcal{A}$, the following advantage $\epsilon^{\text{CDH}}$ is negligible function in $\kappa$:

$$
\epsilon^{\text{CDH}} = \Pr \left[ \mathcal{A}_{\text{CDH}}(p, aP, bP) = abP : \begin{array}{c}
p, G_1 \leftarrow G(\kappa) ; \\
p \leftarrow G_1 ; \\
a, b \leftarrow \mathbb{Z}_p^* ;
\end{array} \right] = \text{negl}(\kappa).
$$

(2)

Definition 3 Decisional bilinear Diffie-Hellman (DBDH) assumption. Let $e : G_1 \times G_1 \rightarrow G_T$ be a bilinear pairing, $P \in G_1$. For $Z \in G_T$, given the tuples $(p, aP, bP, cP, Z)$, we say that the DBDH assumption holds if for any polynomial-time adversary $\mathcal{A}$, the following advantage $\epsilon^{\text{DBDH}}$ is negligible function in $\kappa$:

$$
\epsilon^{\text{DBDH}} = \Pr \left[ \mathcal{A}_{\text{DBDH}}(p, aP, bP, cP, e(P, P)^abc) = 1 \right] - \Pr \left[ \mathcal{A}_{\text{DBDH}}(p, aP, bP, cP, Z) = 1 \right] = \text{negl}(\kappa).
$$

(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) $pp \leftarrow \text{Setup}(\kappa)$: The setup algorithm is a probabilistic polynomial time algorithm which takes as input a security parameter $\kappa \in \mathbb{N}$ and outputs a set of public parameters $pp$.

(ii) $(sk_i, pk_i) \leftarrow \text{KeyGen}(pp)$: 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) $\sigma \leftarrow \text{Sign}(I, sk, Y, pk_{\text{revoke}}, M)$: 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 $pk_{\text{revoke}} \in PK$, and a list $Y$ of public keys in $PK$ which includes the one corresponding to $sk$ and produces a signature $\sigma$.

(iv) $\text{accept/reject} \leftarrow \text{Verify}(I, Y, M, pk_{\text{revoke}}, \sigma)$: 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 $pk_{\text{revoke}} \in PK$, a message $M$, and a signature $\sigma$ and returns accept or reject.

(v) $\text{linked/unlinked} \leftarrow \text{Link}(I_1, I_2, Y_1, Y_2, M_1, M_2, \sigma_1, \sigma_2)$: 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 $\sigma_1$ and $\sigma_2$, such that $\text{Verify}(I_1, Y_1, M_1, \sigma_1) = \text{accept}$ and $\text{Verify}(I_2, Y_2, M_2, \sigma_2) = \text{accept}$ and returns linked or unlinked.

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

$$
\Pr \left[ \text{Verify}(I, Y, M, pk_{\text{revoke}}, \sigma^\ast) = \text{accept} \right] = 1.
$$

(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 $\mathcal{A}$ the following oracles:

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

(ii) CO (corruption oracle). $O_{\text{corrupt}}$; $sk_i \leftarrow \text{CO}(pk_i)$. $\mathcal{A}$ queries this oracle with a public key $pk_i \in PK$ in
of adversary $A$ experiment.

(iii) SO (signing oracle). $O_{\text{sign}}$:

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

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

$$
\mathsf{Adv}_{A}^{\text{unforge}}(\kappa) = \Pr \left[ \text{Verify}(I, Y, p_{\text{revoke}}, M, \sigma^*) = \text{accept} \wedge pk_i \in T_{\text{join}} \land \sigma^* \notin T_{\text{sign}} \right].
$$

(5)

Here, $n \in \mathbb{N}$, $Y = \{p_{k_1}, p_{k_2}, \ldots, p_{k_n}\} \in PK$, and $I$ is a key image. $p_{\text{revoke}}$ is a revocation authority’s public key, and all public keys are in $T_{\text{join}}$. No public keys are in $T_{\text{corrupt}}$.

$\sigma^*$ is not in $T_{\text{sign}}$. $\mathsf{Adv}_{A}^{\text{unforge}}(\kappa)$ is the successful probability of adversary $A$ who wins the unforgeability security experiment.

The process of unforgeability security experiment $\text{Expt}_{A}^{\text{unforge}}$ is briefly described as follows:

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

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

(3) $A$ outputs a forgery signature $\sigma^*$.

The conditions that $A$ wins the experiment are as follows:

(1) All public keys are outputs of $\text{JO}$.

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

(3) No public keys have been queried to $\text{CO}$.

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 $A$, the advantage $\mathsf{Adv}_{A}^{\text{unforge}}(\kappa)$ is negligible in $\kappa$.

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

$$
\mathsf{Adv}_{A}^{\text{anonymous}}(\kappa) = \Pr \left[ s = s' : \begin{cases} pp \leftarrow \text{Setup}(\kappa) ; \\
(I, Y, M) \leftarrow A^{\text{JO}}(pp) ; \\
\sigma_i \leftarrow \text{Sign}(I, Y, sk_i, p_{\text{revoke}}, M) ; \\
\sigma'_i \leftarrow A(\sigma_i) ; \\
\end{cases} \right] - \frac{1}{n}.
$$

(6)

Here, $n \in \mathbb{N}$, $Y = \{p_{k_1}, p_{k_2}, \ldots, p_{k_n}\} \in PK$, and $I$ is a key image. $p_{\text{revoke}}$ is a revocation authority’s public key, and $p_{k_i}$ is chosen by $A$ in $T_{\text{join}}$. $sk_i$ is a corresponding private key of $p_{k_i}$. $\mathsf{Adv}_{A}^{\text{anonymous}}(\kappa)$ is the successful probability of adversary $A$ who wins the anonymity security experiment.

The process of anonymity security experiment $\text{Expt}_{A}^{\text{anonymous}}$ is briefly described as follows:

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

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

(3) $A$ gives $S$ a set $Y$ of public keys in $PK$ such that all of the public keys in $Y$ are query outputs of $\text{JO}$, a key image $I$ and a message $M$. $S$ randomly picks $s \in \{1, \ldots, n\}$ and computes $\sigma_i = \text{Sign}(I, Y, sk_i, M)$, where $sk_i$ is a corresponding private key of $p_{k_i}$. $\sigma_i$ is given to $A$.

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

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 $\mathcal{A}$, the advantage $\text{Adv}_{\text{anonymous}}(\mathcal{A})$ is negligible in $\kappa$.

\[
\mathcal{A} \text{Adv}_{\text{anonymous}}(\mathcal{A}) = \Pr \left[ \text{Verify}(I, M, \sigma, \sigma') = \text{accept} \land \sigma' \neq \sigma \land \sigma \notin T_{\text{sign}} \land \text{Link}(\sigma, \sigma') = \text{unlinked} \right]
\]

Here, $n_i \in \mathbb{N}$, $i = 1, 2$; $Y_i = \{p_{k_1}, p_{k_2}, \ldots, p_{k_n}\} \in \mathcal{P}$, a message $M$ and signature $\sigma_i$, $i = 1, 2$; $I$ is a key image, and $p_{k_{\text{revoke}}}$ is a revocation authority’s public key. All $p_{k_i}$ chosen by $\mathcal{A}$ are in $T_{\text{join}}$, $\sigma_i$ are not in $T_{\text{sign}}$, $\mathcal{C}$ has been queried less than 2 times. $\mathcal{A} \text{Adv}_{\text{anonymous}}(\kappa)$ is the successful probability of adversary $\mathcal{A}$ who wins the anonymity security experiment.

The process of anonymity security experiment $\text{Expt}_{\text{anonymous}}$ is briefly described as follows:

1. $\mathcal{A}$ runs $\text{Setup}(\kappa)$ with security parameter $\kappa$ and sends the public parameter $pp$ to $\mathcal{A}$
2. $\mathcal{A}$ is allowed to make queries to $O_{\text{join}}$, $O_{\text{corrupt}}$, and $O_{\text{sign}}$ according to any adaptive strategy
3. $\mathcal{A}$ outputs a forgery signature $\sigma$

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

\[
\mathcal{A} \text{Adv}_{\text{anonymous}}(\mathcal{A}) = \Pr \left[ \text{Verify}(I, M, \sigma, \sigma') = \text{accept} \land \sigma' \neq \sigma \land \sigma \notin T_{\text{sign}} \land p_{k_i} \notin T_{\text{corrupt}} \land p_{k_j} \notin T_{\text{sign}} \land \text{Link}(\sigma, \sigma') = \text{linked} \right]
\]

Here, $Y, Y^* \in \mathcal{P}$, message $M, M^*$, and signature $\sigma', \sigma^*$; $p_{k_{\text{revoke}}}$ is a revocation authority’s public key, and $I$ is a key image. $\mathcal{A}$ is subject to the restriction that $p_{k_j}$ chosen by $\mathcal{A}$ is not allowed to be either in $T_{\text{corrupt}}$ or $T_{\text{sign}}$. All of the public keys in $Y^*$ and $Y$ are in $T_{\text{join}}$, $\sigma' \neq \sigma^*$ and $\sigma^* \neq \sigma'$ are not in $T_{\text{sign}}$. $\mathcal{A} \text{Adv}_{\text{anonymous}}(\mathcal{A})$ is the successful probability of adversary $\mathcal{A}$ who wins the non-anonymity security experiment.

The process of non-anonymity security experiment $\text{Expt}_{\text{non-anonymous}}$ is briefly described as follows:

1. $\mathcal{A}$ runs $\text{Setup}(\kappa)$ with security parameter $\kappa$ and sends the public parameter $pp$ to $\mathcal{A}$
2. $\mathcal{A}$ is allowed to make queries to $O_{\text{join}}$, $O_{\text{corrupt}}$, and $O_{\text{sign}}$ according to any adaptive strategy
3. $\mathcal{A}$ outputs a forgery signature $\sigma$

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

\[
\mathcal{A} \text{Adv}_{\text{non-anonymous}}(\mathcal{A}) = \Pr \left[ \text{Verify}(I, M, \sigma, \sigma') = \text{accept} \land \sigma' \neq \sigma \land \sigma \notin T_{\text{sign}} \land \text{Link}(\sigma, \tau) = \text{linked} \right]
\]

The process of non-slanderability security experiment $\text{Expt}_{\text{non-slander}}$ is briefly described as follows:

1. $\mathcal{A}$ runs $\text{Setup}(\kappa)$ with security parameter $\kappa$ and sends the public parameter $pp$ to $\mathcal{A}$
2. $\mathcal{A}$ is not an output of $SO$
3. $\mathcal{A}$ can only make queries to $CO$ at most once
4. $\mathcal{A}$ outputs a forgery signature $\sigma$

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

\[
\mathcal{A} \text{Adv}_{\text{non-slander}}(\mathcal{A}) = \Pr \left[ \text{Verify}(I, M) = \text{accept} \land \sigma \notin T_{\text{sign}} \right]
\]

3.2.3. Linkability. We define linkability via the following security experiment $\text{Expt}_{\text{linkable}}$ between $\mathcal{A}$ and $S$:

\[
\mathcal{A} \text{Adv}_{\text{linkable}}(\mathcal{A}) = \Pr \left[ \text{Verify}(I, M, \sigma, \sigma') = \text{accept} \land \sigma' \neq \sigma \land \sigma \notin T_{\text{sign}} \land \text{Link}(\sigma, \sigma') = \text{linked} \right]
\]

(1) All public keys are outputs of $JO$
(2) $\mathcal{A}$ is allowed to make queries to $O_{\text{join}}$, $O_{\text{corrupt}}$, and $O_{\text{sign}}$ according to any adaptive strategy
(3) $\mathcal{A}$ outputs a forgery signature $\sigma$
(4) $\mathcal{A}$ wins the experiment are as follows:

\[
\mathcal{A} \text{Adv}_{\text{linkable}}(\mathcal{A}) = \Pr \left[ \text{Verify}(I, M) = \text{accept} \land \sigma \notin T_{\text{sign}} \right]
\]

3.2.4. Non-slanderability. We define non-slanderability via the following security experiment $\text{Expt}_{\text{non-slander}}$ between $\mathcal{A}$ and $S$:

\[
\mathcal{A} \text{Adv}_{\text{non-slander}}(\mathcal{A}) = \Pr \left[ \text{Verify}(I, M) = \text{accept} \land \sigma \notin T_{\text{sign}} \right]
\]

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

(1) $\mathcal{A}$ runs $\text{Setup}(\kappa)$ with security parameter $\kappa$ and sends the public parameter $pp$ to $\mathcal{A}$
(2) $\mathcal{A}$ is allowed to make queries to $O_{\text{join}}$, $O_{\text{corrupt}}$, and $O_{\text{sign}}$ according to any adaptive strategy
(3) $\mathcal{A}$ outputs a forgery signature $\sigma$

(4) All public keys are in $Y^*$; $Y$ are query outputs of $JO$
(5) \( \text{Link}(\sigma', \sigma^*) = \text{linked} \)

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 \( \mathcal{A} \), the advantage 
\( \text{Adv}^{\text{non-slanderous}}_{\mathcal{A}}(\kappa) \) is negligible in \( \kappa \).

3.2.5. Revocability. We define revocability via the following security experiment \( \text{Expt}_{\mathcal{A}}^{\text{revocable}} \) between \( \mathcal{A} \) and \( S \):

\[
\text{Adv}_{\mathcal{A}}^{\text{revocable}}(\kappa) = \Pr \left[ \text{Verify}(I, Y, p_{\text{revoke}}, M, \sigma) = \text{accept} \wedge 
\quad p_{k_i} \in T_{\text{join}} \wedge \sigma \notin T_{\text{sign}} \wedge 
\quad \exists j \neq s : \text{Revoke}(Y, \sigma, s_{\text{revoke}}), j \neq s 
\right].
\]

Here, \( n \in \mathbb{N}, Y = \{ p_{k_1}, p_{k_2}, \ldots, p_{k_n} \} \in PK, \) signature \( \sigma \), and \( p_{\text{revoke}} \) is a revocation authority’s public key, and \( I \) is a key image. All \( p_{k_i} \) chosen by \( \mathcal{A} \) are in \( T_{\text{join}} \). \( \sigma \) is not in \( T_{\text{sign}} \). \( C \) \( \mathcal{O} \) has been queried less than 2 times, that is, \( \mathcal{A} \) can only obtain at most one private key denotes as \( s_{k_i} \), and \( p_{k_{\text{revoke}}} \) is the corresponding public key of \( s_{\text{revoke}} \). \( \text{Adv}_{\mathcal{A}}^{\text{revocable}}(\kappa) \) is the successful probability of adversary \( \mathcal{A} \) who wins the revocability security experiment.

The process of revocability security experiment \( \text{Expt}_{\mathcal{A}}^{\text{revocable}} \) is briefly described as follows:

1. \( \mathcal{A} \) runs \( \text{Setup}(\kappa) \) with security parameter \( \kappa \) and sends the public parameter \( pp \) to \( \mathcal{A} \).
2. \( \mathcal{A} \) is allowed to make queries to \( O_{\text{Join}}, O_{\text{Corrupt}}, \) and \( O_{\text{Sign}} \) according to any adaptive strategy.
3. \( \mathcal{A} \) outputs a forgery signature \( \sigma \).

The conditions that \( \mathcal{A} \) wins the experiment are as follows:

1. \( \text{Verify}(I, Y, p_{\text{revoke}}, M, \sigma) = \text{accept} \)
2. \( \sigma^* \) is not an output of \( SO \)
3. All public keys are query outputs of \( IO \)
4. \( CO \) has been queried less than 2 times, that is, \( \mathcal{A} \) can only obtain at most one private key denotes as \( s_{k_i} \)
5. \( y_j = \text{Revoke}(Y, \sigma, s_{\text{revoke}}) \) 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 \( \mathcal{A} \), the advantage 
\( \text{Adv}_{\mathcal{A}}^{\text{revocable}}(\kappa) \) is negligible in \( \kappa \).

4. Construction

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

1. Randomly choose \( c_i \in \mathbb{Z}_p^* \), where \( i = \{1, 2, \ldots, n\} \); and randomly choose \( z_i \in \mathbb{G}_1 \), where \( i = \{1, 2, \ldots, s-1, s+1, \ldots, n\} \) (\( i \neq s \)). Then, set the following transformations for all users in the ring.
\[
C_i = \begin{cases} 
  cP, & \text{if } i = s \\
  cP + pk_i H_2(z_i), & \text{if } i \neq s 
\end{cases}
\text{ and } 
Z_i = \begin{cases} 
  cK, & \text{if } i = s \\
  cK + IH_2(z_i), & \text{if } i \neq s 
\end{cases}.
\]

(11)

(4) Get the non-interactive challenge:
\[z = H(M, C_1, \cdots, C_n, Z_1, \cdots, Z_n).\] (12)

(5) Randomly choose \( \Phi_j \in G_1 \), 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 \( h_i = H(\Phi_j, Y, I, M) \), where \( i \neq s \)

(6) Then, when \( i = s \), use notions above and compute:
Compute \( \Phi_s \) as a random number:
\[\Phi_s = zP - \sum_{i=1,i \neq s}^n (h_ipk_i + \Phi_i) \in G_1,\] (13)

then make a hash function \( h_s \) as a random number for \( i = s \):
\[h_s = H(\Phi_s, Y, I, M, B_s) \in \mathbb{Z}_p^*,\] (14)

use private key \( x_i \) to make the response \( V \) when \( i = s \):
\[V = (z + h_s x_s) \cdot B_s \in G_1.\] (15)

(7) To close the ring, set \( l_i = \begin{cases} 
  c_i - x_i H_2(\Phi_s), & \text{if } i = s \\
  c_i, & \text{if } i \neq s 
\end{cases}\)

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

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

(1) Compute \( C'_i = l_i P + pk_i H_2(\Phi_i) \) and \( Z'_i = l_i H_1(pk_i) + IH_2(\Phi_i) \), \( B_3 = c(B_1, I) \), and \( h_j = H(\Phi_j, Y, I, M, B_s) \in \mathbb{Z}_p^* \), where \( i = \{1, 2, \cdots, n\} \)

(2) Check whether \( \sum_{i=1}^n (h_i pk_i + \Phi_i) = H(M, C'_1, \cdots, C'_n, Z'_1, \cdots, Z'_n) \cdot P \) and \( e(P, V) = e(\sum_{i=1}^n [h_i pk_i + \Phi_i], B_1) \). If two equalities hold, accept the signature; otherwise, reject it

Link: On receive the tuples:
\[(\sigma_1 = (\cdot, I_1), Y_1, M_1), \quad (\sigma_2 = (\cdot, I_2), Y_2, M_2).\] (16)

The verifier checks if both \( \sigma_1 \) and \( \sigma_2 \) are two valid signatures. If yes, then outputs link if \( I_1 = I_2 \). Otherwise, reject.

Revoke: On receive the tuples \((Y, \sigma, sk_{\text{revoke}})\). The revocation authority first checks whether \( \sigma \) 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 \( y_s = B_2 - \tilde{x} B_1 \), 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:

\[B_3 = e(B_1, I) = e(\omega P, x_s K) = e(\omega x_s P, K) = e(\omega pk_s, K),\]

\[e \left( \sum_{i=1}^n [h_i \cdot pk_i + \Phi_i], B_1 \right) = e \left( \sum_{i=1,i \neq s}^n [h_i \cdot pk_i + \Phi_i] + h_s \cdot pk_s + \Phi_s, B_1 \right) = e(z \cdot P + h_s \cdot pk_s, B_1) = e(P, V).\] (17)

\[y_s = B_2 - \tilde{x} B_1 = \omega y + pk_s - \tilde{x} \omega P = pk_s,\] (19)

where \( \tilde{x} \) 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 \( \mathcal{A} \) with advantage \( \varepsilon \), which means that \( \mathcal{A} \) can forge a valid signature with probability \( \varepsilon \). 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 \( P \in G_1 \). Through the adversary \( \mathcal{A} \), we use the simulator \( S \) which outputs the CDH solution \( abP \). \( S \) randomly selects \( j \in \{1, \ldots, n\} \) as the actual signer in the simulation. So the simulator \( S \) simulates the oracles by interacting with the adversary \( \mathcal{A} \) in the nature way as follows:

1. Setup: \( S \) sets \( pk_j = aP \) and \( b_j = bP \). \( S \) randomly chooses \( u \in \mathbb{Z}_p^* \) and sets \( y = uP \). \( \mathcal{A} \) is given the public parameters \((G_1, G_2, a, P, H, H_1, H_2)\), and the public keys \( Y = \{pk_1, pk_2, \ldots, pk_n\} \).

2. Queries: \( \mathcal{A} \) 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 \( \mathcal{A} \) queries hash values. If an entry for the query is found in \( L_H \), \( S \) returns the same answer to \( \mathcal{A} \). Otherwise, \( S \) generates a random value as an answer to \( \mathcal{A} \), and then the query and the answer are added in \( L_H \).

   (ii) \( H_1 \)-queries: \( S \) maintains a list \( L_{H_1} \). When \( \mathcal{A} \) issues a query \( H_1(a_1) \), \( S \) randomly chooses \( f_1 \in G_1 \) and sets \( H_1(a_1) = f_1 \) 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 \( \mathcal{A} \) issues a query \( H_2(b_1) \), \( S \) randomly chooses \( f_2 \in \mathbb{Z}_p^* \) and sets \( H_2(b_1) = f_2 \) as the answer. The query and the answer then are stored in list \( L_{H_2} \).

   (iv) \( JO \)-queries: When \( \mathcal{A} \) queries \( JO \) at the \( j \)th query, \( S \) returns the corresponding public key \( pk_j = aP \) to \( \mathcal{A} \). When \( \mathcal{A} \) queries \( JO \) at the \( i \)th 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 \( \mathcal{A} \) queries \( CO \) with the public key \( pk_j \) 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 \( \mathcal{A} \) queries \( SO \) with a tuple \((even, t, M, Y, pk_j)\). If \( pk_k \neq pk_j \), \( S \) outputs a signature \( \sigma \) by Sign algorithm. Otherwise, \( S \) maintains a list \( L_{\text{sign}} \) with a tuple \((\text{event}, pk_j, I)\) and performs as follows:

   (1) \( S \) retrieves the list \( L_{\text{sign}} \), if \((pk_j, I)\) is found, and then takes the value \( I \). If not, \( S \) randomly selects a new value \( I \in G_1 \) and adds \((pk_j, I)\) to the list \( L_{\text{sign}} \).

   (2) \( S \) computes \( B_1 = bP, B_2 = b(uP) + aP, B_3 = \varepsilon(B_1, I) \).

   (3) \( S \) chooses a random \( j \in \{1, \ldots, n\} \).

   (4) \( S \) chooses \( l_i \in \mathbb{Z}_p^* \) and \( \Phi_i \in G_1 \) randomly, where

   
   \[
   i = \{1, 2, \ldots, j - 1, j + 1, \ldots, n\} \quad \text{and} \quad i \neq j;
   \]

   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 \( H(\Phi_j, M, Y, I, B_3) = h_i \).

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

   \[
   \Phi_j = z \cdot p - h_j \cdot pk_j + \sum_{l=1, l \neq j}^{n} h_lpk_l + \Phi_i,
   \]

   and stores the value \( h_i = H(\Phi_j, M, Y, I, B_3) \) 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 \( \sigma = (l_1, \ldots, l_n, \Phi_1, \ldots, \Phi_n, V, I, B_1, B_2) \) as a response to \( \mathcal{A} \).

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

\[
\frac{V - V'}{(h_j - h'_j)} = abP.
\]

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. \( \square \)

Theorem 10 Anonymity. Our proposed scheme is anonymous in the random oracle model with the DDH assumption.
Proof. Suppose there exists a PPT adversary \( \mathcal{A} \) with advantage \( \epsilon \). 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 \( \mathbb{Z}_p^* \), \( Z \in \mathbb{G}_T \), and \( P \in \mathbb{G}_1 \). Through the adversary \( \mathcal{A} \), 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 \( \mathcal{A} \) 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 \( \pi = uP \). In addition, \( S \) sets \( pk_i = aP \) and \( K = bP \). \( \mathcal{A} \) is given the system parameters \( P \) and the public keys \( Y = \{pk_1, pk_2, \ldots, pk_n\} \) of signers.
2. Queries: \( \mathcal{A} \) does the same queries with the oracles \((H, H_1, H_2, I, \text{and CO})\) as Theorem 9.
3. Then, \( \mathcal{A} \) queries \( S \), and \( S \) performs the steps as follows:
   1. \( S \) retrieves the list \( L_{\text{sign}} \), if \( (pk_i, L) \) is found, and then takes the value \( I \). If not, randomly selects a new value \( I \in \mathbb{G}_1 \) and adds \( (pk_i, I) \) to the list \( L_{\text{sign}} \).
   2. \( S \) sets \( B_1 = cp \) and \( B_2 = Z \in \mathbb{G}_T \) and computes \( B_2 = c (upP) + bp \).
   3. \( S \) chooses a random \( j \in \{1, \ldots, n\} \).
   4. \( S \) chooses \( l_i \in \mathbb{Z}_p^* \) and \( \Phi_i \in \mathbb{G}_1 \) randomly, where \( i = \{1, 2, \ldots, j - 1, j + 1, \ldots, n\} \) and \( i \neq j \); and \( S \) computes \( C_i = l_iP + f_i(cP) \) and \( Z_i = l_if_i + f_iI \) and then selects \( h_i \in \mathbb{Z}_p^* \) in \( L_{\text{H}} \) such that

\[
Pr \left[ Z = e(P, P)^{bcd} \right] \geq \frac{1}{2} \left( Pr \left[ Z = e(P, P)^{bcd} | S \leftarrow 1 \right] + Pr \left[ Z = e(P, P)^{bcd} | S \leftarrow 0 \right] \right) \geq \frac{1}{2} \left( \frac{1}{2} + \frac{1}{2n} + \frac{\epsilon}{2} + \left( \frac{1}{2} - \frac{1}{2n} \right) \right) = \frac{1}{2} + \frac{\epsilon}{2}.
\]

Therefore, \( S \) can determine whether \( Z = e(P, P)^{abc} \) with the probability more than 1/2 if \( \mathcal{A} \) 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 \( \mathcal{A} \) the public parameters, where \( S \) has at most one private key \( sk_i \), and this private key can correspond to two different keys in ring group \( Y \) for \( i = 1, 2 \). (When \( \mathcal{A} \) queries the CO, \( \mathcal{A} \) can only get one private key. \( \mathcal{A} \) can be allowed to get only one private key.)

\[
H(\Phi_i, M, Y, I, Z) = h_i
\]

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

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

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

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

\( S \) gives \( \sigma \) to \( \mathcal{A} \), and \( \mathcal{A} \) can query the random oracles adaptively and returns a bit \( \eta \in \{0, 1\} \). Suppose \( \mathcal{A} \) guesses that the signer’s index is \( j \in \{1, n\} \). If \( \mathcal{A} \) 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 \( B_2 = e(copp_k, K) = e(c(aP), bp) = e(P, P)^{abc} = Z \). And when \( S \) returns 0, from \( \mathcal{A} \)’s view, all signers has equal probability to sign the signature. Suppose \( \mathcal{A} \) has advantage \( \epsilon \) in the simulation, then we set \( Pr[Z = e(P, P)^{abc}] = 1/2 + \epsilon \).

The probability of the right choice is computed as

\[
Pr \left[ Z = e(P, P)^{bcd} \right] \geq \frac{1}{2} \left( Pr \left[ Z = e(P, P)^{bcd} | S \leftarrow 1 \right] + Pr \left[ Z = e(P, P)^{bcd} | S \leftarrow 0 \right] \right) \geq \frac{1}{2} \left( \frac{1}{2} + \frac{1}{2n} + \frac{\epsilon}{2} + \left( \frac{1}{2} - \frac{1}{2n} \right) \right) = \frac{1}{2} + \frac{\epsilon}{4}.
\]

So through the queries, \( \mathcal{A} \) can output two valid signatures:

\[
\sigma^{(1)} = (\cdots, \Phi^{(1)}_1, \cdots, \Phi^{(1)}_n, V^{(1)}, I^{(1)}),
\]

\[
\sigma^{(2)} = (\cdots, \Phi^{(2)}_1, \cdots, \Phi^{(2)}_n, V^{(2)}, I^{(2)}),
\]

and the key image of \( \sigma^{(1)} \) is \( I^{(1)} = \chi H_1(pk_j) \), and the key image of \( \sigma^{(2)} \) is \( I^{(2)} = \chi H_1(pk_j) \).

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

\[
\hat{\sigma}^{(1)} = (\cdots, \Phi^{(1)}_1, \cdots, \Phi^{(1)}_n, \hat{V}^{(1)}, \hat{I}^{(1)}).
\]
If $\Phi_s^{(1)} \neq \Phi_s^{(1)}$, abort. If $\Phi_s^{(1)} \neq \Phi_s^{(1)}$, we have
\[
\tilde{V}^{(1)} - V^{(1)} = (z + h_s^{(1)} x_s) P - (z + h_s^{(1)} x_s) P,
\]
where $I^{(1)} = x_s H_1(pk_s)$ and $y_j = x_s P = pk_s$.

Therefore, $x_s = x'_s$ and $l^{(1)} = l^{(2)}$. Two signatures $(\sigma^{(1)}, \sigma^{(2)})$ are linked. $S$ can break DLP, and the advantage of $\mathcal{A}$ 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 $\mathcal{A}$ can query CO to get any public key, but $\mathcal{A}$ cannot be allowed to query the signer’s public key $pk_s$. But $\mathcal{A}$ can give simulator $S$ the tuples $(pk_s, Y, M, pk_{\text{revoke}})$. $S$ uses the tuple to generate a valid signature $\sigma = (I, l)$ which $I$ is the key image computed using $x_s$. $(\sigma^{(1)}, \sigma^{(2)})$ can keep querying oracles with the restriction of submitting $pk_s$ to CO.

Suppose $\mathcal{A}$ generates another valid signature $\sigma' = (I', l')$ which $I'$ is computed by $x'_s$ and $\sigma'$ is not an output of $SO$. Additionally, $\sigma$ and $\sigma'$ are linked. Therefore, $I = I'$ which means $l = l'$. This implies that $\mathcal{A}$ knows the secret key $x_s$ corresponding to $pk_s$. This is opposite to our assumption that $\mathcal{A}$ cannot query CO 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 $\mathcal{A}$ the public parameters, where $\mathcal{A}$ can get one private key denoted as $sk_s = x_s$ corresponding to $pk_s$. Therefore, $\mathcal{A}$ can break revocation with $\mathcal{A}$ randomly and independently distributed, $\mathcal{A}$ cannot tell out the corresponding secret keys according to our assumption. Then, we suppose that $\mathcal{A}$ can generate a valid signature $\sigma = (l, \cdots, l_n, \Phi_1, \cdots, \Phi_n, V, I, B_1, B_2)$ 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. $\mathcal{A}$ 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, $\mathcal{A}$ must know the secret key $x_j$. Due to our assumption that $\mathcal{A}$ 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


