Revocable and Linkable Ring Signature

Xinyu Zhang, Joseph K. Liu(✉), Ron Steinfeld, Veronika Kuchta, and Jiangshan Yu

Faculty of IT, Monash University, Melbourne, Australia
rayzhang.prc@gmail.com,
{Joseph.Liu,Ron.Steinfeld,Veronika.Kuchta,Jiangshan.Yu}@monash.edu

Abstract. In this paper, we construct a revocable and linkable ring signature (RLRS) scheme, which enables a revocation authority to revoke the anonymity of the real signer in linkable ring signature scheme under any circumstances. In other words, the revocability of RLRS is mandatory. The proposed RLRS scheme inherits the desired properties of group signature (anonymity revocation) and linkable ring signature (spontaneous group formation and linkability). In addition, we proved the security of our scheme in the random oracle model. We also provided a revocable ring confidential transaction protocol based on our RLRS scheme, which embedded the revocability in ring confidential transaction protocol.

Keywords: Ring signature · Ring confidential transaction · Revocability · Linkability

1 Introduction

1.1 Ring Signature and Variants

Ring Signature. Ring signature schemes (e.g., [1,6,13,28]) allow the user to sign a message on behalf of a spontaneous group in an anonymous way. Unlike group signature, ring signature scheme does not require the group manager to form the group or distribute keys to group members. In other words, the signer can build the group spontaneously (i.e., without the cooperation of other group members). Another special property of a ring signature scheme is *anonymity*. An honest signer can convince the verifier that the signature is signed by one of the group members, but the identity (i.e., public key) of the real signer remains to be hidden. According to different underlying public key systems, ring signature has enormous amount of different constructions, such as RSA based [28], discrete-logarithm based [13], mixture based [1], pairing based [6], and lattice based [8]. Ring signatures with different features are also proposed, such as forward security [15,22,24], threshold setting [20,23,30,35,36].

Linkable Ring Signature. The notion of linkable ring signature was first introduced in [19]. Linkable ring signature not only inherits the properties of ring

Z. Liu and M. Yung (Eds.): Inscrypt 2019, LNCS 12020, pp. 3–27, 2020.
https://doi.org/10.1007/978-3-030-42921-8_1

signature but also provides *linkability* for the verifier to verify if two signatures are generated by the same signer with respect to an event. *Linkability* is especially important for applications such as e-voting and e-cash. The motivation of [19] is that most of ring signature schemes achieved unconditional anonymity (e.g., [1,6,28]), which means the verifier has no way to determine if two signatures are signed using same private key. After the introduction of linkable ring signature, several schemes with different improvements were proposed (e.g., [2,4,11,12,18,21,31,32]). We summarise their contributions as follows:

- Constant size LRS: [2,31]
- LRS with unconditional anonymity: [18]
- LRS with traceability: [11,12]
- LRS with enhanced security: [21]
- Certificate-based LRS: [4]
- LRS with separability: [32]

Revocable (Traceable) Ring Signature. Another variant of ring signature is called traceable or revocable ring signature (e.g., [3,5,9,12,16,17,25]). The main objective of revocable (traceable) ring signature schemes is to provide a way to reduce the anonymity of ring signature. Revocable (traceable) ring signature is different from group signature since the signer still can form the group spontaneously. There are three categories of revocable (traceable) ring signature:

- Revocable Ring Signature (e.g., [17]): A set of pre-defined revocation authorities are able to open the anonymity of a ring signature at any time they want, [17] called this property a *Mandatory Revocability.*
- Traceable Ring Signature (e.g., [3,5,12]): The revocability (traceability) in traceable ring signature is not mandatory, that is, the signer's identity will be revealed universally if and only if he/she submits two signatures which are generated using the same private key based on an event.
- Convertible/Verifiable Ring Signature (e.g., [9,16,25]): The scheme is revocable if the signer wants to prove the ownership of a ring signature to the verifier. However, if the signer is reluctant to reveal his/her identity, the signature remains anonymous.

Nonetheless, ring signature schemes (and variants) we mentioned above do not provide the mandatory revocability as well as linkability. For example, traceable ring signature [12] can trace a signer only when the signer was double-signing. In other words, no one can determine the identity of an honest signer. In contrast, revocable ring signature [17] provides mandatory revocability, whereas the scheme cannot detect if two signatures are linked.

1.2 Ring Confidential Transaction

Monero, one of the largest cryptocurrencies, was introduced in 2014. Unlike BitCoin [26], Monero concentrates on protecting transaction privacy by applying ring signature techniques. Originally, Monero was based on CryptoNote protocol [33] which exploits properties of traceable ring signature [12] to enhance transaction anonymity as well as prevent the double-spending attack. Later in 2015, Noether [27] proposed *Ring Confidential Transaction* (Ring CT), which is based on linkable ring signature [19], to further advance the technique by solving several practical issues in CryptoNote. Specifically, Monero allows users to have multiple different accounts. Each account contains a "one-time" public key as account address and the coin. To authorise a transaction, the user has to use his/her corresponding private key. In order to construct an anonymous transaction, the user also needs to select several decoys (i.e., other users' account addresses) and generates a linkable ring signature.

Nevertheless, anonymity is not always good. According to the research [7], around US$1.6 trillion was laundered in 2009. As claimed in [14], untraceability and lack of supervision of cryptocurrencies stimulate cyber-crimes like money laundering and terrorist financing. Authorities, such as FBI, found it is hard to "detect suspicious activities, identify users, and obtain the transaction" [10]. Therefore, it is critical to provide a method for the authority to supervise the transactions on blockchain, or at least, to revoke the anonymity of a suspicious spender.

1.3 Our Contributions

The contribution of the paper contains three parts:

- We present a ring signature scheme which achieves both **mandatory revocability** and **linkability**. Specifically, our scheme enables a revocation authority to revoke anonymity of a ring signature in any condition. Besides, our *Revocable and Linkable Ring Signature* (RLRS) scheme inherits the advantages of linkable ring signature schemes - an efficient way to prevent double-signing. We also construct a formal security proof of our RLRS scheme in random oracle model.
- The second contribution is that our scheme has more efficient revocation algorithm than [17] and [12]. The result shows that our scheme only needs one modular multiplication and one exponentiation in the revocation process while the computation time in [17] and [12] is linearly dependent on the group size. We present the comparison between our scheme and [12,17] in Sect. 6.
- The third contribution is that we extended our RLRS scheme to construct *Revocable Ring Confidential Transaction* protocol, which is presented in Appendix A.

Paper Organisation. The paper is organised in 6 parts, including the introduction. We compared our scheme with other revocable/traceable ring signature scheme and point out what are the advantages of our work in Sect. 2. Section 3 initialises the primitives of the scheme, which are utilised throughout the paper. In Sect. 4, we construct a security model and present our revocable and linkable ring signature protocol along with the security analysis of our scheme. Section 5 is an analysis of the efficiency of our scheme. The last section summarises our contributions and proposes several limitations which should be considered in future works.

2 Related Work

Revocable ring signature [17] was proposed in 2007, which shares similar idea with our construction. The scheme in [17] was based on bilinear pairing and proof-of-knowledge. The main advantage of [17] is that their protocol allows a set of revocation authorities to revoke anonymity of the real signer while our construction assumes the authority shares one public key. However, as we mentioned in the previous section, [17] did not introduce linkability to their scheme. The combination of mandatory revocability and linkability can especially benefit the construction of revocable e-cash systems (i.e., supervises users of the system as well as prevents double-spending). Another paper [34] applies revocable ring signature technique to build a bidder-anonymous English auction protocol. However, the scheme in [34] was similar to [17] except for that the revocation authority only has one public-private key pair, and the signer's public key is related to his/her identity in the initial phase.

Another type is called *Traceable ring signature* [12] which is comparable with *Revocable iff linked ring signature* [3,5]. Different from *Revocable ring signature* [17], a traceable ring signature scheme [12] does not enable mandatory signer revocation. Thus, only when the signer tries to generate multiple signatures with the same private key in one event (double-signing), his/her identity (i.e., public key) will be revealed. *Traceable ring signature* are closely related to linkable ring signature. Precisely, in [12], the linking tag in linkable ring signature schemes (e.g., [19,21]) is manipulated to trace the identity of the signer while the instantiation of zero-knowledge-proof in [12] is similar to [21]. We summarise the core function (i.e., signature signing) in [12] to two parts, the first part is to generate the linking tag which can be used to trace the signer, and the second part is based on (non-interactive) zero-knowledge-proof. Nonetheless, the construction in [12] was not very efficient since the signature size linearly depends on the group size. Therefore, in 2011, Fujisaki proposed [11] to enhance the security definition of [12] as well as reduce the signature size to $O(\sqrt{n})$.

The last type which is able to reduce the anonymity of ring signature is called *Convertible(Verifiable) Ring Signature* [9,16,25]. The convertible ring signature scheme [16] and verifiable ring signature scheme [25] achieve similar goals, that is, allow the signer to claim the ownership of the signature. However, if the signer refuses to do so, the signature is still anonymous. Their schemes are mostly based on RSA ring signature proposed in [28]. Nevertheless, as mentioned in [29], their original ring signature [28] is able to perform the function of verifiable ring signature, and they already described such a function in "Generalisations of Special Cases". Another deficiency is that the security model of verifiable and convertible ring signature is too simple. The researchers just explained the security model in [28], where they should build the security model based on their proposed scheme.

3 Preliminaries

3.1 Mathematical Assumptions

Definition 1 (Discrete Logarithm (DL) Assumption). *The Discrete Logarithm assumption in $\mathbb{G}$ is defined as follows: on input a tuple $(y, g) \in \mathbb{G}^2$ where $|\mathbb{G}| = q$ for some prime number q, outputs x such that $y = g^x \pmod q$. We say that $(t, \epsilon) - DL$ assumption holds in $\mathbb{G}$, if no t-time algorithm has advantage at lease ϵ in solving DL problem in $\mathbb{G}$.*

Definition 2 (Decisional Diffie-Hellman (DDH) Assumption). *The Decisional Diffie-Hellman Assumption in $\mathbb{G}$ is defined as follows: on input a quadruple $(g, g^a, g^b, g^c) \in \mathbb{G}^4$, where $|\mathbb{G}| = q$ for some prime number q, output 1 if $c = ab$. Otherwise 0. We say that $(t, \epsilon) - DDH$ assumption holds in $\mathbb{G}$, if no t-time algorithm has advantage at least ϵ over random guessing in solving DDH problem in $\mathbb{G}$.*

3.2 ElGamal Public Key Encryption

In our protocol, we apply ElGamal encryption scheme consisting of the following four algorithms:

1. $param \leftarrow \texttt{Setup}(\lambda)$: On input a security parameter λ, returns public parameters $param = \{\mathbb{G}, q, g\}$, where $\mathbb{G}$ is a group with prime order q such that discrete logarithm is intractable, and g is the generator in $\mathbb{G}$.
2. $(sk, pk) \leftarrow \texttt{KeyGen}(param)$: Takes the input $param = \{\mathbb{G}, q, g\}$, generates a pair of public key ($pk = y$) and secret key ($sk = x$) satisfying $y = g^x \pmod q$.
3. $c \leftarrow \texttt{Encryption}(M, pk_r)$: On input a message M, and a receiver's public key $pk_r = y_r$, the sender randomly picks a number $k \in \mathbb{Z}_q$ and generates the first part of the ciphertext $c_1 = g^k \pmod q$. Then the signer takes y_r and generates the second part of the ciphertext $c_2 = y_r^k M \pmod q$. The final output of the algorithm is $c = \{c_1, c_2\}$.
4. $M \leftarrow \texttt{Decryption}(c, sk_r)$: Takes the input $c = \{c_1, c_2\}$, and receiver's secret key $sk_r = x_r$, recovers the message by computing $M = c_2 \backslash c_1^{x_r} \pmod q$.

3.3 Signature of Knowledge

In our construction, we utilise Honest-Verifier Zero-Knowledge (HVZK) Proof of Knowledge Protocols (PoKs), which can be modified into a signature scheme by setting the challenge to a hash value of a commitment together with the message. The scheme is used in many (linkable) ring signature schemes such as [18,19,21]. A Signature of Knowledge (SoK) protocol contains following algorithms:

1. $param \leftarrow \texttt{Setup}(\lambda)$: On input a security parameter λ, returns a public parameter $param$.
2. $\sigma \leftarrow \texttt{Sign}(M, x, y)$: The algorithm takes a message M, a pair of (x, y), returns a SoK denoted as σ.
3. $0/1 \leftarrow \texttt{Verify}(M, \sigma, y)$: On input a message M, a SoK σ, and a statement y, outputs 0/1.

4 Revocable and Linkable Ring Signature

4.1 Technical Description

A revocable and linkable ring signature scheme **(RLRS)** is a tuple of six algorithms $(\texttt{Setup}, \texttt{KeyGen}, \texttt{Sign}, \texttt{Verify}, \texttt{Link}, \texttt{Revoke})$

- $param \leftarrow \texttt{Setup}(\lambda)$ is a probabilistic polynomial time (PPT) algorithm which, on input a security parameter λ, outputs a set of public parameters $param$.
- $(sk_i, pk_i) \leftarrow \texttt{KeyGen}(param)$ is a PPT algorithm receives public parameters $param$ and returns a private/public key pair (sk_i, pk_i). We denote SK as the domain of possible private keys and PK as the domain of possible public keys.
- $\sigma \leftarrow \texttt{Sign}(event, n, \mathbb{Y}, sk, pk_{rev}, M)$ takes the input of an event description $event$, a group size n, a set $\mathbb{Y}$ contains n public keys $\{pk_1, \ldots, pk_n\}$ such that $pk_i \in PK$ for $i \in [1, n]$, a private key $sk \in SK$ which corresponds to one of the public keys in $\mathbb{Y}$, a public key of revocation authority $pk_{rev} \in PK$, and a message M, produces a signature σ.
- $accept/reject \leftarrow \texttt{Verify}(event, n, \mathbb{Y}, pk_{rev}, M, \sigma)$ accepts the input of an event description $event$, a group size n, a set $\mathbb{Y} = \{pk_1, \ldots, pk_n\}$ of n public keys, where $pk_i \in PK$ for $i \in [1, n]$, a revocation authority's public key $pk_{rev} \in PK$, and a message-signature pair (M, σ). If the message-signature pair is valid, the algorithm outputs $accept$. Otherwise, $reject$.
- $linked/unlinked \leftarrow \texttt{Link}(event, n_1, n_2, \mathbb{Y}_1, \mathbb{Y}_2, M_1, M_2, \sigma_1, \sigma_2)$ receives an event description $event$, two group sizes n_1 and n_2, two sets $\mathbb{Y}_1$ and $\mathbb{Y}_2$ of n_1 and n_2 public keys respectively, where all public keys in $\mathbb{Y}_1$ and $\mathbb{Y}_2$ are in PK, two valid message and signature pairs (M_1, σ_1) and (M_2, σ_2). The algorithm outputs $linked$ if two linking tags in σ_1 and σ_2 are the same. Otherwise $unlinked$.
- $pk \leftarrow \texttt{Revoke}(n, \mathbb{Y}, \sigma, sk_{rev})$ takes as input a group size n, a set of n public keys $\mathbb{Y} = \{pk_1, \ldots, pk_n\}$ such that $pk_i \in PK$ for $i \in [1, n]$, a valid signature σ, and revocation authority's secret key $sk_{rev} \in SK$ corresponding to pk_{rev}, returns a public key pk in $\mathbb{Y}$.

Correctness: A RLRS scheme should satisfy:

- *Verification Correctness*: A signature generated by an honest signer should be identified as a valid signature with overwhelming probability.
- *Linking Correctness*: If two signatures are determined as "linked", then they must have been signed using the same private key with respect to the same event description.
- *Revocation Correctness*: An honest signer's public key will be revealed by the revocation authority with overwhelming probability.

4.2 Security Definitions

Security of RLRS has five aspects, including unforgeability, anonymity, linkability, non-slanderability, and revocability. We define three oracles, which simulate abilities of the adversary:

1. *Joining Oracle* ($\mathcal{JO}$): on request, adds a new user to the system, then returns the public key $pk \in PK$ of the new user.
2. *Corruption Oracle* ($\mathcal{CO}$): on input a public key $pk_i \in PK$, returns the corresponding $sk_i \in SK$.
3. *Signing Oracle* ($\mathcal{SO}$): takes the input of an event description *event*, a group size n, a set $\mathbb{Y} = \{pk_1, \ldots, pk_n\}$ that contains n public keys, a signer's public key $pk_\pi \in \mathbb{Y}$, a revocation authority's public key $pk_{rev} \in PK$, and a message M, returns a valid signature denoted as $\sigma \leftarrow \texttt{Sign}(e, n, \mathbb{Y}, sk_\pi, pk_{rev}, M)$. Note that $\mathcal{SO}$ may query $\mathcal{CO}$ during its operation.

Unforgeability: The unforgeability game is defined between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ with access to $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$:

a. $\mathcal{A}$ runs the `Setup` algorithm on a security parameter λ and outputs *param*.
b. $\mathcal{A}$ can query $\mathcal{JO}, \mathcal{CO}, \mathcal{SO}$ adaptively.
c. $\mathcal{A}$ gives $\mathcal{S}$ an event description *event*, a group size n, a set $\mathbb{Y} = \{pk_1, \ldots, pk_n\}$ of n public keys, where $pk_i \in PK$ for $i \in [1, n]$, a message M, a revocation authority's public key $pk_{rev} \in PK$, and a signature σ.

$\mathcal{A}$ wins the game if:

1. $\texttt{Verify}(event, n, \mathbb{Y}, pk_{rev}, M, \sigma) = accept$;
2. all public keys in $\mathbb{Y}$ are query outputs of $\mathcal{JO}$;
3. no public keys in $\mathbb{Y}$ have been queried to $\mathcal{CO}$; and
4. σ is not a query output of $\mathcal{SO}$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{Unf}(\lambda) = Pr[\mathcal{A} \text{ wins the game}]$$

the success probability of adversary $\mathcal{A}$ in winning the unforgeability game.

Definition 3 (Unforgeability). *A RLRS scheme is existential unforgeable against adaptive chosen message and chosen public key attack if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Unf}(\lambda)$ is negligible.*

Anonymity: any verifier should not have a non-negligible probability greater than $1/n$ of correctly guessing the signer's identity in a valid ring signature when none of the ring members is known. Moreover, any party who has revocation authority's secret key can break anonymity due to the mandatory revocability of our scheme. Therefore, RLRS scheme is computationally anonymous if revocation authority has not been compromised. The anonymity game is defined between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ who is given access to $\mathcal{JO}$:

a. $\mathcal{A}$ runs the `Setup` algorithm on a security parameter λ and outputs *param*.
b. $\mathcal{A}$ can make query to $\mathcal{JO}$ adaptively.
c. $\mathcal{A}$ gives $\mathcal{S}$ an event description *event*, a group size n, a set $\mathbb{Y}$ of n public keys such that all public keys in $\mathbb{Y}$ are generated by $\mathcal{JO}$, a revocation authority's public key $pk_{rev} \in PK$, a message M. $\mathcal{S}$ parses $\mathbb{Y}$ as $\{pk_1, \ldots, pk_n\}$ and randomly picks $\pi \in \{1, \ldots, n\}$. $\mathcal{S}$ computes a "Challenge Signature" σ_π using sk_π, where sk_π is a corresponding private key of pk_π. σ_π is given to $\mathcal{A}$.
d. $\mathcal{A}$ guesses $\pi' \in \{1, \ldots, n\}$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda) = |Pr[\pi' = \pi] - \frac{1}{n}|$$

the success probability of adversary $\mathcal{A}$ in winning the anonymity game.

Definition 4 (Anonymity). *A RLRS scheme is computationally anonymous if for any adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda)$ is negligible.*

Linkability: linkability is a mandatory property of RLRS scheme, which means that the signer cannot generate two signatures using the same private key such that they are determined to be *unlinked* by `Link` algorithm. We adopt the linkability game defined by Liu et al. [19] to capture the scenario, where an adversary tries to generate two RLRS signatures (σ_1, σ_2) using the same private key yet $\texttt{Link}(\cdot, \sigma_1, \sigma_2)$ algorithm outputs *unlinked*. Actually, if RLRS scheme is unforgeable, then the unlinked signatures can only be generated by different private keys with respect to the same event. The linkability game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ who is given access to $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$, is defined as follows:

a. $\mathcal{A}$ runs the `Setup` algorithm on a security parameter λ and outputs *param*.
b. $\mathcal{A}$ can query $\mathcal{JO}, \mathcal{CO}, \mathcal{SO}$ adaptively.
c. $\mathcal{A}$ gives $\mathcal{S}$ an event description *event*, two group size n_1, n_2 with assumption of $n_1 \leq n_2$ without loss of generality, two set $\mathbb{Y}_1, \mathbb{Y}_2$ with n_1, n_2 public keys respectively, two message-signature pairs $(M_1, \sigma_1), (M_2, \sigma_2)$, and a revocation authority's public key pk_{rev}.

$\mathcal{A}$ wins the game if:

1. All public keys in $\mathbb{Y}_1 \cup \mathbb{Y}_2$ are outputs of $\mathcal{JO}$;
2. $\texttt{Verify}(event, n_i, \mathbb{Y}_i, pk_{rev}, M_i, \sigma_i) = accept$ for $i = 1, 2$ such that σ_i is not the output of $\mathcal{SO}$;
3. $\mathcal{CO}$ has been queried less than 2 times, that is, $\mathcal{A}$ can only have at most one user private key; and
4. $\texttt{Link}(\cdot, \sigma_1, \sigma_2) = unlinked$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{Link}(\lambda) = Pr[\mathcal{A} \text{ wins the game}]$$

the success probability of adversary $\mathcal{A}$ in winning the linkability game.

Definition 5 (Linkability). *A RLRS is linkable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Link}(\lambda)$ is negligible.*

Non-slanderability: the attacker should be unable to accuse an honest user for generating a signature which is determined to be *linked* with a malicious signature generated by attacker. The non-slanderability game is defined between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ who is given access to $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$:

a. $\mathcal{A}$ runs the `Setup` algorithm on a security parameter λ and outputs *param*.
b. $\mathcal{A}$ can query $\mathcal{JO}, \mathcal{CO}, \mathcal{SO}$ adaptively.
c. $\mathcal{A}$ gives $\mathcal{S}$ an event description *event*, a group size n, a set $\mathbb{Y}$ of n public keys, a message m, a revocation authority's public key pk_{rev}, and a public key of an insider $pk_\pi \in \mathbb{Y}$ such that pk_π has not been queried to $\mathcal{CO}$ or has not been included as the insider public key of any query to $\mathcal{SO}$. $\mathcal{S}$ uses sk_π corresponding to pk_π to run `Sign`$(event, n, \mathbb{Y}, sk_\pi, pk_{rev}, M)$ and produces σ to $\mathcal{A}$.
d. $\mathcal{A}$ queries oracles with arbitrary interleaving. Particularly, $\mathcal{A}$ can make query to $\mathcal{CO}$ of any public key except for pk_π.
e. $\mathcal{A}$ delivers group size n^*, $\mathbb{Y}^*$ with n^* public keys, a message M^*, a revocation authority's public key pk_{rev}, and a signature $\sigma^* \neq \sigma$.

$\mathcal{A}$ wins the game if

1. `Verify`$(event, n^*, \mathbb{Y}^*, pk_{rev}, M^*, \sigma^*) = accept$;
2. σ^* is not an output of $\mathcal{SO}$;
3. All public keys in $\mathbb{Y}^*, \mathbb{Y}$ are query outputs of $\mathcal{JO}$;
4. pk_π has not been queried to $\mathcal{CO}$; and
5. `Link`$(\sigma^*, \sigma) = linked$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{NS}(\lambda) = Pr[\mathcal{A} \text{ wins the game}]$$

the success probability of adversary $\mathcal{A}$ in winning the non-slanderability game.

Definition 6 (Non-slanderability). *A RLRS scheme is non-slanderable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{NS}$ is negligible.*

Revocability: revocability in RLRS scheme is compulsory, that is, the probability of a signer generates a signature without his/her identity gets revealed by revocation authority should be negligible. We define revocability game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ who is given access to $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$:

a. $\mathcal{A}$ runs the `Setup` algorithm on a security parameter λ and outputs *param*.
b. $\mathcal{A}$ can query $\mathcal{JO}, \mathcal{CO}, \mathcal{SO}$ adaptively.
c. $\mathcal{A}$ can only obtain at most one private key of ring member from $\mathcal{CO}$.
d. $\mathcal{A}$ gives $\mathcal{S}$ an event description *event*, a group size n, a set $\mathbb{Y}$ contains n public keys, a message M, a revocation authority's public key pk_{rev}, and a signature σ.

$\mathcal{A}$ wins the game if

1. $\texttt{Verify}(event, n, \mathbb{Y}, M, \sigma) = accept$;
2. all public keys in $\mathbb{Y}$ are query outputs of $\mathcal{JO}$;
3. σ is not an output of $\mathcal{SO}$;
4. $\mathcal{CO}$ has been queried less than two times ($\mathcal{A}$ can only obtain at most one private key denotes as x_π); and
5. $y_j = \texttt{Revoke}\ (n, \mathbb{Y}, \sigma, sk_{rev})$ where $j \neq \pi$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{Rev}(\lambda) = Pr[\mathcal{A} \text{ wins the game}]$$

the success probability of adversary $\mathcal{A}$ in winning the revocability game.

Definition 7 (Revocability). *A RLRS scheme is revocable if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Rev}(\lambda)$ is negligible.*

4.3 Scheme Description

$\texttt{Setup}(\lambda)$: Let $\mathbb{G}$ be a group with prime order q such that the underlying discrete logarithm problem is intractable, and g is the generator of $\mathbb{G}$. Define two hash functions: $H_1 : \{0,1\}^* \to \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \to \mathbb{G}$. The public parameters are $param = \{\mathbb{G}, g, q, H_1, H_2\}$.

$\texttt{KeyGen}(param)$: Assume there are n users. User i, where $i \in [1, n]$, randomly chooses $x_i \in \mathbb{Z}_q$ and computes $y_i \leftarrow g^{x_i} \pmod q$. User i has secret key and public key pair $\{sk_i, pk_i\}$ such that $sk_i = x_i$ and $pk_i = y_i$.

$\texttt{Sign}(event, n, \mathbb{Y}, sk_\pi, pk_{rev}, M)$: Takes as input $(event, n, \mathbb{Y}, sk_\pi, pk_{rev}, M)$, where *event* is the description of the event, n is the number of users in the ring, $\mathbb{Y} = \{pk_1, pk_2, \ldots, pk_n\}$ is a set of public keys of users in the ring, $sk_\pi = x_\pi$ is the secret key of user π and the corresponding public key is $pk_\pi = y_\pi$, note that $pk_\pi \in \mathbb{Y}$ with $\pi \in [1, n]$, $pk_{rev} = \tilde{y}$ is the public key of the revocation authority, and M is the message to be signed. Assume that the secret key $sk_{rev} = \tilde{x}$ of the authority and the corresponding public key $\tilde{y}$ are generated by $\texttt{KeyGen}$. User π with the knowledge of x_π computes a signature of knowledge as follows:

1. Compute the linking tag L by committing to x_π:
   (a) $h \leftarrow H_2(event)$,
   (b) $L \leftarrow h^{x_\pi}$.
2. Randomly pick $u \in \mathbb{Z}_q$, and compute the ciphertext by using ElGamal Encryption:
   (a) $C_1 \leftarrow g^u$,
   (b) $C_2 \leftarrow \tilde{y}^u y_\pi$,
   (c) $C \leftarrow \{C_1, C_2\}$.
3. Randomly pick $t_1, t_2 \in \mathbb{Z}_q$ and compute the following commitments:
   (a) $a_{1,\pi} \leftarrow g^{t_1}$ and $a_{2,\pi} \leftarrow \tilde{y}^{t_1}$,
   (b) $S'_{\pi+1} \leftarrow H_1(event, \mathbb{Y}, L, M, a_{1,\pi}, a_{2,\pi})$,
   (c) $\bar{a}_{1,\pi} \leftarrow g^{t_2}$ and $\bar{a}_{2,\pi} \leftarrow h^{t_2}$,

   (d) $S''_{\pi+1} \leftarrow H_1(event, \mathbb{Y}, L, M, \bar{a}_{1,\pi}, \bar{a}_{2,\pi})$.
4. For $i = \pi + 1, \ldots, n, 1, \ldots, \pi - 1$, randomly pick $r_{1,i}, r_{2,i} \in \mathbb{Z}_q$, and compute:
   (a) $a_{1,i} \leftarrow g^{r_{1,i}} C_1^{S'_i}$ and $a_{2,i} \leftarrow \tilde{y}^{r_{1,i}} (\frac{C_2}{y_i})^{S'_i}$,
   (b) $S'_{i+1} \leftarrow H_1(event, \mathbb{Y}, L, M, a_{1,i}, a_{2,i})$,
   (c) $\bar{a}_{1,i} \leftarrow g^{r_{2,i}} y_i^{S''_i}$ and $\bar{a}_{2,i} \leftarrow h^{r_{2,i}} L^{S''_i}$,
   (d) $S''_{i+1} \leftarrow H_1(event, \mathbb{Y}, L, M, \bar{a}_{1,i}, \bar{a}_{2,i})$.
5. Compute $r_{1,\pi} \leftarrow t_1 u - S'_\pi u \pmod q$ and $r_{2,\pi} \leftarrow t_2 - S''_\pi x_\pi \pmod q$.
6. The signature is $\sigma = (S'_1, S''_1, r_{1,1}, \ldots, r_{1,n}, r_{2,1}, \ldots, r_{2,n}, L, C)$.

`Verify`$(event, n, \mathbb{Y}, pk_{rev}, M, \sigma)$: On input an event description *event*, a group $\mathbb{Y}$ of n public keys, a revocation authority's public key $pk_{rev} = \tilde{y}$, a message M, and a signature σ, verify the signature as follows:

1. On input σ, parse the ciphertext $C = \{C_1, C_2\}$.
2. For $i = 1, \ldots, n$, compute:
   (a) $Z'_{1,i} \leftarrow g^{r_{1,i}} C_1^{S'_i}$ and $Z'_{2,i} \leftarrow \tilde{y}^{r_{1,i}} (\frac{C_2}{y_i})^{S'_i}$,
   (b) $S'_{i+1} \leftarrow H_1(event, \mathbb{Y}, L, M, Z'_{1,i}, Z'_{2,i})$ if $i \neq n$,
   (c) $Z''_{1,i} \leftarrow g^{r_{2,i}} y_i^{S''_i}$ and $Z''_{2,i} \leftarrow h^{r_{2,i}} L^{S''_i}$,
   (d) $S''_{i+1} \leftarrow H_1(event, \mathbb{Y}, L, m, Z''_{1,i}, Z''_{2,i})$ if $i \neq n$.
3. Check
   (a) $S'_1 \overset{?}{=} H_1(event, \mathbb{Y}, L, m, Z'_{1,n}, Z'_{2,n})$,
   (b) $S''_1 \overset{?}{=} H_1(event, \mathbb{Y}, L, m, Z''_{1,n}, Z''_{2,n})$.

`Link`$(event, n_1, n_2, \mathbb{Y}_1, \mathbb{Y}_2, M_1, M_2, \sigma_1, \sigma_2)$: On input an event description *event*, two groups $\mathbb{Y}_1$, $\mathbb{Y}_2$ with group sizes n_1, n_2 respectively, and two valid message-signature pairs (M_1, σ_1), where $\sigma_1 = (\cdot, L_1)$; (M_2, σ_2), where $\sigma_2 = (\cdot, L_2)$, output *linked* if $L_1 = L_2$. Otherwise, *reject*.

`Revoke`$(n, \mathbb{Y}, \sigma, sk_{rev})$: On input$(n, \mathbb{Y}, \sigma, sk_{rev})$, where n is the group size of ring group $\mathbb{Y}$, σ is the signature generated, $sk_{rev} = \tilde{x}$ is authority's secret key corresponds to $pk_{rev} = \tilde{y}$. The revocation authority first check whether the signature is valid. If yes, continue. Otherwise, abort. To revoke the anonymity of the real signer, the revocation authority computes as follows:

1. $C = \{C_1, C_2\}$.
2. $\exists y_\pi \in \mathbb{Y}(\pi \in [1, n])$ such that $y_\pi = C_2 \backslash C_1^{\tilde{x}}$.

y_π is the public key of the real signer.

4.4 Correctness Analysis

Verification Correctness. From the construction of revocable and linkable ring signature, we start with $S'_{\pi+1}, S''_{\pi+1}$, where π denotes the real signer's index such that $\pi \in [1, n]$ without loss of generality:

$$\begin{aligned}
S'_{\pi+1} &= H_1(event, \mathbb{Y}, L, m, g^{t_1}, (\tfrac{C_2}{y_\pi})^{t_1}) \\
S'_{\pi+2} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,\pi+1}} C_1^{S'_{\pi+1}}, \tilde{y}^{r_{1,\pi+1}} (\tfrac{C_2}{y_{\pi+1}})^{S'_{\pi+1}}) \\
&\vdots \\
S'_{n} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,n-1}} C_1^{S'_{n-1}}, \tilde{y}^{r_{1,n-1}} (\tfrac{C_2}{y_{n-1}})^{S'_{n-1}}) \\
S'_{1} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,n}} C_1^{S'_{n}}, \tilde{y}^{r_{1,n}} (\tfrac{C_2}{y_{n}})^{S'_{n}}) \\
S'_{2} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,1}} C_1^{S'_{1}}, \tilde{y}^{r_{1,1}} (\tfrac{C_2}{y_{1}})^{S'_{1}}) \\
&\vdots \\
S'_{\pi-1} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,\pi-2}} C_1^{S'_{\pi-2}}, \tilde{y}^{r_{1,\pi-2}} (\tfrac{C_2}{y_{\pi-2}})^{S'_{\pi-2}})
\end{aligned}$$

With the same sequence, we also start from computing $S''_{\pi+1}$ until $S^{\prime\prime}_{\pi-1}$.

$$\begin{aligned}
S''_{\pi+1} &= H_1(event, \mathbb{Y}, L, m, g^{t_2}, h^{t_2}) \\
S''_{\pi+2} &= H_1(event, \mathbb{Y}, L, m, g^{r_{2,\pi+1}} y_{\pi+1}^{S''_{\pi+1}}, h^{r_{2,\pi+1}} L^{S''_{\pi+1}}) \\
&\vdots \\
S''_{n} &= H_1(event, \mathbb{Y}, L, m, g^{r_{2,n-1}} y_{n-1}^{S''_{n-1}}, h^{r_{2,n-1}} L^{S''_{n-1}}) \\
S''_{1} &= H_1(event, \mathbb{Y}, L, m, g^{r_{2,n}} y_{n}^{S''_{n}}, h^{r_{2,n}} L^{S''_{n}}) \\
S''_{2} &= H_1(event, \mathbb{Y}, L, m, g^{r_{2,1}} y_{1}^{S''_{1}}, h^{r_{2,1}} L^{S''_{1}}) \\
&\vdots \\
S''_{\pi-1} &= H_1(event, \mathbb{Y}, L, m, g^{r_{2,\pi-2}} y_{\pi-2}^{S''_{\pi-2}}, h^{r_{2,\pi-2}} L^{S''_{\pi-2}})
\end{aligned}$$

For the verification, the verifier can simulate the process with the starting point S'_2 and S''_2, since S'_1 and S''_1 is given in the signature.

$$\begin{aligned}
S'_{2} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,1}} C_1^{S'_{1}}, \tilde{y}^{r_{1,1}} (\tfrac{C_2}{y_{1}})^{S'_{1}}) \\
&\vdots \\
S'_{\pi} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,\pi-1}} C_1^{S'_{\pi-1}}, \tilde{y}^{r_{1,\pi-1}} (\tfrac{C_2}{y_{\pi-1}})^{S'_{\pi-1}}) \\
S'_{\pi+1} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,\pi}} C_1^{S'_{\pi}}, \tilde{y}^{r_{1,\pi}} (\tfrac{C_2}{y_{\pi}})^{S'_{\pi}}) \\
S'_{\pi+2} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,\pi+1}} C_1^{S'_{\pi+1}}, \tilde{y}^{r_{1,\pi+1}} (\tfrac{C_2}{y_{\pi+1}})^{S'_{\pi+1}}) \\
&\vdots \\
S'_{n} &= H_1(event, \mathbb{Y}, L, m, g^{r_{1,n-1}} C_1^{S'_{n-1}}, \tilde{y}^{r_{1,n-1}} (\tfrac{C_2}{y_{n-1}})^{S'_{n-1}})
\end{aligned}$$

We verify $S'_1 \stackrel{?}{=} \bar{S}'_1 = H_1(event, \mathbb{Y}, L, m, g^{r_{1,n}} C_1^{S'_n}, \tilde{y}^{r_{1,n}} (\frac{C_2}{y_n})^{S'_n})$.
Note that the verification of $S'_{\pi+1}$ holds because:

$$\begin{aligned} g^{r_{1,\pi}} C_1^{S'_\pi} &= g^{t_1 - S'_\pi u} \cdot C_1^{S'_\pi} = g^{t_1 - S'_\pi u} \cdot g^{uS'_\pi} = g^{t_1} \\ \tilde{y}^{r_{1,\pi}} (\tfrac{C_2}{y_\pi})^{S'_\pi} &= \tilde{y}^{t_1 - S'_\pi u} (\tfrac{\tilde{y}^u y_\pi}{y_\pi})^{S'_\pi} = \tilde{y}^{t_1 - S'_\pi u} \cdot \tilde{y}^{uS'_\pi} = \tilde{y}^{t_1} \end{aligned}$$

Again, we start with verifying S''_2 in the same order.

$$\begin{gathered} S''_2 = H_1(event, \mathbb{Y}, L, m, g^{r_{2,1}} y_1^{S''_1}, h^{r_{2,1}} L^{S''_1}) \\ \vdots \\ S''_\pi = H_1(event, \mathbb{Y}, L, m, g^{r_{2,\pi-1}} y_{\pi-1}^{S''_{\pi-1}}, h^{r_{2,\pi-1}} L^{S''_{\pi-1}}) \\ S''_{\pi+1} = H_1(event, \mathbb{Y}, L, m, g^{r_{2,\pi}} y_\pi^{S''_\pi}, h^{r_{2,\pi}} L^{S''_\pi}) \\ S''_{\pi+2} = H_1(event, \mathbb{Y}, L, m, g^{r_{2,\pi+1}} y_{\pi+1}^{S''_{\pi+1}}, h^{r_{2,\pi+1}} L^{S''_{\pi+1}}) \\ \vdots \\ S''_n = H_1(event, \mathbb{Y}, L, m, g^{r_{2,n-1}} y_{n-1}^{S''_{n-1}}, h^{r_{2,n-1}} L^{S''_{n-1}}) \end{gathered}$$

We verify $S''_1 \stackrel{?}{=} \bar{S}''_1 = H_1(event, \mathbb{Y}, L, m, g^{r_{2,n}} y_n^{S''_n}, h^{r_{2,n}} L^{S''_n})$.
Similar to the verification of $S'_{\pi+1}$, we verify $S''_{\pi+1}$ by:

$$\begin{aligned} g^{r_{2,\pi}} y_\pi^{S''_\pi} &= g^{t_2 - S''_\pi x_\pi} \cdot g^{x_\pi S''_\pi} = g^{t_2} \\ h^{r_{2,\pi}} L^{S''_\pi} &= h^{t_2 - S''_\pi x_\pi} \cdot h^{x_\pi S''_\pi} = h^{t_2} \end{aligned}$$

Linking Correctness. Linking correctness is guaranteed if the signer computes the linking tag as follows:

$$h = H_2(event) \text{and } L = h^x.$$

Therefore, for the same event, the user can only compute the linking tag once.

Revoking Correctness. If the signer follows the protocol, the revocation can successfully recover signer's public with its secret by decrypting the cipher text in the following way:

$$y_\pi = \frac{C_2}{C_1^{\tilde{x}}},$$

where $\tilde{x}$ is revocation authority's private key and y_π is the real signer's public key.

4.5 Security Analysis

Theorem 1 (Existential Unforgeability). *RLRS scheme is existential unforgeable in the random oracle model if DLP is hard.*

Proof. The simulator $\mathcal{S}$ simulates the oracles as follows:

- *Random Oracle* H_1: $\mathcal{S}$ randomly picks $\alpha \in \mathbb{Z}_q$ and returns the value which has not been assigned.
- *Random Oracle* H_2: $\mathcal{S}$ randomly picks $k \in \mathbb{Z}_q$ and returns g^k.
- *Joining Oracle* $\mathcal{JO}$: Assume $\mathcal{A}$ can query $\mathcal{JO}$ at most n' times, where $n' \geq n$. $\mathcal{S}$ randomly chooses a subset $\mathcal{I}_n$ which contains n indexes. We assign these n indexes with $1, \ldots, n$, note that $\mathcal{S}$ dose not know any secret key corresponding to public keys with index 1 to n. We denote $n' - n$ indexes as $n + 1, \ldots, n'$, and $\mathcal{S}$ generates the public/private key pairs according to the algorithm for these $n' - n$ indexes. On the ith query, $\mathcal{S}$ returns the corresponding public key.
- *Corruption Oracle* $\mathcal{CO}$: For query input the public key pk which is an output of $\mathcal{JO}$, $\mathcal{S}$ first checks if it is corresponding to the subset $\mathcal{I}_n$. If yes, $\mathcal{S}$ halts. Otherwise, $\mathcal{S}$ returns the corresponding private key.
- *Signing Oracle* $\mathcal{SO}$: On input a signing query with an event description *event*, a group size n, a public key set $\mathbb{Y} = \{y_1, \ldots, y_n\}$, a signer's public key pk_π, where $\pi \in [1, n]$, a revocation authority's public key pk_{rev}, and a message M, $\mathcal{S}$ simulates as follows:
    - If the query of $H(event)$ has not been made, $\mathcal{S}$ queries H_2 on *event* and sets $h = H_2(event)$. Note that $\mathcal{S}$ knows k of h to the base g such that $h = g^k$.
    - If y_π is not corresponding to any element in $\mathcal{I}_n$, $\mathcal{S}$ knows the private key and computes the signature according to the algorithm. Otherwise, we let y_π be the πth index from $\mathcal{JO}$. $\mathcal{S}$ sets the linking tag $L = y_\pi^k$.
    - $\mathcal{S}$ randomly pick $u \in \mathbb{Z}_q$ and compute cipher text $C_1 = g^u$, $C_2 = \tilde{y}^u y_\pi$, and $C = \{C_1, C_2\}$, where $\tilde{y}$ is the revocation authority's public key and y_π is signer's public key.
    - $\mathcal{S}$ randomly chooses $S'_{\pi'}$ and $S''_{\pi'} \in \mathbb{Z}_q$, For $i = \pi', \ldots, n, 1, \ldots, \pi' - 1$, randomly picks $r_{1,i}$, $r_{2,i} \in \mathbb{Z}_q$ and computes:

$$S'_{i+1} = H_1(event, \mathbb{Y}, L, M, g^{r_{1,i}} C_1^{S'_i}, \tilde{y}^{r_{1,i}} (\tfrac{C_2}{y_i})^{S'_i}),$$
$$S''_{i+1} = H_1(event, \mathbb{Y}, L, M, g^{r_{2,i}} y_i^{S''_i}, h^{r_{2,i}} L^{S''_i}).$$

    $\mathcal{S}$ sets the oracle outcome:

$$H_1(event, \mathbb{Y}, L, M, g^{r_{1,\pi-1}} C_1^{S'_{\pi-1}}, \tilde{y}^{r_{1,i}} (\tfrac{C_2}{y_{\pi-1}})^{S'_{\pi-1}}) = S'_\pi,$$
$$H_1(event, \mathbb{Y}, L, M, g^{r_{2,\pi-1}} y_i^{S''_{\pi-1}}, h^{r_{2,\pi-1}} L^{S''_{\pi-1}}) = S''_\pi.$$

    If collision occurs, repeat this step.
    - $\mathcal{S}$ returns the signature $\sigma = (S'_1, S''_1, r_{1,1}, \ldots, r_{1,n}, r_{2,1}, \ldots, r_{2,n}, L, C)$. $\mathcal{A}$ cannot distinguish between $\mathcal{S}$'s simulation from REAL scenario.

For one successful simulation, suppose $\mathcal{A}$ forged

$$\sigma^{(1)} = (S_1^{(1)\prime}, S_1^{(1)\prime\prime}, r_{1,1}^{(1)}, \ldots, r_{1,n}^{(1)}, r_{2,1}^{(1)}, \ldots, r_{2,n}^{(1)}, L^{(1)}, C^{(1)})$$

on an *event*, and a set $\mathbb{Y}^{(1)}$ of $n^{(1)}$ public keys such that it is a subset of public keys corresponding to the indexes in $\mathcal{I}_n$. We let $n^{(1)} = n$ without loss of generality. By the assumption of random oracle model, $\mathcal{A}$ queries $H_2(event)$ which is denoted as h and queries $H_1(event, \mathbb{Y}^{(1)}, L, M, a_{1,i}, a_{2,i})$, $H_1(event, \mathbb{Y}^{(1)}, L, M, \bar{a}_{1,i}, \bar{a}_{2,i})$ for $i \in \{1, n\}$ where

$$a_{1,i} = g^{r_{1,i}^{(1)}} C_1^{S_i^{(1)\prime}} \text{ and } a_{2,i} = \tilde{y}^{r_{1,i}^{(1)}} (\tfrac{C_2}{y_i})^{S_i^{(1)\prime}},$$
$$\bar{a}_{1,i} = g^{r_{2,i}^{(1)}} y_i^{S_i^{(1)\prime\prime}} \text{ and } \bar{a}_{2,i} = h^{r_{2,i}^{(1)}} L^{S_i^{(1)\prime\prime}}.$$

Suppose $\mathcal{A}$ forges the signature after kth query to the oracles and $\mathcal{S}$ returns $S_1^{(1)\prime}$ and $S_1^{(1)\prime\prime}$. In the rewind simulation, suppose $\mathcal{S}$ first invokes $\mathcal{A}$ to get its output and its Turing Transcript $\mathcal{T}$. Then $\mathcal{S}$ rewinds $\mathcal{T}$ to get $\mathcal{T}'$ while $\mathcal{S}$ consistently answers kth query. That is, kth query is common in transcript $\mathcal{T}$ and $\mathcal{T}'$, denoted as:

$$H_1(event, \mathbb{Y}^{(1)}, L, M, a_{1,\pi}, a_{2,\pi}),$$
$$H_1(event, \mathbb{Y}^{(1)}, L, M, \bar{a}_{1,\pi}, \bar{a}_{2,\pi}).$$

$\mathcal{S}$ knows the value of $a_{1,\pi}, a_{2,\pi}, \bar{a}_{1,\pi}, \bar{a}_{2,\pi}$ at the time of the rewind. After $\mathcal{A}$ returns its output from the rewind simulation, $\mathcal{S}$ can solve the discrete logarithm problem of pk_π and $\tilde{y}$ in $\mathbb{Y}^{(1)}$ by computing following steps:

$$\begin{aligned} g^{r_{1,\pi}^{(1)}} C_1^{S_\pi^{(1)\prime}} &= g^{r_{1,\pi}^{(2)}} C_1^{S_\pi^{(2)\prime}} \\ \tilde{y}^{r_{1,\pi}^{(1)}} (\frac{C_2}{y_\pi})^{S_\pi^{(1)\prime}} &= \tilde{y}^{r_{1,\pi}^{(2)}} (\frac{C_2}{y_\pi})^{S_\pi^{(2)\prime}} \\ g^{r_{2,\pi}^{(1)}} y_\pi^{S_\pi^{(1)\prime\prime}} &= g^{r_{2,\pi}^{(2)}} y_\pi^{S_\pi^{(2)\prime\prime}} \\ h^{r_{2,\pi}^{(1)}} L^{S_\pi^{(1)\prime\prime}} &= h^{r_{2,\pi}^{(2)}} L^{S_\pi^{(2)\prime\prime}} \end{aligned}$$

That is

$$g^{r_{1,\pi}^{(1)}} g^{uS_\pi^{(1)\prime}} = g^{r_{1,\pi}^{(2)}} g^{uS_\pi^{(2)\prime}} \tag{1}$$
$$g^{\tilde{x}r_{1,\pi}^{(1)}} g^{\tilde{x}uS_\pi^{(1)\prime}} = g^{\tilde{x}r_{1,\pi}^{(2)}} g^{\tilde{x}uS_\pi^{(2)\prime}} \tag{2}$$
$$g^{r_{2,\pi}^{(1)}} g^{x_\pi S_\pi^{(1)\prime\prime}} = g^{r_{2,\pi}^{(2)}} g^{x_\pi S_\pi^{(2)\prime\prime}} \tag{3}$$
$$h^{r_{2,\pi}^{(1)}} h^{x_\pi S_\pi^{(1)\prime\prime}} = h^{r_{2,\pi}^{(2)}} h^{x_\pi S_\pi^{(2)\prime\prime}} \tag{4}$$

From Eq. (1), $\mathcal{S}$ derives $u = \frac{r_{1,\pi}^{(2)} - r_{1,\pi}^{(1)}}{S_\pi^{(1)\prime} - S_\pi^{(2)\prime}}$. Since $\mathcal{S}$ knows u, $\mathcal{S}$ can now derive $\tilde{x} = \frac{q-1}{r_{1,\pi}^{(1)} - r_{1,\pi}^{(2)} + u(S_\pi^{(1)\prime} - S_\pi^{(2)\prime})}$. From Eqs. (3) and (4), $\mathcal{S}$ can derive $x_\pi = \frac{r_{2,\pi}^{(2)} - r_{2,\pi}^{(1)}}{S_\pi^{(1)\prime\prime} - S_\pi^{(2)\prime\prime}}$. $\mathcal{S}$ solves DLP, contradiction occurs. According to the forking lemma, the successful rewind simulation is at least $\epsilon/4$, where ϵ is the probability that $\mathcal{A}$ successfully forges a signature. Therefore, the successful chance of $\mathcal{S}$ breaks DLP is at least $\epsilon/4$. □

Before we prove the anonymity of RLRS scheme, we provide a different definition of Decisional Diffie-Hellman (DDH) Assumption, which is used to derive the contradiction:

Definition 8 (A Different Decisional Diffie-Hellman (DDH) Assumption). *We define a different DDH assumption in $\mathbb{G}$ as follows: on input uniformly random $(l_0, l_1, l_2, l'_0, l'_1, l'_2) \in \mathbb{G}^6$, where the order of $|\mathbb{G}| = q$ for some prime number q. We set $\alpha_0 = g^{l_0}, \beta_0 = g^{l_1}, \gamma_0 = g^{l_2}, \alpha_1 = g^{l'_0}, \beta_1 = g^{l'_1}, \gamma_1 = g^{l'_0 l'_1}$. Any PPT adversary $\mathcal{A}$ takes a guess of $b \leftarrow \{0,1\}; (\alpha, \beta, \gamma) = (\alpha_b, \beta_b, \gamma_b)$. We say that $Pr[(\alpha, \beta, \gamma) = b] = \frac{1}{2} + \frac{1}{Q_2(\lambda)}$ where Q_2 is some polynomial and λ is the security parameter.*

Theorem 2 (Anonymity). *RLRS scheme is computational anonymity in the random oracle model if DDHP (Definition 8) is hard.*

Proof. For each $\mathcal{JO}$ query, a DL instance $y = g^x$ is returned for some randomly generated value x. Assume $\mathcal{A}$ can query $\mathcal{JO}$ at most n' times where $n' \geq n$. The challenge signature is created using the randomly picked public key in $\mathbb{Y}$. We assume $H_2(\mathbb{Y}) = \beta$. Since β is randomly generated, H_2 remains random. In order to simulate the process, $\mathcal{S}$ generates a challenge signature σ_π with signer $\pi \in [1, n]$, where π is randomly picked by $\mathcal{S}$ on the request from $\mathcal{A}$:

- $\mathcal{S}$ randomly picks $u \in \mathbb{Z}_q$ and computes ciphertext $C_1 = g^u$, $C_2 = \tilde{y}^u y_\pi$, and $C = \{C_1, C_2\}$, where $\tilde{y}$ is the revocation authority's public key.
- $\mathcal{S}$ sets $y_\pi = \alpha$ and then randomly picks $t_1, t_2 \in \mathbb{Z}_q^*$. $\mathcal{S}$ computes $\tilde{S}'_\pi = g^{t_1}$ and $\tilde{S}''_\pi = g^{t_2}$.
- For $i = \pi, \ldots, n, 1, \ldots, \pi - 1$, $\mathcal{S}$ randomly picks $r_{1,i}, r_{2,i} \in \mathbb{Z}_q$ and computes:

$$S'_{i+1} = H_1(event, \mathbb{Y}, \gamma, M, g^{r_{1,i}} C_1^{S'_i}, \tilde{y}^{r_{1,i}} (\tfrac{C_2}{y_i})^{S'_i})$$
$$S''_{i+1} = H_1(event, \mathbb{Y}, \gamma, M, g^{r_{2,i}} y_i^{S''_i}, h^{r_{2,i}} \gamma^{S''_i})$$

- $\mathcal{S}$ sets the oracle outcome

$$H_1(event, \mathbb{Y}, \gamma, M, g^{r_{1,\pi-1}} C_1^{S'_{\pi-1}}, \tilde{y}^{r_{1,\pi-1}} (\tfrac{C_2}{y_{\pi-1}})^{S'_{\pi-1}}) = S'_\pi$$
$$H_1(event, \mathbb{Y}, \gamma, M, g^{r_{2,\pi-1}} y_i^{S''_{\pi-1}}, h^{r_{2,\pi-1}} \gamma^{S''_{\pi-1}}) = S''_\pi$$

- $\sigma_\pi = (S'_1, S''_1, r_{1,1}, \ldots, r_{1,n}, r_{2,1}, \ldots, r_{2,n}, \gamma, C)$

$\mathcal{S}$ outputs σ_π to $\mathcal{A}$. $\mathcal{A}$ can query H_1, H_2 adaptively. Note that $H_2(\mathbb{Y}) = \beta$ and the output of

$$H_1(event, \mathbb{Y}, \gamma, M, g^{r_{1,i}} C_1^{S'_i}, \tilde{y}^{r_{1,i}} (\tfrac{C_2}{y_i})^{S'_i})$$
$$H_1(event, \mathbb{Y}, \gamma, M, g^{r_{2,i}} y_i^{S''_i}, h^{r_{2,i}} \gamma^{S''_i})$$

for $i \in \{1, \ldots, n\}$ are predetermined since $\mathcal{S}$ has queried these values.

Suppose $\mathcal{A}$ guesses the signer's index is $j \in [1, n]$ and returns j to $\mathcal{S}$. By convention, $\mathcal{A}$ returns 0 if it cannot identify a signer. $\mathcal{S}$ returns 1 if $j = \pi$; returns 0 if $j = 0$; and returns 1/0 with equal probability otherwise. Then

$$\begin{aligned}
&\Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 1] \\
&= \Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 1, \mathcal{A}(\sigma_\pi) = \pi] \\
&+ \Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 1, \mathcal{A}(\sigma_\pi) \neq \pi, \neq 0] \\
&\geq 1 \cdot (\frac{1}{n} + \frac{1}{Q(\lambda)}) + \frac{1}{2}(1 - \frac{1}{n} - \frac{1}{Q(\lambda)}) \\
&\geq \frac{1}{2} + \frac{1}{2n} + \frac{1}{2Q(\lambda)}
\end{aligned}$$

If $b = 0$, then all signers has equal probability to sign the signature from $\mathcal{A}$'s perspectives. Thus, $\mathcal{A}$ can do no better than random guessing.

$$\begin{aligned}
&\Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 0] \\
&= \Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 0, \mathcal{A}(\sigma_\pi) = \pi] \\
&+ \Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 0, \mathcal{A}(\sigma_\pi) \neq \pi] \\
&\geq 0 \cdot \frac{1}{n} + \frac{1}{2}(1 - \frac{1}{n})
\end{aligned}$$

Combining two probabilities, we have

$$\begin{aligned}
&\Pr[\mathcal{S}(\alpha, \beta, \gamma) = b] \\
&\geq \frac{1}{2}(\Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 1] \\
&+ \Pr[\mathcal{S}(\alpha, \beta, \gamma) = b | b = 0]) \\
&= \frac{1}{2} + \frac{1}{4Q(\lambda)}
\end{aligned}$$

Therefore, $\mathcal{S}$ solves DDHP with probability non-negligibly than $\frac{1}{2}$. Contradiction occurs. □

Theorem 3 (Linkability). *RLRS scheme is linkable in the random oracle model, if DLP is hard.*

Proof. In order to prove linkability of our RLRS scheme, we use the same oracle setting as the proof in Theorem 1 except we allow $\mathcal{S}$ to have at most one private key, say sk_π corresponding to two different public keys in ring group $\mathbb{Y}_i$ for $i = \{1, 2\}$. This private key is given to $\mathcal{A}$ during the query to the $\mathcal{CO}$, which is the only private key that $\mathcal{A}$ is allowed to have.
Suppose $\mathcal{A}$ produces two valid signature

$$\sigma^{(1)} = (S_1^{(1)\prime}, S_1^{(1)\prime\prime}, r_{1,1}^{(1)}, \ldots, r_{1,n_1}^{(1)}, r_{2,1}^{(1)}, \ldots, r_{2,n_1}^{(1)}, L^{(1)}, C^{(1)})$$
$$\sigma^{(2)} = (S_1^{(2)\prime}, S_1^{(2)\prime\prime}, r_{1,1}^{(2)}, \ldots, r_{1,n_2}^{(2)}, r_{2,1}^{(2)}, \ldots, r_{2,n_2}^{(2)}, L^{(2)}, C^{(2)})$$

where $L^{(1)} = H_2(event)^{x_\pi}$ and $L^{(2)} = H_2(event)^{x'_\pi}$ denote two linking tags of two signatures respectively. Note that the event description *event* is fixed during both runs. For $\sigma^{(1)}$, $\mathcal{S}$ rewinds the tape with a different value for H_1 to obtain another valid signature $\bar{\sigma}^{(1)}$. We can derive

$$x_\pi = \frac{\bar{r}^{(1)}_{2,\pi} - r^{(1)}_{2,\pi}}{S^{(1)''}_\pi - \bar{S}^{(1)''}_\pi}$$

where $L^{(1)} = h^{x_\pi}$ and $y_\pi = g^{x_\pi}$.
For the second rewind simulation for $\sigma^{(2)}$, $\mathcal{S}$ obtains with non-negligible probability of $\bar{\sigma}^{(2)}$. The similar derivation shows that

$$x'_\pi = \frac{\bar{r}^{(2)}_{2,\pi} - r^{(2)}_{2,\pi}}{S^{(2)''}_\pi - \bar{S}^{(2)''}_\pi}.$$

Therefore, $x_\pi = x'_\pi$ and $L^{(1)} = L^{(2)}$. Two signatures $(\sigma^{(1)}, \sigma^{(2)})$ are linked. $\mathcal{S}$ can break DLP if the rewind simulation is successful. □

Theorem 4 (Non-Slanderability). *RLRS scheme is non-slanderable in the random oracle model, if DLP is hard.*

Proof. The adversary $\mathcal{A}$ can query $\mathcal{CO}$ on any public key in $\mathbb{Y}$ except for signer's public key pk_π. $\mathcal{A}$ gives simulator $\mathcal{S}$ pk_π, an event description *event*, a message M, a set $\mathbb{Y}$ of n public keys, and a revocation authority's public key pk_{rev}, $\mathcal{S}$ generates a valid signature $\sigma = (\cdot, L)$ where L is the linking tag computed using sk_π. $\mathcal{A}$ can keep querying oracles with the restriction of submitting pk_π to $\mathcal{CO}$. Suppose $\mathcal{A}$ generates another valid signature $\sigma^* = (\cdot, L^*)$ which is not an output of $\mathcal{SO}$, and σ^* is linked to $\sigma = (\cdot, L)$. Therefore, $L^* = L$, which means:

$$L' = H_2(event)^{x^*_\pi} = L = H_2(event)^{x_\pi}$$

That is, $x_\pi = x^*_\pi$ which implies $\mathcal{A}$ knows the secret key sk_π corresponding to pk_π. This contradicts with the assumption that $\mathcal{A}$ cannot submit a query to $\mathcal{CO}$ to get the secret key of pk_π. □

Theorem 5 (Revocability). *RLRS scheme is revocable in the random oracle model if the construction is unforgeable.*

Proof. We use the same setting as the proof in Theorem 1 but the adversary $\mathcal{A}$ is able to get one private key denoted as $sk_\pi = x_\pi$ corresponding to $pk_\pi = y_\pi$ in $\mathbb{Y}$ from $\mathcal{CO}$. Since $\{pk_1, \ldots, pk_{\pi-1}, pk_{\pi+1}, \ldots, pk_n\}$ are $n-1$ discrete logarithm instances generated from fresh coin flips, $\mathcal{A}$ cannot find the corresponding secret keys under our assumption. For contradiction, suppose $\mathcal{A}$ successfully generates one valid signature:

$$\sigma = (S'_1, S''_1, r_{1,1}, \ldots, r_{1,n}, r_{2,1}, \ldots, r_{2,n}, L, C),$$

where $C = \{C_1, C_2\}$ and $C_1 = g^u, C_2 = \tilde{y}^u y_j$ for some randomly picked $u \in \mathbb{Z}_q$, and $\tilde{y}$ is revocation authority's public key. Since RLRS scheme is unforgeable, a valid signature is strictly generated by $sk_\pi = x_\pi$. There are two cases to break revocability of RLRS scheme:

- Case 1:
  1. $\mathcal{A}$ randomly picks t_1, $t_2 \in \mathbb{Z}_q$ and computes:
     (a) $S'_{j+1} = H_1(event, \mathbb{Y}, L, M, g^{t_1}, \tilde{y}^{t_1})$,
     (b) $S''_{j+1} = H_1(event, \mathbb{Y}, L, M, g^{t_2}, h^{t_2})$.
  2. For $i = j+1, \ldots, n, 1, \ldots, j-1$, $\mathcal{A}$ randomly picks $r_{1,i}, r_{2,i} \in \mathbb{Z}_q$ and computes:
     (a) $S'_{i+1} = H_1(event, \mathbb{Y}, L, M, g^{r_{1,i}} C_1^{S'_i}, \tilde{y}^{r_{1,i}} (\frac{C_2}{y_i})^{S'_i})$,
     (b) $S''_{i+1} = H_1(event, \mathbb{Y}, L, M, g^{r_{2,i}} y_i^{S''_i}, h^{r_{2,i}} L^{S''_i})$.

  Therefore, in order to close the ring, $\mathcal{A}$ has to know the secret key $x_j \neq x_\pi$ which contradicts with our assumption of $\mathcal{A}$ can only know one private key of ring member.
- Case 2:
  1. $\mathcal{A}$ randomly picks t_1, $t_2 \in \mathbb{Z}_q$ and computes:
     (a) $S'_{\pi+1} = H_1(event, \mathbb{Y}, L, M, g^{t_1}, \tilde{y}^{t_1})$,
     (b) $S''_{\pi+1} = H_1(event, \mathbb{Y}, L, M, g^{t_2}, h^{t_2})$.
  2. For $i = \pi+1, \ldots, n, 1, \ldots, \pi-1$, randomly picks $r_{1,i}, r_{2,i} \in \mathbb{Z}_q$ and computes:
     (a) $S'_{i+1} = H_1(event, \mathbb{Y}, L, M, g^{r_{1,i}} C_1^{S'_i}, \tilde{y}^{r_{1,i}} (\frac{C_2}{y_i})^{S'_i})$,
     (b) $S''_{i+1} = H_1(event, \mathbb{Y}, L, M, g^{r_{2,i}} y_i^{S''_i}, h^{r_{2,i}} L^{S''_i})$.
  3. $\mathcal{A}$ computes $r_{1,\pi} = t_1 - S'_\pi u$ and $r_{2_\pi} = t_2 - S''_\pi x_\pi$ to close the ring.

  However, The construction of σ will not pass the verification since an honest verifier will follow the protocol and computes as follows:

$$\begin{aligned} S'_{\pi+1} &= H_1(event, \mathbb{Y}, L, M, g^{r_{1,\pi}} C_1^{S'_\pi}, \tilde{y}^{r_{1,\pi}} (\frac{C_2}{y_\pi})^{S'_\pi}) \\ &\neq H_1(event, \mathbb{Y}, L, M, g^{t_1}, \tilde{y}^{t_1}) \end{aligned}$$

  This contradicts with our assumption that the signature σ is a valid signature.

The revocability of RLRS scheme is proved. □

5 Efficiency Analysis

This section compares the efficiency (i.e., computational cost and signature size) between our proposed scheme and *Revocable Ring Signature* [17] and *Traceable Ring Signature* [12]. We start by addressing some computational notions as follows:

- T_{exp}: The time for one exponentiation computation
- T_{mul}: The time for one modular multiplication computation
- T_{add}: The time for one modular addition computation
- T_{pair}: The time for one pairing computation
- T_h: The time for executing the one-way hash function
- n: The number of public keys in the ring
- ℓ: The number of revocation authority's public key

– λ: The length of the elements in $\mathbb{Z}_q$

From the comparison (Table 1), we can see that our scheme is a lot efficient in revocation phase than [17] and [12]. Since [17] allows a group of authorities to revoke the anonymity of the signer, the signature size also depends on the amount of authority's public keys. Besides, our RLRS scheme highly depends on the computational time of hash functions which could be faster than bilinear pairing functions in practice. Another contribution of our scheme is that we introduce the first ring signature scheme which enables mandatory revocability and linkability.

Table 1. Comparison between RLRS and [17] and [12]

Scheme	Sign	Verify	Revoke	Signature size	Mandatory revocability	Linkability
[17]	$2\ell T_{pair} + T_h + (2n+2)T_{mul} + (n+2)T_{add}$	$\ell T_{pair}+T_h+(2n+\ell)T_{mul} + (\ell n+2n)T_{add} + \ell T_{exp}$	T_{pair}(best case) nT_{pair}(worst case)	$(\ell + 2n + 2)\lambda$	✓	×
[12]	$3T_h + (5n+1)T_{exp} + (3n-2)T_{mul} + (n+1)T_{add}$	$3T_h + 3nT_{mul} + 5nT_{exp} + nT_{add}$	$4T_h + 2nT_{mul} + 2nT_{exp}$	$(2n + 1)\lambda$	×	✓
RLRS	$(2n+1)T_h + (8n-1)T_{exp} + (5n-1)T_{mul} + 2T_{add}$	$8nT_{exp} + 5nT_{mul} + 2nT_h$	$T_{mul} + T_{exp}$	$(2n + 5)\lambda$	✓	✓

6 Conclusion

In this paper, we extended [21] to construct a revocable and linkable ring signature (RLRS) scheme, which is the first ring signature scheme achieves mandatory revocability and linkability. In addition, our scheme is more efficient than [17] and [12] in terms of revocation time. We also provided a formal security proof of RLRS in random oracle model. In Appendix A, we further applied our RLRS scheme to design a revocable ring confidential transaction protocol.

There are several problems in our scheme that can be solved in future works such as:

1. Considering how to reduce signature size of RLRS scheme;
2. Considering how to construct a RLRS scheme with unconditional anonymity;
3. Providing a concrete security proof of *Revocable Ring Confidential Transaction*.

Appendix A. Revocable Ring Confidential Transaction

In Appendix A, we present a revocable ring confidential transaction protocol based on our RLRS scheme.

`Setup`(λ): Let $\mathbb{G}$ be a group of prime order q such that underlying discrete logarithm problem is intractable. Let $H_1 : \{0,1\}^* \to \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \to \mathbb{G}$ be two hash functions, and g, h are two generators in $\mathbb{G}$. The public parameters are $param = \{\mathbb{G}, g, h, q, H_1, H_2\}$

`KeyGen`$(param)$: Randomly choose $x \in \mathbb{Z}_q$ and compute $y = g^x \pmod q$. The secret key is $sk = x$ and the corresponding public key is $pk = y$

`Mint`(a, pk): Given an amount a and a coin address pk, randomly choose $r \in \mathbb{Z}_q$ and compute $C = h^a g^r \pmod q$, where the coin in address pk is denoted as $cn_{pk} = C$ and the corresponding coin key $ck = r$. The public information of an account is $act = (y, C)$ and the secrete information is $ask = (x, r)$.

`Spend`$(A_s, R, m, t, \mathbb{Y}, M, pk_{rev})$: On input the spender s's a set of m accounts A_s, a set of t output accounts R, a set of n group public keys $\mathbb{Y}$ such that $\mathbb{Y} = Y_1, \ldots, Y_n$, a transaction string M, and a revocation authority's public key $pk_{rev} = \tilde{y}$. The spender s can spend his/her m accounts to t output accounts by performing following steps:

1. The spender s parses $A_s = \{ack^{(k)}\}_{k\in[m]}$ into $\{(y_s^{(1)}, C_s^{(1)}), \ldots, (y_s^{(m)}, C_s^{(m)})\}$ and $K_s = \{ask^{(k)}\}_{k\in[m]}$ into $\{(x_s^{(1)}, r_s^{(1)}), \ldots, (x_s^{(m)}, r_s^{(m)})\}$ where $\{y_s^{(k)} = g^{x_s^{(k)}}\}_{k\in[m]}$ and $\{C_s^{(k)} = h^{a_s^{(k)}} g^{r_s^{(k)}}\}_{k\in[m]}$
2. Denote R as a set of output accounts where $R = \{pk_{out}^{(j)}\}_{j\in[t]}$, spender s randomly chooses $r_1, \ldots, r_t \in \mathbb{Z}_q$ and computes $C_{out}^{j} = h^{a_{out}^{(j)}} g^{r_j}$ for $j \in [t]$ where $a_{out}^{(1)} + \cdots + a_{out}^{(t)} = a_s^{(1)} + \cdots + a_s^{(m)}$
3. The spender s uses a public key encryption scheme $ENC_{pk}(\cdot)$ with public key pk to compute the cipher text $ctxt_j = ENC_{pk_{out}^{(j)}}(r_j)$ for $j \in [t]$ and send $\{ctxt_j\}_{j\in[t]}$ to the corresponding receiver's address.
4. In order to ensure that the amount of output coins equal to input coins, the spender s creates a new public key

$$y_s^{(m+1)} = \frac{\prod_{k=1}^{m}(y_s^{(k)} \cdot C_s^{(k)})}{\prod_{j=1}^{t} C_{out}^{(j)}}.$$

Since $a_{out}^{(1)} + \cdots + a_{out}^{(t)} = a_s^{(1)} + \cdots + a_s^{(m)}$, the $m+1$ public key is

$$y_s^{(m+1)} = g^{\sum_{k=1}^{m}(x_s^{(k)} + r_s^{(k)}) - \sum_{j=1}^{t} r_j} = g^{x_s^{(m+1)}}$$

such that $x_s^{(m+1)} = \sum_{k=1}^{m}(x_s^{(k)} + r_s^{(k)}) - \sum_{j=1}^{t} r_j$.

5. The spender s randomly picks $n-1$ group public keys from the blockchain, where each group contains $m+1$ public keys. We denote these public keys as:

$$\begin{gathered} Y_1 = \{y_1^{(1)}, \ldots, y_1^{(m+1)}\} \\ \vdots \\ Y_{s-1} = \{y_{s-1}^{(1)}, \ldots, y_{s-1}^{(m+1)}\} \\ Y_{s+1} = \{y_{s+1}^{(1)}, \ldots, y_{s+1}^{(m+1)}\} \\ \vdots \\ Y_n = \{y_n^{(1)}, \ldots, y_n^{(m+1)}\} \end{gathered}$$

   The spender's public key is further denoted as $Y_s = \{y_s^{(1)}, \ldots, y_s^{(m+1)}\}$.
6. Compute $m+1$ linking base as $h_k = H_2(y_s^{(k)})$ for $k \in [m+1]$ and the linking tags are $L_k = h_k^{x_s^{(k)}}$ for $k \in [m+1]$. We denote $L = \{L_1, \ldots, L_{m+1}\}$.
7. Encrypt the spender's $m+1$ public keys by using revocation authority's public key $pk_{rev} = \tilde{y}$ as follows:
   For $k = 1, \ldots, m+1$, randomly pick $u_1, \ldots, u_{m+1} \in \mathbb{Z}_q$ and compute:
   (a) $CT_1^{(k)} = g^{u_k}$,
   (b) $CT_2^{(k)} = \tilde{y}^{u_k} y_s^{(k)}$,
   (c) Combine the cipher text $CX_k = (CT_1^{(k)}, CT_2^{(k)})$.
8. For $k = 1, \ldots, m+1$, randomly pick $t_1^{(k)}$, $t_2^{(k)} \in \mathbb{Z}_q$ and compute:
   (a) $a_{1,s}^{(k)} = g^{t_1^{(k)}}$ and $a_{2,s}^{(k)} = (\frac{CT_2^{(k)}}{y_s^{(k)}})^{t_1^{(k)}}$,
   (b) $c'_{s+1} = H_1(\mathbb{Y}, L, M, \{a_{1,s}^{(1)}, a_{2,s}^{(1)}\}, \ldots, \{a_{1,s}^{(m+1)}, a_{2,s}^{(m+1)}\})$,
   (c) $\bar{a}_{1,s}^{(k)} = g^{t_2^{(k)}}$ and $\bar{a}_{2,s}^{(k)} = h_k^{t_2^{(k)}}$,
   (d) $c''_{s+1} = H_1(\mathbb{Y}, L, M, \{\bar{a}_{1,s}^{(1)}, \bar{a}_{2,s}^{(1)}\}, \ldots, \{\bar{a}_{1,s}^{(m+1)}, \bar{a}_{2,s}^{(m+1)}\})$.
9. Generate a linkable ring signature with a group of n public key vectors $\mathbb{Y} = \{Y_1, \ldots, Y_n\}$ using spender's $m+1$ secret keys $\{x_s^{(1)}, \ldots, x_s^{(m+1)}\}$ with $m+1$ linking tags $\{L_1, \ldots, L_{m+1}\}$ and $m+1$ ciphertexts $\{CX_1, \ldots, CX_{m+1}\}$ on some transaction string M as follows:
   (a) For $i = s+1, \ldots, n, 1, \ldots, s-1$, randomly pick $v_{1,i}^{(1)}, \ldots, v_{1,i}^{(m+1)}$ and $v_{2,i}^{(1)}, \ldots, v_{2,i}^{(m+1)} \in \mathbb{Z}_q$ and compute:
   (b) $a_{1,i}^{(k)} = g^{v_{1,i}^{(k)}} (CT_1^{(k)})^{c'_i}$ and $a_{2,i}^{(k)} = \tilde{y}^{v_{(1,i)}^{(k)}} (\frac{CT_2^{(k)}}{y_i^{(k)}})^{c'_i}$ for $k \in [m+1]$,
   (c) $c'_{i+1} = H_1(\mathbb{Y}, L, M, \{a_{1,i}^{(1)}, a_{2,i}^{(1)}, \}, \ldots, \{a_{1,i}^{(m+1)}, a_{2,i}^{(m+1)}\})$,
   (d) $\bar{a}_{1,i}^{(k)} = g^{v_{2,i}^{(k)}} (y_i^{(k)})^{c''_i}$ and $\bar{a}_{2,i}^{(k)} = h_k^{v_{2,i}^{(k)}} L_k^{(c''_i)}$ for $k \in [m+1]$,
   (e) $c''_{i+1} = H_1(\mathbb{Y}, L, M, \{\bar{a}_{1,i}^{(1)}, \bar{a}_{2,i}^{(1)}\}, \ldots, \{\bar{a}_{1,i}^{(m+1)}, \bar{a}_{2,i}^{(m+1)}\})$.
10. For $k = 1, \ldots, m+1$, compute:
   (a) $v_{1,s}^{(k)} = t_1^{(k)} - c'_s u_k$,
   (b) $v_{2,s}^{(k)} = t_2^{(k)} - c''_s x_s^{(k)}$.

11. The signature is $\sigma = (c'_1, c''_1, \{v^{(1)}_{1,1}, \ldots, v^{(m+1)}_{1,1}\}, \ldots, \{v^{(1)}_{1,n}, \ldots, v^{(m+1)}_{1,n}\}, \{v^{(1)}_{2,1}, \ldots, v^{(m+1)}_{2,1}\}, \ldots, \{v^{(1)}_{2,n}, \ldots, v^{(m+1)}_{2,n}\}, \{L_1, \ldots, L_{m+1}\}, \{CX_1, \ldots, CX_{m+1}\})$.

`Verify`$(n, \mathbb{Y}, \sigma, M)$: The algorithm takes the input of a group $\mathbb{Y} = \{Y_1, \ldots, Y_2\}$ of n groups of public keys, a signature σ, and a transaction string M. To verify a transaction, the verifier computes follows:

1. First parse the $m+1$ ciphertext $CX_k = \{CT^{(k)}_1, CT^{(k)}_2\}_{k \in [m+1]}$
2. For $i = 1, \ldots, n$, compute
   (a) $Z'^{(k)}_{1,i} = g^{v^{(k)}_{1,i}} (CT^{(k)}_1)^{c'_i}$ and $Z'^{(k)}_{2,i} = \tilde{y}^{v^{(k)}_{1,i}} (\frac{CT^{(k)}_2}{y^{(k)}_i})^{c'_i}$ for $k \in [m+1]$,
   (b) $c'_{i+1} = H_1(\mathbb{Y}, L, M, \{Z'^{(1)}_{1,i}, Z'^{(1)}_{2,i}\}, \ldots, \{Z'^{(m+1)}_{1,i}, Z'^{(m+1)}_{2,i}\})$ if $i \neq n$,
   (c) $Z''^{(k)}_{1,i} = g^{v^{(k)}_{2,i}} (y^{(k)}_i)^{c''_i}$ and $Z''^{(k)}_{2,i} = h_k^{v^{(k)}_{2,i}} (L_k)^{c''_i}$ for $k \in [m+1]$,
   (d) $c''_{i+1} = H_1(\mathbb{Y}, L, M, \{Z''^{(1)}_{1,i}, Z''^{(1)}_{2,i}\}, \ldots, \{Z''^{(m+1)}_{1,i}, Z''^{(m+1)}_{2,i}\})$ if $i \neq n$.
3. Check whether
   (a) $c'_1 \overset{?}{=} H_1(\mathbb{Y}, L, M, \{Z'^{(1)}_{1,n}, Z'^{(1)}_{2,n}\}, \ldots, \{Z'^{(m+1)}_{1,n}, Z'^{(m+1)}_{2,n}\})$,
   (b) $c''_1 \overset{?}{=} H_1(\mathbb{Y}, L, M, \{Z''^{(1)}_{1,n}, Z''^{(1)}_{2,n}\}, \ldots, \{Z''^{(m+1)}_{1,n}, Z''^{(m+1)}_{2,n}\})$.

`Revoke`$(n, \mathbb{Y}, sk_{rev}, \sigma)$: The algorithm receives a set $\mathbb{Y} = \{Y_1, \ldots, Y_n\}$ of n groups of public keys, a revocation authority's private key $sk_{rev} = \tilde{x}$, and a valid signature σ. The revocation authority with the knowledge of secret key $\tilde{x}$ corresponding to $\tilde{y}$ decrypts the $m+1$ ciphertexts to get $m+1$ public keys which belong to the real spender as follows

1. For $k = 1, \ldots, m+1$, parse $CT_k = (CT^{(k)}_1, CT^{(k)}_2)$.
2. Get the k-th public key $y'^{(k)}_s = CT^{(k)}_2 / CT^{(k)^{\tilde{x}}}_1$ and output all public keys into a set of $Y'_s = \{y'^{(1)}_s, \ldots, y'^{(m+1)}_s\}$.
3. There exists a public key vector $Y_s \in \mathbb{Y}$ such that $Y_s = Y'_s$.

References

1. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 415–432. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_26
2. Au, M.H., Chow, S.S.M., Susilo, W., Tsang, P.P.: Short linkable ring signatures revisited. In: Atzeni, A.S., Lioy, A. (eds.) EuroPKI 2006. LNCS, vol. 4043, pp. 101–115. Springer, Heidelberg (2006). https://doi.org/10.1007/11774716_9
3. Au, M.H., Liu, J.K., Susilo, W., Yuen, T.H.: Constant-size ID-based linkable and revocable-iff-linked ring signature. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 364–378. Springer, Heidelberg (2006). https://doi.org/10.1007/11941378_26
4. Au, M.H., Liu, J.K., Susilo, W., Yuen, T.H.: Certificate based (linkable) ring signature. In: Dawson, E., Wong, D.S. (eds.) ISPEC 2007. LNCS, vol. 4464, pp. 79–92. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72163-5_8

5. Au, M.H., Liu, J.K., Susilo, W., Yuen, T.H.: Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction. Theoret. Comput. Sci. **469**, 1–14 (2013)
6. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_26
7. Brenig, C., Accorsi, R., Müller, G.: Economic analysis of cryptocurrency backed money laundering. In: ECIS (2015)
8. Cayrel, P.-L., Lindner, R., Rückert, M., Silva, R.: A lattice-based threshold ring signature scheme. In: Abdalla, M., Barreto, P.S.L.M. (eds.) LATINCRYPT 2010. LNCS, vol. 6212, pp. 255–272. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14712-8_16
9. Changlun, Z., Yun, L., Dequan, H.: A new verifiable ring signature scheme based on Nyberg-Rueppel scheme. In: 2006 8th International Conference on Signal Processing, vol. 4. IEEE (2006)
10. FBI: Bitcoin virtual currency: Unique features present distinct challenges for deterring illicit activity. Intelligence Assessment (2012)
11. Fujisaki, E.: Sub-linear size traceable ring signatures without random oracles. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 393–415. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_25
12. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 181–200. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71677-8_13
13. Herranz, J., Sáez, G.: Forking lemmas for ring signature schemes. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 266–279. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-24582-7_20
14. Houben, R., Snyers, A.: Cryptocurrencies and blockchain: legal context and implications for financial crime, money laundering and tax evasion (2018)
15. Huang, X., et al.: Cost-effective authentic and anonymous data sharing with forward security. IEEE Trans. Comput. **64**(4), 971–983 (2015)
16. Lee, K.C., Wen, H.A., Hwang, T.: Convertible ring signature. IEE Proc.-Commun. **152**(4), 411–414 (2005)
17. Liu, D.Y., Liu, J.K., Mu, Y., Susilo, W., Wong, D.S.: Revocable ring signature. J. Comput. Sci. Technol. **22**(6), 785–794 (2007)
18. Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Linkable ring signature with unconditional anonymity. IEEE Trans. Knowl. Data Eng. **26**(1), 157–165 (2013)
19. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 325–335. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27800-9_28
20. Liu, J.K., Wong, D.S.: On the security models of (threshold) ring signature schemes. In: Park, C., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 204–217. Springer, Heidelberg (2005). https://doi.org/10.1007/11496618_16
21. Liu, J.K., Wong, D.S.: Linkable ring signatures: security models and new schemes. In: Gervasi, O., et al. (eds.) ICCSA 2005. LNCS, vol. 3481, pp. 614–623. Springer, Heidelberg (2005). https://doi.org/10.1007/11424826_65
22. Liu, J.K., Wong, D.S.: Solutions to key exposure problem in ring signature. IJ Netw. Secur. **6**(2), 170–180 (2008)

23. Liu, J.K., Yeo, S.L., Yap, W., Chow, S.S.M., Wong, D.S., Susilo, W.: Faulty instantiations of threshold ring signature from threshold proof-of-knowledge protocol. Comput. J. **59**(7), 945–954 (2016)
24. Liu, J.K., Yuen, T.H., Zhou, J.: Forward secure ring signature without random oracles. In: Qing, S., Susilo, W., Wang, G., Liu, D. (eds.) ICICS 2011. LNCS, vol. 7043, pp. 1–14. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25243-3_1
25. Lv, J., Wang, X.: Verifiable ring signature. In: Proceedings of DMS 2003-The 9th International Conference on Distribted Multimedia Systems, pp. 663–667 (2003)
26. Nakamoto, S., et al.: Bitcoin: a peer-to-peer electronic cash system (2008)
27. Noether, S.: Ring signature confidential transactions for monero. IACR Cryptology ePrint Archive 2015, 1098 (2015)
28. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
29. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret: theory and applications of ring signatures. In: Goldreich, O., Rosenberg, A.L., Selman, A.L. (eds.) Theoretical Computer Science. LNCS, vol. 3895, pp. 164–186. Springer, Heidelberg (2006). https://doi.org/10.1007/11685654_7
30. Tsang, P.P., Au, M.H., Liu, J.K., Susilo, W., Wong, D.S.: A suite of non-pairing ID-based threshold ring signature schemes with different levels of anonymity (extended abstract). In: Heng, S.-H., Kurosawa, K. (eds.) ProvSec 2010. LNCS, vol. 6402, pp. 166–183. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16280-0_11
31. Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. In: Deng, R.H., Bao, F., Pang, H.H., Zhou, J. (eds.) ISPEC 2005. LNCS, vol. 3439, pp. 48–60. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31979-5_5
32. Tsang, P.P., Wei, V.K., Chan, T.K., Au, M.H., Liu, J.K., Wong, D.S.: Separable linkable threshold ring signatures. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 384–398. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30556-9_30
33. Van Saberhagen, N.: Cryptonote v 2.0 (2013)
34. Xiong, H., Chen, Z., Li, F.: Bidder-anonymous english auction protocol based on revocable ring signature. Expert Syst. Appl. **39**(8), 7062–7066 (2012)
35. Yuen, T.H., Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Threshold ring signature without random oracles. In: ASIACCS 2011, pp. 261–267. ACM (2011)
36. Yuen, T.H., Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Efficient linkable and/or threshold ring signature without random oracles. Comput. J. **56**(4), 407–421 (2013). https://doi.org/10.1093/comjnl/bxs115