Traceable Ring Signatures from Group Actions: Logarithmic, Flexible, and Quantum Resistant *

Wei Wei, Min Luo(✉), Zijian Bao, Cong Peng, and Debiao He(✉)

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China
{weiwei_only,mluo,cpeng}@whu.edu.cn, {bao_zijian,hedebiao}@163.com

Abstract. Traceable ring signature (TRS) is a variation of ring signature, allowing to expose the users identity whenever he signs two different messages under the same tag. The accountable anonymity of TRS makes it widely used in many restrained anonymous applications, e.g., e-voting system, offline coupon service. Traditional TRS schemes are built on mathematical problems, which are believed to be easy to solve by quantum computers. While numerous post-quantum (traceable) ring signature schemes have been proposed so far, there has been no TRS scheme based on isogenies proposed. We construct two TRS schemes from group actions that can be instantiated with isogenies and lattices. The critical technique is to generate multiple tags for the message and design an OR sigma protocol to generate proofs for multiple tag sets, which provides traceability for the TRS scheme. The signature size can be expressed as $O(\log N)$, where N represents the ring size. Based on different instantiation parameters, our proposed scheme enables ring members to negotiate the signature size and signing time according to their specific requirements. Moreover, we prove the security of our scheme under the standard random oracle model.

Keywords: Traceable ring signature · Post-Quantum cryptography · Isogeny-based cryptography · Lattice-based cryptography · OR sigma protocol

1 Introduction

Ring Signature (RS) [30] allows the signer to sign a message on behalf of the group without revealing the signer's identity. Traceable ring signature (TRS) is a variation of the RS, if a signer produces two signatures for different messages under the same tag, then the identity of the signer can be extracted by the ring

* This work was supported by the Key Research and Development of Shandong Province under Grant 2021CXG010107; in part by the National Natural Science Foundation of China under Grant U21A20466, Grant 62172307, Grant 61972294 and Grant 61932016.

members, if signatures are for the same message, everyone can know that the two signatures were generated by the same signer. TRS limits the indubitable anonymity of ring signature, the tag in TRS consists of a group of members and a topic, the topic string refers to a social issue or voting. In many anonymous information systems such as e-voting [9] and offline coupon services [21], users are not expected to sign messages twice under the same tag, e.g., double-spending, multiple voting. The TRS scheme mitigates this dishonest behavior, and further protects the privacy of members, therefore, it becomes a powerful cryptographic tool in such systems.

The concept of TRS was proposed by Fujisaki and Suzuki [22] in 2007. Since then, several variant schemes [21,2] have been proposed to improve security or performance. However, these proposals are built on number theory, which can be solved by a large-scale quantum computer running Shors algorithm [33]. Consequently, a quantum-resistant ring signature and related variant schemes have drawn much attention over the past ten years. Lattice-based cryptography is one of the most promising candidates in post-quantum cryptography. In addition to resisting quantum attacks, it has the advantage of better performance. Isogeny-based cryptography was first proposed by [11,31], it is an extension and thorough study of classical elliptic curve cryptography. Compared with other post-quantum cryptography candidates, isogeny-based cryptography stands out for its comparatively shorter key sizes [24].

Recently, various schemes based on isogeny assumption have been proposed: signature schemes [5,14], ring signature scheme [4], revocable ring signature scheme [24] and accountable ring signature scheme [10]. Although these constructions and many post-quantum ring signature schemes (including variants) from lattices, code and symmetric cryptographic primitives have been proposed successively, an efficient isogeny-based TRS scheme has yet to be reported in the literature.

To fill this gap, we propose a general TRS scheme from group actions and instantiate the group action with isogenies and lattices. To the best of our knowledge, the isogeny-based instantiation is the first isogeny-based TRS scheme. It provides a smaller signature size than lattice-based instantiation. However, its significant overhead is signing time which caused by the complex operation of isogeny. Under different instantiation parameters, users can flexibly customize the signature size and signing time according to their requirements with different instantiation parameters. Note that the isogeny-based instantiation in this paper is from CSIDH [8]. The latest attack on isogenies proposed by Castryck and Decru [7] leads to key leakage in SIDH. This method has no impact on the security of primitives such as CSIDH and SQISign [14]. The efficiency of our TRS scheme is discussed in detail in Section 5.1.

1.1 Related work on Post Quantum Ring Signature

Lattice-based schemes. The first lattice-based ring signature was proposed by Libert et al. [26], which is non-linkable. Lu et al. [27] developed a general lattice-based (linkable) ring signature scheme from the short integer solution

(SIS) and NTRU assumptions. Esgin et al. [17] extended discrete logarithm proof techniques to the lattice setting in the one-out-of-many proofs, and designed a short ring signature scheme. They further optimized one out of many proofs, resulting in a smaller size ring signature scheme [16]. Then, they introduced a zero-knowledge proof and extractable commitment scheme from lattices, and designed an efficient RingCT protocol [18]. Feng et al. [19] constructed an efficient TRS scheme and instantiated the scheme with lattice-based building blocks: non-interactive zero-knowledge proof, collision-resistant hash function, and pseudorandom function. Nguyen et al. [28] proposed a unique ring signature (URS) scheme from lattices, which exploited a Merkle tree based accumulator as the building block.

Isogeny-based schemes. Beullens et al. [4] constructed an efficient (linkable) ring signature scheme and gave two concrete instances from isogenies and lattices. The signature size of their scheme scales with the number of ring members. Then, they constructed an accountable ring signature based on isogeny and lattice assumptions [3]. Through adding a valid ciphertext proof to their OR protocol and building an online extractable non-interactive zero-knowledge proof system, the signature size grows in $O(\log N)$. Chung et al. [10] proposed a group signature and accountable ring signature scheme based on the decisional CSIDH assumptions (D-CSIDH) and proved the security of scheme under the quantum random oracle model (QROM), the signature size grows in $O(N^2)$. Lai and Dobson [24] introduced the first revocable ring signature (RRS) scheme from isogenies, which is proved secure under the QROM, the signature size grows in $O(N \log(N))$.

Other post-quantum schemes. Branco and Mateus [6] built a post-quantum resistant TRS scheme based on the syndrome decoding problem. Their scheme was built on the Fiat-Shamir heuristic [20], they gave the security proof under the classic random oracle. Derler et al. [15] proposed the first sub-linear ring signature scheme from symmetric primitives. Scafuro and Zhang [32] introduced a one-time TRS scheme based on hash-function symmetric-key primitive.

Overall, the TRS schemes based on lattices, code and symmetric primitives have better performance, but the signature size of these schemes is large. Especially, The post-quantum TRS schemes that can be instantiated by isogenies and lattices are still in their infancy. This work proposes a general OR sigma protocol construction and constructs two TRS schemes from isogeny-based and lattice-based group action primitives. Both instantiations of the TRS schemes have a logarithmic communication complexity. Compared with other post-quantum schemes, the isogeny-based instantiation has the advantage of a smaller signature size, the lattice-based instantiation has a shorter signing time. Finally, we prove the security of our TRS scheme under the random oracle model.

1.2 Contribution

The major contribution of this work is the construction of a TRS from restricted group action in the random oracle model (ROM). As far as we know, this is the *first* TRS scheme that can be instantiated with isogenies.

- We propose a general TRS scheme based on restricted pair of group actions, OR sigma protocol and collision-resistant hash function. Furthermore, we instantiate the group action from *isogenies* and *lattices* to construct two TRS schemes.
- We design a special OR protocol for the TRS scheme. The core of our technique is to provide *traceability* by generating *tag sets* based on messages and user identities. Traceability will be possible by checking whether each tag/vector in the two tag/vector sets is equal. Further, we add OR proof for multiple tag sets to ensure validity.
- The scheme has *logarithmic communication complexity*. In order to reduce the signature size, we generate two Merkle trees and set the response as two paths in the tree in case challenge bit $\mathsf{chall} = 0$, when challenge bit $\mathsf{chall} = 1$, we send a seed as the response. Compared with other post-quantum TRS schemes, the signature size of our proposed TRS extends well with the ring size N, and our multiplicative factor on $\log N$ is much lower since the signatures mainly consist of two paths in a Merkle tree of depth $\log N$.
- The time and size of the signature can be *flexibly customized* from different instantiation parameters of the OR sigma protocol. The isogeny-based instantiation has a smaller signature size and lattice-based instantiation has better performance.

1.3 Overview of Results

In this paper, we will construct a general TRS scheme with generic security in terms of tag-linkability, anonymity and exculpability. With isogeny and lattice instantiation, it is resistant to attacks by quantum adversaries.

There is a (linkable) ring signature framework [4] that has been proposed, this general construction utilizes the admissible group action primitive and is built upon a OR sigma protocol. However, for the dishonest users who signs the same message or two different messages twice, it does not have the ability to track the identity of dishonest users. By adding multiple tags to the OR proof, we prove that the tag was generated by the signer while tracing the identity of the signer by comparing each tag in the tag set.

The security of TRS scheme from isogenies in Section 3 relies on the group action inverse problem and its equivalent hard problems [34], the security of lattice-based instantiation relies on the module short integer solution problem and module learning with errors problem [25], which are believed to be resistant to attacks by quantum adversaries. According to the experimental results, the smaller the value of Q is, the less time it takes for signature generation and verification. The minimum signature size is obtained when $K = 36$, where K and $Q - K$ are the number of 0 and the number of 1 in the challenge space.

Specifically, we have the 64/4096 bytes public key size under two different instantiations. The secret key of a user is 16 bytes. The signature size of our TRS scheme relies on the proof size of the OR protocol, which is logarithmic. It is approximately $\mathbf{2 \log N + 2.45/2 \log N + 55.37\ KB}$ under the specific

instantiation parameters and outperforms the post-quantum traceable signature size of [32,19,6].

Table 1 indicates the comparison results among the proposed TRS and other (traceable) ring signature schemes with respect to signature size complexity and supported attributes. Among the post-quantum signature schemes we investigated, these schemes support either linkability or traceability. In the case of supporting two properties, the signature size scales linearly with the ring size. Compared with lattice-based TRS schemes, our signature size under lattice-based instantiation is acceptable. Compared with isogeny-based linkable ring signature (without traceability), our scheme provides traceability with a smaller signature size. The details of the performance of TRS will be presented in Section 5.1.

Table 1: Comparison of our TRS with other (traceable) ring signature.

Schemes	**Signature size**	**Linkability**	**Traceability**	**Implementation**	**Hardness Assumption**
Alessandra[32]	$O(N)$	✓	✓	✓	NONE
Branco[6]	$O(N)$	✓	✓	×	SD[1]
Falafl[4]	$O(\log(N))$	✓	×	✓	MSIS[2], MLWE[3]
Feng H[19]	$O(\log(N))$	✓	✓	×	SIS [2], LWE[3]
MatRiCT[18]	$O(\log(N))$	×	×	✓	MSIS[2], MLWE[3]
Esgin[16]	$O(\log(N))$	×	×	✓	SIS[2], LWE[3]
Raptor[27]	$O(N)$	✓	×	✓	NTRU[4]
Calamari[4]	$O(\log(N))$	✓	×	✓	CSIDH[5]
CHH[10]	$O(N^2)$	×	✓	×	CSIDH[5]
KYM[24]	$O(N\log(N))$	×	✓	×	CSIDH[5]
This work	$O(\log(N))$	✓	✓	✓	MSIS[2], MLWE[3], CSIDH[5]

[1] SD: Syndrome Decoding
[2] SIS: Short Integer Solution, MSIS: Module Short Integer Solution
[3] LWE: Learning with Errors, MLWE: Module Learning with Errors
[4] NTRU: Number Theory Research Unit
[5] CSIDH: Commutative Supersingular Isogeny Diffie Hellman

2 Preliminaries

2.1 Traceable Ring Signature

In this section, we review the TRS scheme proposed by Fujisaki and Suzuki [22]. Assuming that N is the number of users in the ring, $\mathsf{PK} = (pk_1, \ldots, pk_N)$ is ring member's public keys set, $issue$ is a string representing the specific event of the signature and $L = (issue, \mathsf{PK})$ is the tag of the signature. A TRS scheme consists of five algorithms $\mathsf{TRS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Trace})$ described as follows:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$: The algorithm run by the trusted authority, which takes as input security parameter $\lambda \in \mathbb{N}$ and outputs public parameter pp.
- $(pk, sk) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$: The algorithm run by the ring member, which takes as input public parameter pp and returns public key pk and secret key sk.

- $\sigma \leftarrow \mathsf{Sign}(sk_\pi, L, M)$: The algorithm run by the ring member, which takes as input the secret key sk_π, a tag L and a message $M \in \{0,1\}^*$, and returns a signature σ.
- $\{\mathsf{accept}, \mathsf{reject}\} \leftarrow \mathsf{Verify}(L, M, \sigma)$: The algorithm run by the signature receiver, which takes as input the tag L, message M and signature σ, and returns either accept or reject.
- $\{\mathsf{indep}, \mathsf{linked}, pk\} \leftarrow \mathsf{Trace}(L, M, \sigma, M', \sigma')$: The algorithm run by the ring member or trusted authority, which takes as input two traceable ring signatures σ on message M and σ' on message M' with the same tag L, and returns a string that is either indep, linked or an element $pk \in \mathsf{PK}$. If $\sigma = \mathsf{Sign}(sk_\pi, L, M)$ and $\sigma' = \mathsf{Sign}(sk_{\pi'}, L, M')$, it holds that :

$$\mathsf{Trace}(L, M, \sigma, M', \sigma') = \begin{cases} \mathsf{indep} & \text{if } \pi \neq \pi', \\ \mathsf{linked} & \text{else if } M = M', \\ pk_i & \text{otherwise } (\pi = \pi' \wedge M \neq M'). \end{cases}$$

2.2 Security Model

A secure TRS scheme should satisfy the following properties: *correctness* and *security*. We use the security model in [21]. The security requirement for a TRS scheme has three: *tag-linkability*, *anonymity* and *exculpability*. The *unforgeability* can be derived from tag-linkability and exculpability [21].

Tag-linkability. Given N pairs of public and secret keys and N pairs of message-signature under tag L, the adversary can output $N+1$ valid pairs of message-signature. We define the tag-linkability game $\mathsf{Game}_{\mathcal{A}}^{\mathsf{tag-link}}$, if for all PPT adversaries $\mathcal{A}$, we have $Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{tag-link}}(\lambda) \leq \mathrm{nelg}(\lambda)$, then we say that TRS scheme is tag-linkable. The definition of $Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{tag-link}}$ is as follows:

$$\begin{aligned} Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{tag-link}}(\lambda) = \Pr[b_1 = \mathsf{accept} \wedge ... \wedge b_{N+1} = \mathsf{accept} \\ \wedge s_{\{1,2\}} = \mathsf{indep} \wedge ... \wedge s_{\{N+1,N\}} = \mathsf{indep}] \end{aligned}$$

$\mathsf{Game}_{\mathcal{A}}^{\mathsf{tag-link}}$: Tag-linkability game

1: $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2: $(L, M_i, \sigma_i) \leftarrow \mathcal{A}(\mathsf{pp})$, $\forall i \in 1, ..., N+1$
3: $b_i \leftarrow \mathsf{Verify}(L, M_i, \sigma_i)$, $\forall i \in 1, ..., N+1$
4: $s_{\{i,j\}} \leftarrow \mathsf{Trace}(L, M_i, \sigma_i, M_j, \sigma_j)$, $\forall i, j \in 1, ..., N+1 \wedge i \neq j$
5: **return:** $b_1, ..., b_{N+1}, s_{\{1,2\}}, ..., s_{\{N+1,N\}}$

Anonymity. It is infeasible for the adversary to know who signed the message. Considering the anonymity game $\mathsf{Game}_{\mathcal{A}}^{\mathsf{anon}}$, if for all PPT adversaries $\mathcal{A}$, we have $Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{anon}}(\lambda) \leq \mathrm{nelg}(\lambda)$, then we say that TRS scheme is anonymous. The definition of $Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{anon}}$ is as follows:

$$Adv_{\mathcal{A}}^{\mathsf{anon}}(\lambda) = \Pr[c = c'] - \frac{1}{2}$$

$\mathsf{Game}_{\mathcal{A}}^{\mathsf{anon}}$: Anonymity game

1: $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2: $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp}), i = 0, 1$
3: $c \leftarrow \{0, 1\}$
4: $c' \leftarrow \mathcal{A}^{\mathsf{Sign}(sk_c,\cdot),\mathsf{Sign}(sk_0,\cdot),\mathsf{Sign}(sk_1,\cdot)}(pk_0, pk_1)$
5: **return:** c'

Exculpability. This ensures that the adversary cannot construct two valid pairs of message-signature under tag L without knowing the secret key of the user. Consider the exculpability game $\mathsf{Game}_{\mathcal{A}}^{\mathsf{excu}}$, if for all PPT adversaries $\mathcal{A}$, we have $Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{excu}}(\lambda) \leq \mathrm{nelg}(\lambda)$, then we say that TRS scheme is exculpable, The definition of $Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{anon}}$ is as following:

$$Adv_{\mathcal{A},\mathsf{Game}}^{\mathsf{excu}}(\lambda) = \Pr[s = pk]$$

$\mathsf{Game}_{\mathcal{A}}^{\mathsf{excu}}$: Exculpability game

1: $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2: $(pk, sk) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
3: $(L, M_1, \sigma_1), (L, M_2, \sigma_2) \leftarrow \mathcal{A}^{\mathsf{Sign}(sk,\cdot)}(pk)$
4: $s \leftarrow \mathsf{Trace}(L, M_1, \sigma_1, M_2, \sigma_2)$
5: **return:** s

2.3 Restricted Pair of Group Actions

The restricted effective group actions (REGA) can be endowed with the properties: *one-wayness* (OW), *weak unpredictability* (wU), and *weak pseudo-randomness* (wPR) [1]. The special restricted pair of group actions used in this paper is called "admissible pair of group actions ", which is proposed by Beullens et al. [4].

Definition 1. *Given a finite commutative group $\boldsymbol{\mathcal{G}}$, $\boldsymbol{\mathcal{G}_1}$ and $\boldsymbol{\mathcal{G}_2}$ are two subsets of $\boldsymbol{\mathcal{G}}$. Let $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$ be two finite sets, $\boldsymbol{D_{\mathcal{S}}}$ and $\boldsymbol{D_{\mathcal{T}}}$ are distributions over two group actions $\star : \boldsymbol{\mathcal{G}} \times \boldsymbol{\mathcal{S}} \to \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{G}} \times \boldsymbol{\mathcal{T}} \to \boldsymbol{\mathcal{T}}$. For $(S_0, T_0) \in \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{T}}$, we say that $\mathsf{ResPGA} = (\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{G}_1}, \boldsymbol{\mathcal{G}_2}, \boldsymbol{D_{\mathcal{S}}}, \boldsymbol{D_{\mathcal{T}}})$ is a ξ-restricted pair of group actions if the following holds:*

1. ***Efficient Group Action**: For any $g \in \boldsymbol{\mathcal{G}_1} \cup \boldsymbol{\mathcal{G}_2}$ and $(S, T) \in \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{T}}$, it is efficient to compute $g \star S$ and $g \star T$, and uniquely represent the element of set $\boldsymbol{\mathcal{G}}$, $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$.*
2. ***Efficient Rejection Sampling**: For all $g \in \boldsymbol{\mathcal{G}_1}$, the intersection of all sets $\boldsymbol{\mathcal{G}_2} + g$ is large enough. Let $\boldsymbol{\mathcal{G}_3} = \bigcap_{g \in \boldsymbol{\mathcal{G}_1}} \boldsymbol{\mathcal{G}_2} + g$, then $|\boldsymbol{\mathcal{G}_3}| = \xi |\boldsymbol{\mathcal{G}_2}|$.*

3. ***Efficient Membership Testing****: It is efficient to verify that an element* $z \in \boldsymbol{\mathcal{G}_1}$*, or* $z \in \boldsymbol{\mathcal{G}_2}$*, or* $z \in \boldsymbol{\mathcal{G}_3}$*.*
4. *Given* $(g \star S_0, g \star T_0)$ *for any element* g *sampled from* $\boldsymbol{\mathcal{G}_1}$ *uniformly, it is indistinguishable from the elements* (S, T) *sampled from* $\boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{T}}$ *uniformly.*
5. *It is difficult to find two elements* $g, g' \in \boldsymbol{\mathcal{G}_2} + \boldsymbol{\mathcal{G}_3}$*, that satisfy* $g \star S_0 = g' \star S_0$ *and* $g \star T_0 \neq g' \star T_0$*.*
6. *For the element* g *sampled from set* $\boldsymbol{\mathcal{G}_1}$ *uniformly, given* $S = g \star S_0$ *and* $T = g \star T_0$*, it is difficult to find* $g' \in \boldsymbol{\mathcal{G}_2} + \boldsymbol{\mathcal{G}_3}$ *such that* $T = g' \star T_0$*.*

By instantiating the group action of ResPGA with isogenies and lattices, we achieve security against quantum adversaries as the underlying hard problems of isogenies and lattices. Further details on the underlying hard problems can be found in Appendix A and Appendix B.

2.4 Collision-Resistant Hash Function

In this paper, the cryptographic primitives used in the TRS scheme, such as pseudo-random number generators (PRG) and commitment schemes, are instantiated by the hash function. Specifically, we define five collision-resistant hash functions: $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, $\mathcal{H}_4$ and $\mathcal{H}_5$, where:

$$\begin{aligned}
&\mathcal{H}_1 : \{0,1\}^* \to \boldsymbol{\mathcal{G}_1}, \\
&\mathcal{H}_2 : \{0,1\}^* \to \{0,1\}^{2\lambda}, \\
&\mathcal{H}_3 : \{0,1\}^* \to C_K^Q, \\
&\mathcal{H}_4 : \{0,1\}^* \to \boldsymbol{\mathcal{G}_1^\dagger}, \mathcal{H}_5 : \{0,1\}^* \to \boldsymbol{\mathcal{G}_1^{\dagger\dagger}}^1.
\end{aligned}$$

For the Fiat-Shamir transform, we define a hash function $\mathcal{H}_3$ to produce an unbalanced challenge space C_K^Q, which is a set of string in $\{0,1\}^Q$, such that K bits are 0. The integers Q, K satisfying $\binom{Q}{K} \geq 2^\lambda$.

2.5 Sigma Protocol

A sigma protocol is a three-move public coin interactive protocol between the prover and verifier for the relation $R \subseteq X \times W$, where X is the space of statements and W is the space of witnesses. The sigma protocol under the random oracle includes the following three properties: *correctness*, *special honest-verifier zero-knowledge* and *special soundness* [12].

Definition 2. *A sigma protocol* Π_Σ *for the relation* $R \subseteq X \times W$ *consists of four PPT algorithms* $(P = (P_1, P_2), V = (V_1, V_2))$*, where* V_2 *is deterministic,* P_1 *and* P_2 *share the same information. Under the random oracle, the* Π_Σ *protocol has the three-move flow as follows:*

[1] Under the and lattice-based instantiations, $\boldsymbol{\mathcal{G}_1^\dagger}$ and $\boldsymbol{\mathcal{G}_1^{\dagger\dagger}}$ are two different subsets of $\boldsymbol{\mathcal{G}}$, the specific sets are shown is Section 5.

- $P_1(X, W) \to \mathsf{com}$. *The prover runs* $P_1(X, W)$ *on input* $(X, W) \in R$ *to generate a commitment* com*, and sends it to the verifier.*
- $V_1(\mathsf{com}) \to \mathsf{chall}$. *The verifier runs* $V_1(\mathsf{com})$ *on input* com *to generate a random challenge bit* chall*, and sends it to the prover.*
- $P_2(X, W, \mathsf{chall}) \to \mathsf{rsp}$. *The prover, after receiving* chall*, runs* $P_2(X, W, \mathsf{chall})$ *to obtain the response* rsp *and sends it to the verifier. In the case of* P_2 *termination, the prover sets* rsp *with symbol* $\perp$ *and sends it to the verifier.*
- $V_2(X, \mathsf{com}, \mathsf{chall}, \mathsf{rsp}) \to \{\mathsf{accept}, \mathsf{reject}\}$. *The verifier runs* $V_2(X, \mathsf{com}, \mathsf{chall}, \mathsf{rsp})$ *to check whether* X *is valid under the transcript* $(\mathsf{com}, \mathsf{chall}, \mathsf{rsp})$*, and outputs* accept *or* reject.

3 General Construction of Traceable Ring Signature

In this section, we will present a general construction of the TRS scheme from restricted pair of group actions. We first design a sigma protocol for the OR relation, then obtain a TRS scheme by applying the Fiat-Shamir transformation to the OR sigma protocol.

3.1 Our Special OR Sigma Protocol for Traceable Ring Signature

Our construction is based on a special OR sigma protocol, a variant of OR sigma protocol presented in [4] by adding the tag set. The essential technique of the protocol is to generate proofs for multiple tags and multiple public-secret key pairs, and the proof size of our sigma protocol grows logarithmically in N.

Let the relation $R \subset \boldsymbol{\mathcal{S}}^{N+1} \times \boldsymbol{\mathcal{T}}^{N+1} \times (\boldsymbol{\mathcal{G}_1}, \mathbb{Z}_N)$, where $R = \{(S_0, S_1, \ldots, S_N), (T_0, T_1, \ldots, T_N), (g, \pi), | g \in \boldsymbol{\mathcal{G}_1}, S_i \in \boldsymbol{\mathcal{S}}, T_i \in \boldsymbol{\mathcal{T}}, S_\pi = g \star S_0, T_\pi = g \star T_0\}$. We define a relation R' slightly wider than the relation R, and (R, R') satisfies $R \subseteq R'$, in addition to the relation R, R' contain two pairs of hash-preimage, and the extractor in special-soundness only extracts the witness of relation R'. Under the relation (R, R'), the OR sigma protocol is still useful as long as the relation (R, R') is sufficiently difficult.

$$R' = \left\{ (S_0, \ldots, S_N), (T_0, \ldots, T_N), w \;\middle|\; \begin{array}{c} S_i \in \boldsymbol{\mathcal{S}}, T_i \in \boldsymbol{\mathcal{T}} \text{ and} \\ w = (g, \pi) : g \in \boldsymbol{\mathcal{G}}_2 + \boldsymbol{\mathcal{G}}_3, S_\pi = g \star S_0, \\ T_\pi = g \star T_0 \text{ or} \\ w = (x, x') : x \neq x', \mathcal{H}_2(x) = \mathcal{H}_2(x') \end{array} \right\}$$

The difficulty of applying the accumulator to our construction is that each instance in the relation (R, R') is a pair of elements (S_i, T_i) rather than a single element. We solve this problem by hashing the commitments after applying the accumulator scheme to get the final commitments.

Based on the relation (R, R'), the OR sigma protocol proves that: 1) the prover owns a secret g and there exists $i \in N$, such that $g \star S_0 = S_i, g \star T_0 = T_i$,

without revealing the secret g and specific index i. Otherwise, 2) the prover owns a pair of collisions for $\mathcal{H}_2$.

Given $(S_0, S_1, \ldots, S_N)$, $(T_0, T_1, \ldots, T_N)$, we propose a sigma protocol $(P, V) = ((P_1, P_2), (V_1, V_2))$ under binary challenge space, for proving the ring member $\mathcal{P}_\pi$ possesses the secret key sk that satisfies relation (R, R'). A simple OR sigma protocol is shown in Figure 1 and the specific OR sigma construction as Figure 2.

Through repeating the basic OR sigma protocol under binary space, we construct a main OR sigma protocol under large challenge space and optimize it using three optimizations: unbalanced challenge space, Seed tree, and adding salt [4].

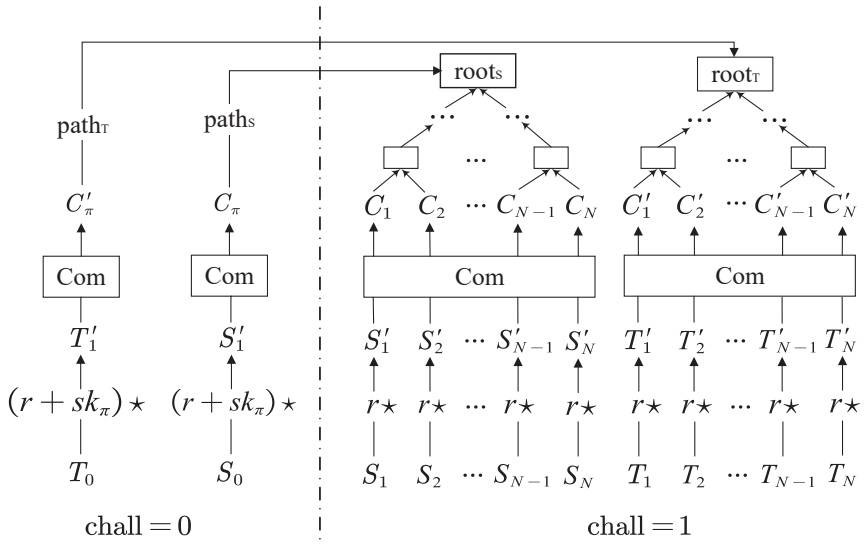

Fig. 1: The base OR sigma protocol, which proves that secret (sk_π, π) satisfies $sk_\pi \star S_0 = S_\pi$ and $sk_\pi \star T_0 = T_\pi$. If $\mathsf{chall} = 0$, then the commitments C_π and C'_π will be revealed, otherwise all commitments will be revealed.

We do not give specific security proof of OR sigma protocol over large challenge space, it is similar to the proof of [4]. Special attention should be paid to the special zero-knowledge of main OR sigma protocol. The probability of the adversary distinguishing a real protocol from a simulator in main OR sigma protocol is at most $4B/2^\lambda$.

3.2 Traceable Ring Signature from OR sigma protocol

We now present two concrete TRS schemes based on isogenies and lattices. The instantiation of both schemes is built on the aforementioned main OR sigma

- **Common Input:** A ring public key set $\mathsf{rpk} = (S_0, S_1, \ldots, S_N)$ and a tag set $\mathsf{TagSet} = (T_0, T_1, \ldots, T_N)$ are provided for the prover $\mathcal{P}_\pi$ and the verifier $\mathcal{V}$.
- **Private Input:** The prover $\mathcal{P}_\pi$ owns $sk \in \boldsymbol{\mathcal{G}_1}$ such that $(\mathsf{rpk}, \mathsf{TagSet}, (sk, \pi))$ is in the relation (R, R').

Commitment: $P_1(\mathsf{rpk}, \mathsf{TagSet}, \mathsf{seed})$

1. $\mathcal{P}_\pi$ generates $(r, (\mathsf{rnd}_1, ..., \mathsf{rnd}_N)) \leftarrow \mathrm{PRG}(\mathsf{seed})$
2. **for** all $i \in [N]$, $\mathcal{P}_\pi$ computes
3. $S_i' \leftarrow r \star S_i, T_i' \leftarrow r \star T_i$ ▷ Randomize rpk and TagSet
4. $C_i \leftarrow \mathcal{H}_2(S_i' || \mathsf{rnd}_i), C_i' \leftarrow \mathcal{H}_2(T_i' || \mathsf{rnd}_i)$ ▷ Create commitments C_i, C_i'
5. $(\mathsf{root_S}, \mathsf{tree_S}) \leftarrow \mathsf{MerkleTree}(C_1, ..., C_N)$
6. $(\mathsf{root_T}, \mathsf{tree_T}) \leftarrow \mathsf{MerkleTree}(C_1', ..., C_N')$
7. $\mathsf{com} \leftarrow \mathcal{H}_2(\mathsf{root_S}, \mathsf{root_T})$ ▷ Create the final commitment com
8. $\mathcal{P}_\pi$ sends com to verifier $\mathcal{V}$.

Challenge: $V_1(\mathsf{com})$

1. $\mathcal{V}$ samples challenge bit $\mathsf{chall} \leftarrow \{0, 1\}$ randomly
2. $\mathcal{V}$ sends the challenge bit chall to $\mathcal{P}_\pi$.

Response: $P_2((sk_\pi, \pi), \mathsf{chall}, \mathsf{seed})$

1. **if** $\mathsf{chall} = 0$, $\mathcal{P}_\pi$ computes $z = r + sk_\pi$, if $z \notin \boldsymbol{\mathcal{G}_3}$, abort, else $\mathcal{P}_\pi$ computes:
2. $\mathsf{path_S} \leftarrow \mathrm{getMerklePath}\,(\mathsf{tree_S}, \pi)$ ▷ Generate the path for rpk
3. $\mathsf{path_T} \leftarrow \mathrm{getMerklePath}\,(\mathsf{tree_T}, \pi)$ ▷ Generate the path for TagSet
4. $\mathsf{rsp} \leftarrow (z, \mathsf{path_S}, \mathsf{path_T}, \mathsf{rnd}_\pi)$
5. **else**
6. $\mathsf{rsp} \leftarrow \mathsf{seed}$
7. $\mathcal{P}_\pi$ sends rsp as response to $\mathcal{V}$.

Verification: $V_2(\mathsf{com}, \mathsf{chall}, \mathsf{rsp})$

1. **if** $\mathsf{chall} = 0$, $\mathcal{V}$ computes:
2. $(z, \mathsf{path_S}, \mathsf{path_T}, \mathsf{rnd}_\pi) \leftarrow \mathsf{rsp}$
3. $\hat{S} = z \star S_0, \hat{T} = z \star T_0$
4. $\hat{C} = \mathcal{H}_2(\hat{S} || \mathsf{rnd}_\pi), \hat{C}' = \mathcal{H}_2(\hat{T} || \mathsf{rnd}_\pi)$
5. $\widehat{\mathsf{root_S}} = \mathrm{ReconstructRoot}(\hat{C}, \mathsf{path_S})$ ▷ Recovery root for rpk
6. $\widehat{\mathsf{root_T}} = \mathrm{ReconstructRoot}(\hat{C}', \mathsf{path_T})$ ▷ Recovery root for TagSet
7. **if** $z \in \boldsymbol{\mathcal{G}_3} \wedge \mathcal{H}_2(\widehat{\mathsf{root_S}}, \widehat{\mathsf{root_T}}) = \mathsf{com}$ ▷ Verify the final commitment
8. $\mathcal{V}$ outputs accept.
9. **else** $\mathcal{V}$ outputs reject.
10. **else**
11. $\mathsf{seed} \leftarrow \mathsf{rsp}$
12. $\mathcal{V}$ computes $\mathsf{com} \leftarrow P_1(\mathsf{rpk}, \mathsf{TagSet}, \mathsf{seed})$
13. **if** $\mathsf{com} = \mathcal{H}_2(\mathsf{root_S}, \mathsf{root_T})$
14. $\mathcal{V}$ outputs accept.
15. **else** $\mathcal{V}$ outputs reject.

Fig. 2: The details of binary challenge space OR sigma protocol $(P, V) = ((P_1, P_2), (V_1, V_2))$, under a restricted pair of group actions $\mathsf{ResPGA} = (\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{G}_1}, \boldsymbol{\mathcal{G}_2}, \boldsymbol{D_\mathcal{S}}, \boldsymbol{D_\mathcal{T}})$ and $(S_0, T_0) \in \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{T}}$, PRG and hash function $\mathcal{H}_2$ is an instantiation of random oracle.

protocol, mainly by combining the design principles of Fujisaki and Suzuki [22] with restricted pair of group actions. Given the security parameters λ, the main OR sigma protocol (P_{main}, V_{main}), and the collision-resistant hash function $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$, $\mathcal{H}_4$ and $\mathcal{H}_5$, we construct two secure TRS schemes Π_{ISO} and Π_{LAT} by applying FS transform to main OR sigma protocol. Figure 3 illustrates two instantiations of the TRS scheme under lattice and isogeny. The general construction of the Setup and KeyGen in both schemes is as follows.

- $\mathsf{Setup}(1^\lambda)$: takes security parameter λ as input, selects $S_0 \leftarrow \boldsymbol{\mathcal{S}}$, and outputs public parameter $\mathsf{rpp} = S_0, \mathsf{ResPGA} = (\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{G}_1}, \boldsymbol{\mathcal{G}_2}, \boldsymbol{D_{\mathcal{S}}}, \boldsymbol{D_{\mathcal{T}}})$.
- $\mathsf{KeyGen}(\mathsf{rpp})$: takes public parameter as input, selects $g \leftarrow \boldsymbol{\mathcal{G}_1}, S = g \star S_0$, and outputs public key $pk_i = S$ and secret key $sk_i = g$.

To ensure that the secret key is embedded in the tag, and that the components in the signature do not disclose any information about the secret key and the identity of the member, we apply group action operations in the calculation of the tags and auxiliary parameters. Concretely, we set $T_0 = \mathcal{H}_1(L) \star S_0$ and the auxiliary parameter $T = (sk_\pi - \mathcal{H}_1(a, \pi)) \star T_0$ in the isogeny-based instantiation. The lattice-based instantiation is slightly different since the secret key is not sampled in the addition group, but in the polynomial ring, which supports multiplication and addition, we set $T_0 = \mathcal{H}_4(L), T_\pi = sk_\pi \star T_0$ and auxiliary parameter $\mathsf{aux} = \frac{(T_\pi - a)}{\pi}$.

As a result, the cost of the Trace algorithm differs between the two instantiations, with the isogeny-based instantiation requiring $2N$ group actions and the lattice-based instantiation requiring only $2N$ polynomial multiplications and additions.

Let $(S_0, S_1, \ldots, S_N)$ be the public parameter, each member $\mathcal{P}_i$ possesses a pair of public and secret keys: $sk_i = g, pk_i = g \star S_0$. Moreover, each member will generate N different tags $(T_0, T_1, \ldots, T_N)$ to link or trace signatures.

In order to generate the ring signature σ for the message $M \in \{0,1\}^*$ under the tag $L = (issue, \mathsf{rpk})$, the ring member $\mathcal{P}_\pi$ invokes $\mathsf{RSign}_{\{\mathsf{ISO},\mathsf{LAT}\}}(sk_\pi, L, M)$. The receivers verify signature σ on (L, M) by running $\mathsf{RVer}_{\{\mathsf{ISO},\mathsf{LAT}\}}(L, M, \sigma)$. To trace the relation between two valid signatures σ on M and σ' on M' with the same tag L, the ring members invoke $\mathsf{RTrace}_{\{\mathsf{ISO},\mathsf{LAT}\}}(L, M, \sigma, M', \sigma')$ which outputs linked, indep or pk_i.

4 Analysis of Our Traceable Ring Signature Scheme

In this section, we analyzed the correctness and security of TRS scheme under isogeny-based instantiation. A detailed proof of the lattice-based TRS scheme is presented in Appendix C.

4.1 Correctness

The correctness of our TRS scheme Π_{ISO} is composed of *completeness* and *traceability*. The completeness can be deduced from the correctness of the main OR

$\underline{\mathsf{RSign\_ISO}((sk_\pi, \pi), L, M)}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $T_0 = \mathcal{H}_1(L) \star S_0$, $a = \mathcal{H}_1(L, M)$
3. $T = (sk_\pi - \mathcal{H}_1(a, \pi)) \star T_0$
4. **for** all $i \in N$
5. $\quad k = \mathcal{H}_1(a, i)$
6. $\quad T_i = k \star T$
7. $\mathsf{TagSet} \leftarrow (T_0, T_1, ...T_N)$
8. $\mathsf{com} \leftarrow P^1_{main}(M, \mathsf{rpk}, \mathsf{TagSet})$
9. $\mathsf{chall} \leftarrow \mathcal{H}_3(M, \mathsf{rpk}, \mathsf{TagSet}, \mathsf{com})$
10. $\mathsf{rsp} \leftarrow P^2_{main}((sk_\pi, \pi), \mathsf{chall})$
11. **return** $\sigma = (T, \mathsf{com}, \mathsf{chall}, \mathsf{rsp})$.

$\underline{\mathsf{RVer\_ISO}(L, M, \sigma)}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $(T, \mathsf{com}, \mathsf{chall}, \mathsf{rsp}) \leftarrow \sigma$
3. $T_0 = \mathcal{H}_1(L) \star S_0$, $a = \mathcal{H}_1(L, M)$
4. **for** all $i \in N$
5. $\quad k = \mathcal{H}_1(a, i)$
6. $\quad T_i = k \star T$
7. $\mathsf{TagSet} \leftarrow (T_0, T_1, ...T_N)$
8. **if** $V^2_{main}(\mathsf{com}, \mathsf{chall}, \mathsf{rsp}) = \mathsf{accept}$ $\wedge \mathcal{H}_3(M, \mathsf{rpk}, \mathsf{TagSet}, \mathsf{com}) = \mathsf{chall}$
9. $\quad$ **return** accept.
10. **else return** reject.

$\underline{\mathsf{RTrace\_ISO}(L, M, \sigma, M', \sigma')}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $(T, \mathsf{com}, \mathsf{chall}, \mathsf{rsp}) \leftarrow \sigma$
3. $(T', \mathsf{com}', \mathsf{chall}', \mathsf{rsp}') \leftarrow \sigma'$
4. $a = \mathcal{H}_1(L, M), a' = \mathcal{H}_1(L, M')$
5. **for** all $i \in N$
6. $\quad k = \mathcal{H}_1(a, i), k' = \mathcal{H}_1(a', i)$
7. $\quad T_i = k \star T, T'_i = k' \star T'$
8. **if** for all $i \in [N]$, $T_i = T'_i$
9. $\quad$ **return** linked.
10. **if** only exist one $i \in [N]$, such that $T_i = T'_i$
11. $\quad$ **return** pk_i.
12. **else return** indep.

$\underline{\mathsf{RSign\_LAT}((sk_\pi, \pi), L, M)}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $T_0 = \mathcal{H}_4(L)$, $a = \mathcal{H}_5(L, M)$
3. $T_\pi = sk_\pi \star T_0$, $\mathsf{aux} = \frac{(T_\pi - a)}{\pi}$
4. **for** all $i \in N, i \neq \pi$
5. $\quad k = a + \mathsf{aux} \cdot i$
6. $\quad T_i = k \star T_0$
7. $\mathsf{TagSet} \leftarrow (T_0, T_1, ...T_N)$
8. $\mathsf{com} \leftarrow P^1_{main}(M, \mathsf{rpk}, \mathsf{TagSet})$
9. $\mathsf{chall} \leftarrow \mathcal{H}_3(M, \mathsf{rpk}, \mathsf{TagSet}, \mathsf{com})$
10. $\mathsf{rsp} \leftarrow P^2_{main}((sk_\pi, \pi), \mathsf{chall})$
11. **return** $\sigma = (\mathsf{aux}, \mathsf{com}, \mathsf{chall}, \mathsf{rsp})$.

$\underline{\mathsf{RVer\_LAT}(L, M, \sigma)}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $(\mathsf{aux}, \mathsf{com}, \mathsf{chall}, \mathsf{rsp}) \leftarrow \sigma$
3. $T_0 = \mathcal{H}_4(L) \star S_0$, $a = \mathcal{H}_5(L, M)$
4. **for** all $i \in N$
5. $\quad k = a + \mathsf{aux} \cdot i$
6. $\quad T_i = k \star T_0$
7. $\mathsf{TagSet} \leftarrow (T_0, T_1, ...T_N)$
8. **if** $V^2_{main}(\mathsf{com}, \mathsf{chall}, \mathsf{rsp}) = \mathsf{accept}$ $\wedge \mathcal{H}_3(M, \mathsf{rpk}, \mathsf{TagSet}, \mathsf{com}) = \mathsf{chall}$
9. $\quad$ **return** accept.
10. **else return** reject.

$\underline{\mathsf{RTrace\_LAT}(L, M, \sigma, M', \sigma')}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $(\mathsf{aux}, \mathsf{com}, \mathsf{chall}, \mathsf{rsp}) \leftarrow \sigma$
3. $(\mathsf{aux}', \mathsf{com}', \mathsf{chall}', \mathsf{rsp}') \leftarrow \sigma'$
4. $a = \mathcal{H}_5(L, M), a' = \mathcal{H}_5(L, M')$
5. **for** all $i \in N$
6. $\quad k_i = a + \mathsf{aux} \cdot i$
7. $\quad k'_i = a' + \mathsf{aux}' \cdot i$
8. **if** for all $i \in [N]$, $k_i = k'_i$
9. $\quad$ **return** linked.
10. **if** only exist one $i \in [N]$, such that $k_i = k'_i$
11. $\quad$ **return** pk_i.
12. **else return** indep.

Fig. 3: The isogeny-based TRS scheme (left column) and lattice-based TRS scheme (right column) from group actions.

sigma protocol. We demonstrate the traceability of the scheme in all possible situations under the same tag L.

- **Situation 1** ($\pi = \pi' \wedge M = M'$). From $\mathsf{RSign\_ISO}$ algorithm we have $T_0 = T_0'$, $T_\pi = T_\pi'$, $a = \mathcal{H}_1(L, M) = a' = \mathcal{H}_1(L, M')$, so that $T = T' = (sk_\pi - \mathcal{H}_1(a, \pi)) \star T_0$, for all $i \in N$, there is always $T_i = T_i'$. In this situation, the $\mathsf{TRace\_ISO}$ algorithm outputs linked.
- **Situation 2** ($\pi = \pi' \wedge M \neq M'$). When $x \neq x'$, $\mathcal{H}_1(x) = \mathcal{H}_1(x')$ means we find a collision of $\mathcal{H}_1$, hence we can conclude that $a \neq a'$ and $T \neq T'$ with overwhelming probability. For all $i \in N$, $i \neq \pi$, it holds that: $T_i = k \star T = [sk_\pi - \mathcal{H}_1(a, \pi) + \mathcal{H}_1(a, i)] \star T_0, T_i' = k' \star T' = [sk_\pi - \mathcal{H}_1(a', \pi) + \mathcal{H}_1(a', i)] \star T_0$. Given T_0, T and T', it's hard to find i such that $\mathcal{H}_1(a, i) - \mathcal{H}_1(a', i) = \mathcal{H}_1(a, \pi) - \mathcal{H}_1(a', \pi)$ since the collision resistance of $\mathcal{H}_1$, thus it holds that $T_i \neq T_i'$. For $i = \pi$, we have $T_\pi = sk_\pi \star T_0, T_\pi' = sk_\pi \star T_0$. Therefore $T_\pi = T_\pi'$ and the $\mathsf{RTrace\_ISO}$ algorithm outputs pk_π.
- **Situation 3** ($\pi \neq \pi'$). When $M = M'$, we have $a = a'$ and $T \neq T'$, thus $T_i \neq T_i'$ for all $i \in N, i \neq \pi$. When $M \neq M'$, we have $a \neq a'$, as proof in situation 2, it holds $T \neq T'$, given T, T' and a, a', it's hard to find i such that $\mathcal{H}_1(a, i) \star T = \mathcal{H}_1(a', i) \star T'$, therefore $T_i \neq T_i'$ for all $i \in N, i \neq \pi$. Consequently, it is difficult to find $i \in N$ such that $T_i = T_i'$ and the $\mathsf{RTrace\_ISO}$ algorithm outputs indep.

4.2 Security

Theorem 1. *If the OR sigma protocol is soundness and zero-knowledge, the hash function $\mathcal{H}_1, \mathcal{H}_2$ are collision-resistant, the* ResPGA *is a restricted pair of group actions, then our TRS scheme Π_{ISO} satisfies tag-linkability, anonymity and exculpability.*

Proof. **Tag-Linkability.** Conversely, assuming there exists an adversary $\mathcal{A}$ that makes at most B random oracle queries, the probability of $\mathcal{A}$ winning the game is not negligible. Then we demonstrate how to construct an algorithm $\mathcal{B}$ using $\mathcal{A}$, $\mathcal{B}$ breaks the Item 5 of ResPGA and collision resistance of $\mathcal{H}_2$. The simulation of $\mathcal{B}$ under random oracle is as follows (To distinguish from the T_i in the TagSet, we use T^i to represent the public tag in the signature at i-th query):

- $\mathsf{Sim}_{\mathcal{B}}^1$: The output of $\mathcal{A}$ in winning the tag-linkability game is the input of $\mathcal{B}$. Let $\{(L, (M_1, \sigma_1)), \ldots, (L, (M_{N+1}, \sigma_{N+1}))\}$ be the output of $\mathcal{A}$, $\sigma_i = (T^i, \mathsf{com}_i, \mathsf{chall}_i, \mathsf{rsp}_i)$, chall_i is the output of $\mathcal{H}_3$ on input $(M_i, \mathsf{rpk_i}, \mathsf{TagSet}_i, \mathsf{com}_i)$, $\mathcal{A}$ records these transcripts into list $\mathsf{List} = \{i, T^i, (\mathsf{com}_i, \mathsf{chall}_i, \mathsf{rsp}_i), M_i\}_{i \in [N+1]}$.
- $\mathsf{Sim}_{\mathcal{B}}^2$: $\mathcal{B}$ re-invokes $\mathcal{A}$ until $\mathcal{A}$ wins the tag-linkability game. Different from $\mathsf{Sim}_{\mathcal{B}}^1$, $\mathcal{B}$ controls the randomness used in the underlying main OR sigma protocol to generate signatures in each query. Specifically, for responding to the j-th signing query, the randomness used by the underlying main OR sigma protocol is the same as $\mathsf{Sim}_{\mathcal{B}}^1$ before q_i-th ($q_i \in [B]$) random

oracle, after that, $\mathcal{B}$ uses fresh randomness to interact with $\mathcal{A}$. Assuming that the output form of $\mathcal{A}$ is $\sigma' = (T^{j'}, (\mathsf{com}'_j, \mathsf{chall}'_j, \mathsf{rsp}'_j), M'_j, L)_{j\in[N+1]}$. If the signature σ'_j does not appear in the q_j-th random oracle query, the simulation of $\mathcal{B}$ starts again from $\mathsf{Sim}^2_{\mathcal{B}}$, otherwise, $\mathcal{B}$ updates list $\mathsf{List} = \mathsf{List} \cup \{j, T^{j'}, (\mathsf{com}'_j, \mathsf{chall}'_j, \mathsf{rsp}'_j), M'_j\}$, chall'_j is the result of the q_j-th random oracle query. Since we fix the randomness before q_j-th for both simulations, for all the entries in the list, we have $(M_j, T^j, \mathsf{com}_j) = (M'_j, T^{j'}, \mathsf{com}'_j)$.

- $\mathsf{Sim}^3_{\mathcal{B}}$: $\mathcal{B}$ extracts two entries from List that satisfy the above requirements, which one generated in $\mathsf{Sim}^1_{\mathcal{B}}$: $(T^j, (\mathsf{com}_j, \mathsf{chall}_j, \mathsf{rsp}_j), M_j)$ and the other generated in $\mathsf{Sim}^2_{\mathcal{B}}$: $(T^j, (\mathsf{com}_j, \mathsf{chall}'_j, \mathsf{rsp}'_j), M_j)$. If $\mathsf{chall}_j = \mathsf{chall}'_j$, $\mathcal{B}$ aborts the simulation, otherwise, $\mathcal{B}$ invokes the underlying main OR sigma protocol extraction algorithm, on input the statement $(\mathsf{rpk}_j, \mathsf{TageSet}_j)$ and two accepted transcripts $(\mathsf{com}_j, \mathsf{chall}_j, \mathsf{rsp}_j), (\mathsf{com}_j, \mathsf{chall}'_j, \mathsf{rsp}'_j)$, where TagSet_j is generated by the public input T^j, M_j and L, it outputs a witness w_j.
- $\mathsf{Sim}^4_{\mathcal{B}}$: For $j, j' \in [N+1]$, if there exists $w_j = (sk_j, \pi), w'_j = (sk'_j, \pi)$, then $\mathcal{B}$ outputs (sk_j, sk'_j), if w_j forms a collision of $\mathcal{H}_2$, $\mathcal{B}$ outputs w_j, otherwise, $\mathcal{B}$ aborts the simulation.

We can see that the witness $(w_j)_{j\in[N+1]}$ has the form: $w_j = (sk_j, \pi_j)$ such that $S_{\pi_j} = sk_j \star S_0, T_{\pi_j} = sk_j \star T_0$ or a collision of $\mathcal{H}_2$. If no collision occurs, then there must have two indexes $j', j \in [N+1]$ such that $w_j = (sk_j, \pi), w'_j = (sk'_j, \pi)$, since the pigeonhole principle and the conditions for winning the tag-linkability game, which indicate that $sk_j \star S_0 = sk'_j \star S_0$ but $sk_j \star T_0 \neq sk'_j \star T_0$, this violates the Item 5 of the restricted pair of group actions. Otherwise w_j is a collision of $\mathcal{H}_2$.

The running time of $\mathcal{B}$ is polynomial and the probability of aborting the simulation is negligible, detailed analysis refers to [4]. □

Anonymity. We demonstrated the anonymity of the scheme by building a sequence of games. The first game is the same as the original anonymity game, where $c = 0$. Similarly, the last game is exactly like the original anonymity game, where $c = 1$. We will prove that for any PPT adversary $\mathcal{A}$, the probability that he distinguishes between any two games is negligible. Let $Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_i}$ denotes the advantage of adversary $\mathcal{A}$ in Game_i.

- Game_1: This is an actual anonymity game $\mathsf{Game}^{\mathsf{anon}}_{\mathcal{A}}$ where $c = 0$, the adversary $\mathcal{A}$ is allowed to query $\mathsf{RSign}(sk_0, \cdot), \mathsf{RSign}(sk_1, \cdot)$ and $\mathsf{RSign}(sk_c, \cdot)$. Challenger $\mathcal{C}$ invokes the actual signing algorithm with the secret key to generate the signature, and outputs it as the result of a signing query.
- Game_2: In the second game, challenger $\mathcal{C}$ invokes the underlying main OR sigma protocol zero-knowledge simulation protocol Sim to respond to the signing query of $\mathcal{A}$ instead of running the real main OR sigma protocol. From the zero-knowledge property of the underlying main OR sigma protocol, the output distribution of Game_1 and Game_2 is indistinguishable, we have:

$$Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_1}(\lambda) \approx Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_2}(\lambda)$$

– Game_3: In the third game, the challenger $\mathcal{C}$ simulates N public-secret key pairs $\{(sk_i, pk_i)\}_{i\in[N]}$ instead of running the Setup algorithm, then $\mathcal{C}$ guesses that the pair of public keys $\{pk_j^*, pk_k^*\}$ sent by the adversary happens to be the j-th and k-th public keys, and generates two tag sets:

$$\mathsf{TagSet}_0 = \Big(T_0, T_j = sk_j^* \star T_0, \big(T_i = \big(sk_j^* + \mathcal{H}_1(a,i) - \mathcal{H}_1(a,j)\big) \star T_0\big)_{i\in[N]\setminus j}\Big)$$
$$\mathsf{TagSet}_1 = \Big(T_0, T_k = sk_k^* \star T_0, (T_i = (sk_k^* + \mathcal{H}_1(a,i) - \mathcal{H}_1(a,k)) \star T_0)_{i\in[N]\setminus k}\Big)$$

where $a = \mathcal{H}_1(L, M), T_0 = \mathcal{H}_1(L) \star S_0$. If the guess is incorrect, the challenger randomly samples a bit as the output of $\mathcal{A}$ and terminates the game. Otherwise, it responds to the signing queries using a pre-computed TagSet_0, TagSet_1 and $T^0 = (sk_j^* - \mathcal{H}_1(a,j)) \star T_0, T^1 = (sk_k^* - \mathcal{H}_1(a,k)) \star T_0$ at the beginning of Game_2. Since the probability of the challenger correctly guessing the two public keys is at most $1/N^2$, thus we have :

$$Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_3}(\lambda) \approx \frac{1}{N^2} Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_2}(\lambda)$$

– Game_4: Different from Game_3, the challenger $\mathcal{C}$ samples $\{i_0, i_1\} \leftarrow [N]$ and simulates $N-2$ public-secret key pairs $\{(sk_i, pk_i)\}_{i\in[N]\setminus\{i_0,i_1\}}$, then samples (E_0, E_1) uniformly from $\mathcal{T}$ and computes:

$$T^0 = (-\mathcal{H}_1(a, i_0)) \star E_0, \mathsf{TagSet}_0 = \Big(T_0, \big(T_i = \mathcal{H}_1(a,i) \star T^0\big)_{i\in[N]}\Big)$$
$$T^1 = (-\mathcal{H}_1(a, i_1)) \star E_1, \mathsf{TagSet}_1 = \Big(T_0, \big(T_i = \mathcal{H}_1(a,i) \star T^1\big)_{i\in[N]}\Big)$$

where $T_0 = \mathcal{H}_1(L) \star S_0$, the rest is the same as Game_3. Using the *weak-pseudrandom* of the restricted pair of group actions, we have:

$$(sk \star T_0 : sk \leftarrow \boldsymbol{\mathcal{G}_1}) \approx (E : E \leftarrow \boldsymbol{\mathcal{T}})$$

Thus Game_4 is computationally indistinguishable from Game_3: $Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_1}(\lambda) \approx Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_3}(\lambda)$. Now, the secret key is no longer used to generate the signature, i.e., the output of signing query does not reveal any information about the bit c in Game_4.

– Game_5: This is the same as an actual anonymous game, where $c = 1$.

It can be deduced from the above game sequence, there is no such adversary $\mathcal{A}$ that can distinguish any two games with a non-negligible probability, that is, the probability of the adversary winning the real anonymity game is negligible. □

Exculpability. If there exists an adversary $\mathcal{A}$ wins $\mathsf{Game}^{\mathsf{excu}}_{\mathcal{A}}$ with non-negligible probability, then we show how to construct an algorithm $\mathcal{B}$ from $\mathcal{A}$ that breaks the property Item 6 of restricted pair of group actions and collision resistance of $\mathcal{H}_1, \mathcal{H}_2$.

First, we simulate a game Game_1, it is indistinguishable from the real game $\mathsf{Game}^{\mathsf{excu}}_{\mathcal{A}}$. In Game_1, the challenger invokes the OR sigma protocol zero-knowledge simulation protocol Sim to simulate the signature, and generates N public-secret key pairs $\{(sk_i, pk_i)\}_{i\in[N]}$, let $T^i = (sk_i - \mathcal{H}_1(a,i)) \star T_0$, then challenger computes N tag sets $\mathsf{TagSet}_i = \big(T_{0,i} = \mathcal{H}_1(L) \star S_0, (T_{j,i} = \mathcal{H}_1(a,j) \star T^i)_{j\in[N]}\big)_{i\in[N]}$ before the game starts. If the signing query made by the adversary contains index i, then the challenger uses N public-secret key pairs, precomputed N tag sets TagSet_i and N elements T^i to generate the response. From the zero-knowledge of OR sigma protocol, indistinguishable of tag sets and collision resistance of $\mathcal{H}_1$, we have $Adv^{\mathsf{excu}}_{\mathcal{A},\mathsf{Game}_1} \approx Adv^{\mathsf{excu}}_{\mathcal{A},\mathsf{Game}^{\mathsf{excu}}}$. We show that when $\mathcal{A}$ wins Game_1, the simulation of $\mathcal{B}$ on the input (S,T) as follows:

- $\mathsf{Sim}^1_{\mathcal{B}}$: $\mathcal{B}$ randomly samples index $j \leftarrow [N]$, sets $pk_j = S, T^j = (-\mathcal{H}_1(a,j)) \star T$, and computes $\mathsf{TagSet}_j = \big(T_0 = \mathcal{H}_1(L) \star S_0, (T_i = \mathcal{H}_1(a,i) \star T^j)_{i\in[N]}\big)$, then generates the remaining $N-1$ public-secret key pairs $\{(pk_i, sk_i)\}_{i\in[N]\setminus j}$.
- $\mathsf{Sim}^2_{\mathcal{B}}$: $\mathcal{B}$ simulates the view of Game_1, since Game_1 does not contain any information of secret key. After interacting with $\mathcal{B}$, $\mathcal{A}$ outputs a forgery $(M, \mathsf{rpk}^*, \sigma^* = (T^*, \mathsf{com}^*, \mathsf{chall}^*, \mathsf{rsp}^*))$. To make sure the signature σ^* wins the Game_1, $\mathcal{B}$ must have responded to the signing query (i, M, rpk) with signature $\sigma = (T', \mathsf{com}', \mathsf{chall}', \mathsf{rsp}')$. If $i \neq j$, $\mathcal{B}$ terminates the simulation, otherwise, we have $T' = T$ and $T^* = T$, then $\mathcal{B}$ can extract witness w from the signature σ^* by rerunning $\mathcal{A}$. It is the same as what we have shown in the proof for tag-linkability.
- $\mathsf{Sim}^3_{\mathcal{B}}$: If w does not constitute a collision of $\mathcal{H}_2$, then we have $w = (sk, \pi)$ such that $sk \star T_0 = T_\pi$, $\mathcal{B}$ outputs $w = (sk, \pi)$, which violates the Item 6 of the underlying restricted pair of group actions, otherwise $\mathcal{B}$ outputs a pair of collisions of $\mathcal{H}_1$. □

5 Instantiations

Isogeny-based Instantiation. Theoretically, our TRS scheme can be instantiated with any CSIDH parameter set, e.g., CSIDH-512, CSIDH-1024 and CSIDH-1792 [4,13]. Nevertheless, taking into account that efficiency plays a vital role in the implementation, we implemented our TRS with the first group action parameter set proposed by Beullens et al. [5], which relies on the CSIDH group action proposed by [8], cSHAKE proposed by [23]. Let the ideal class group $\mathcal{C}\ell(\mathcal{O})$ be a cyclic group, and the order of generator $\mathfrak{g}$ is N. Then the group action $\star := \mathcal{C}\ell(\mathcal{O}) \times \mathcal{E}\ell\ell(\mathcal{O},\pi) \to \mathcal{E}\ell\ell(\mathcal{O},\pi)$ can be instantiated $(a, E) := \mathfrak{g}^a \star E$. We set $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{G}_1} = \boldsymbol{\mathcal{G}_2} = \mathcal{C}\ell(\mathcal{O}) = \mathbb{Z}_N$, $\xi = 1$, $\boldsymbol{\mathcal{T}} = \mathcal{E}\ell\ell(\mathcal{O},\pi)$, $\mathcal{S} = \mathcal{E}\ell\ell(\mathcal{O},\pi)$, and $S_0 = E_0, T_0 = \mathcal{H}_1(L) \star E_0$, where E_0 is the elliptic curve $y^2 = x^3 + x$ over $\mathbb{F}_p$.

Lattice-based Instantiation. Let $q = 5 \mod 4$, and let k, m be integers, n be a power of 2, B_1 and B_2 are integers such that $B_1 < B_2 < q$. Then the group action $\star := (\mathbf{s}, \mathbf{e}) \star \mathbf{t} \to \mathbf{As} + \mathbf{e} + \mathbf{t}$. We set $(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}) = (R_q^{k\times m} \times R_q^k \times R_q^m, R_q^m, R_q^m)$, $\boldsymbol{\mathcal{G}_1} = \{(\mathbf{s}, \mathbf{e_1}) \in \boldsymbol{\mathcal{G}} \mid \|\mathbf{s}\|_\infty, \|\mathbf{e_1}\|_\infty \leq B_1\}$, $\boldsymbol{\mathcal{G}_2} = \{(\mathbf{s}, \mathbf{e_2}) \in$

$\mathcal{G} \mid \|\mathbf{s}\|_\infty, \|\mathbf{e_2}\|_\infty \leq B_2\}$, where $R_q = \mathbb{Z}[X]/(q, X^n+1)$. For the collision-resistant hash function, let $\mathcal{H}_4 : \{0,1\}^* \to R_q^{k\times m}$, $\mathcal{H}_5 : \{0,1\}^* \to R_q^m$.

On Intel(R) Core(TM) i7-11700 CPU platform, we implemented the isogeny-based instantiation and tested the performance of lattice-based instantiation based on the number of group actions required. The source code is available at https://github.com/vivian-dev/TRS_ISO.

5.1 Implementation and Performance

Table 2 presents a detailed performance of our scheme, including signature size and time. Note that the addition of traceability necessitates additional group action computations, which impact the efficiency of our TRS scheme, especially for groups with large numbers of members. This explains why our scheme may be less efficient than the original linkable ring signature scheme proposed by Beullens et al. [4].

Table 2: Performance of proposed TRS scheme under different instantiations.

N			2^1	2^2	2^3	2^4	2^5	2^6
TRS_ISO	**Time**	KeyGen(ms)	39	39	39	39	39	39
		Sign(s)	3.37×10^1	6.63×10^1	1.31×10^2	2.64×10^2	5.23×10^2	1.07×10^3
		Verify(s)	3.20×10^1	6.02×10^1	1.16×10^2	2.31×10^2	4.64×10^2	9.22×10^2
	Size	Public Key(Byte)	64					
		Secret Key(Byte)	16					
		Signature(KB)	4.45	6.43	8.25	10.09	12.06	13.87
TRS_LAT (NIST 2)	**Time**	KeyGen(ms)	0.2	0.2	0.2	0.2	0.2	0.2
		Sign(ms)	68.5	101.3	131.4	230.8	390.3	764.0
		Verify(ms)	27.4	34.9	50.3	81.1	144.0	265.4
	Size	Public Key(Byte)	4096					
		Secret Key(Byte)	16					
		Signature(KB)	56.37	57.37	58.37	59.37	60.37	61.37

Furthermore, we compare our TRS with existing post-quantum (traceable) ring signature schemes. The results are shown in Table 3. The signature size of our lattice-based TRS scheme outperforms the size of [6,17,3,19]. When the ring members are small, the signature size of [32] and [27] is advantageous. However, once the ring members exceed 2^6, the signature size of [32] and [27] becomes significantly larger than in our lattice-based TRS scheme. Compared with the original scheme [4] and [3], Our isogeny-based instantiation has a larger signature size. This could be attributed to our scheme generating a tag set for each ring member.

Table 3: Comparison of public key size, secret key size and signature size of our TRS scheme with post-quantum (traceable) ring signature schemes.

Schemes		Public key (KB)	Secret key (KB)	Signature size (KB)				Security Level
				2^1	2^3	2^6	2^{10}	
Calamari[4]		64 (Byte)	16 (Byte)	3.5	5.4	8.2	10	*
Beullens_ISO[3]		64 (Byte)	16 (Byte)	3.6	-	6.6	9.0	*
Raptor[27]		0.9	9.1	2.6	11	82	1331.2	100bits
Beullens_LAT[3]		5120 (Byte)	16 (Byte)	124	-	126	129	NIST 2
Falafl[4]		5120 (Byte)	16 (Byte)	49	50	52	55	NIST 2
Branco[6]		1577	0.5	-	1920	1536	245(MB)	NIST 5
Alessandra[32]		6	4	4	16	131	1024	NIST 5
Feng H[19]		-	-	135.1	136.3	138.2	140.7	NIST 5
Esign[17]		≤ 8.33	≤ 0.83	-	-	774	1021	NIST 5
this work	**ISO**	**64 (Byte)**	**16 (Byte)**	**4.5**	**8.3**	**13.9**	**22.2**	*****
	LAT	**4096 (Byte)**	**16 (Byte)**	**56.3**	**58.3**	**61.3**	**65.3**	**NIST 2**
	LAT	**6144 (Byte)**	**16 (Byte)**	**74.3**	**76.3**	**79.3**	**83.3**	**NIST 5**

* : 128bits classical security and 60bits quantum security [29]

With isogeny-based instantiation, our TRS scheme offers flexible customization of signature size and time for signature generation and verification. We adopt the number of group actions to represent the time spent on signature generation and verification. When the parameter satisfies $\binom{Q}{K}$>128, it can be concluded from Figure 4 (right) that the smaller the value of Q is, the less time it takes for signature generation and verification.

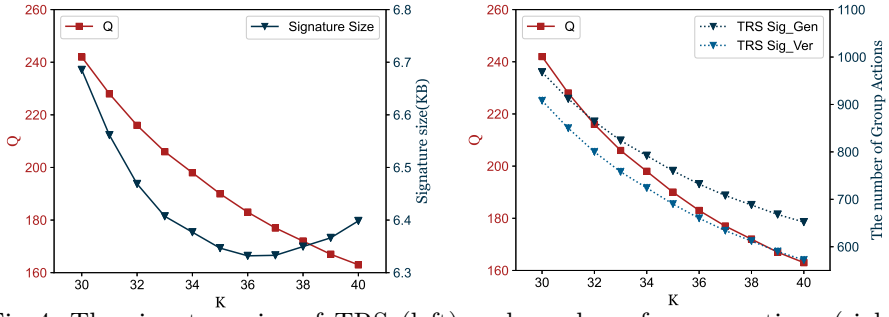

Fig. 4: The signature size of TRS (left) and number of group actions (right) under different (Q, K).

To obtain the threshold of K, we conducted experiments with $N = 2$ to analyze the relationship between signature size and the value of K. The results are presented on the left side of Figure 4. It can be seen that the minimum signature size is obtained when $K = 36$. We can conclude that if the user prioritizes

minimizing the time spent on the signature, they can choose a smaller value for K. On the other hand, if the user prioritizes minimizing the signature size, they can choose a specific value ($K = 36$) to achieve a smaller signature.

With the same security level, we set constant rounds for OR sigma protocol to observe the effect of different values of K on the signature size and signing time. The results are shown on the right in Figure 5, it can be observed that as the value of K increases, the time required for signature verification decreases while the signature size increases. Users can customize the value of K according to their specific requirements. In addition, we provide three optimal (Q, K) pairs under different ring sizes in Figure 5 (left), which result in smaller signature sizes and can be selected by users.

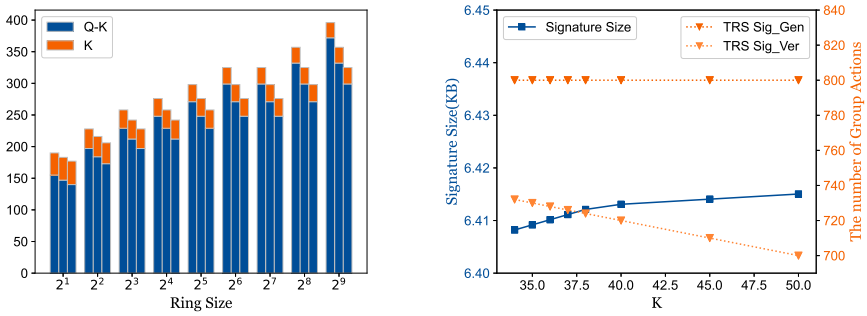

Fig. 5: Three superior (Q, K) pairs under different ring sizes (left) and the relationship among K, signature size and time spent on signature generation and verification (right) under the same Q.

6 Conclusion

This work presents a quantum-resistant TRS scheme from group action. First, we construct a special OR sigma protocol based on the restricted group action, which can be instantiated by isogenies and lattices. Then, using Fiat-Shamir transform to the OR sigma protocol, we derive two concrete TRS schemes. The core of our technique is to construct an OR proof for multiple tags and public key set. Under the random oracle model, we further prove the security of our TRS scheme in terms of tag-linkability, anonymity and exculpability. Finally, we give two TRS implementations from CSIDH-512/CSI-Fish, Dilithium and cSHAKE, the results show that our TRS is competitive in signature size and performance compared with other post-quantum (traceable) ring signature schemes.

References

1. Alamati, N., De Feo, L., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: Advances in Cryptology–ASIACRYPT 2020: 26th

International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26. pp. 411–439. Springer (2020)
2. 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. Theoretical Computer Science **469**, 1–14 (2013)
3. Beullens, W., Dobson, S., Katsumata, S., Lai, Y.F., Pintore, F.: Group signatures and more from isogenies and lattices: generic, simple, and efficient. Designs, Codes and Cryptography pp. 1–60 (2023)
4. Beullens, W., Katsumata, S., Pintore, F.: Calamari and falafl: logarithmic (linkable) ring signatures from isogenies and lattices. In: Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II. pp. 464–492. Springer (2020)
5. Beullens, W., Kleinjung, T., Vercauteren, F.: Csi-fish: efficient isogeny based signatures through class group computations. In: Advances in Cryptology–ASIACRYPT 2019: 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part I. pp. 227–247. Springer (2019)
6. Branco, P., Mateus, P.: A traceable ring signature scheme based on coding theory. In: Post-Quantum Cryptography: 10th International Conference, PQCrypto 2019, Chongqing, China, May 8–10, 2019 Revised Selected Papers 10. pp. 387–403. Springer (2019)
7. Castryck, W., Decru, T.: An efficient key recovery attack on sidh. In: Advances in Cryptology–EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V. pp. 423–447. Springer (2023)
8. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: Csidh: an efficient post-quantum commutative group action. In: Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24. pp. 395–427. Springer (2018)
9. Chow, S.S., Liu, J.K., Wong, D.S.: Robust receipt-free election system with ballot secrecy and verifiability. In: NDSS. vol. 8, pp. 81–94 (2008)
10. Chung, K.M., Hsieh, Y.C., Huang, M.Y., Huang, Y.H., Lange, T., Yang, B.Y.: Group signatures and accountable ring signatures from isogeny-based assumptions. arXiv e-prints pp. arXiv–2110 (2021)
11. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive (2006)
12. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Advances in CryptologyCRYPTO94: 14th Annual International Cryptology Conference Santa Barbara, California, USA August 21–25, 1994 Proceedings. pp. 174–187. Springer (2001)
13. De Feo, L., Galbraith, S.D.: Seasign: compact isogeny signatures from class group actions. In: Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38. pp. 759–789. Springer (2019)
14. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: Sqisign: compact post-quantum signatures from quaternions and isogenies. In: Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application

of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26. pp. 64–93. Springer (2020)
15. Derler, D., Ramacher, S., Slamanig, D.: Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In: Post-Quantum Cryptography: 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings 9. pp. 419–440. Springer (2018)
16. Esgin, M.F., Steinfeld, R., Liu, J.K., Liu, D.: Lattice-based zero-knowledge proofs: new techniques for shorter and faster constructions and applications. In: Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I. pp. 115–146. Springer (2019)
17. Esgin, M.F., Steinfeld, R., Sakzad, A., Liu, J.K., Liu, D.: Short lattice-based one-out-of-many proofs and applications to ring signatures. In: Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17. pp. 67–88. Springer (2019)
18. Esgin, M.F., Zhao, R.K., Steinfeld, R., Liu, J.K., Liu, D.: Matrict: efficient, scalable and post-quantum blockchain confidential transactions protocol. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 567–584 (2019)
19. Feng, H., Liu, J., Wu, Q., Li, Y.N.: Traceable ring signatures with post-quantum security. In: Topics in Cryptology–CT-RSA 2020: The Cryptographers Track at the RSA Conference 2020, San Francisco, CA, USA, February 24–28, 2020, Proceedings. pp. 442–468. Springer (2020)
20. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Advances in CryptologyCRYPTO86: Proceedings 6. pp. 186–194. Springer (1987)
21. Fujisaki, E.: Sub-linear size traceable ring signatures without random oracles. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences **95**(1), 151–166 (2012)
22. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Public Key Cryptography–PKC 2007: 10th International Conference on Practice and Theory in Public-Key Cryptography Beijing, China, April 16-20, 2007. Proceedings 10. pp. 181–200. Springer (2007)
23. Kelsey, J., Chang, S.j., Perlner, R.: Sha-3 derived functions: cshake, kmac, tuplehash, and parallelhash. NIST special publication **800**, 185 (2016)
24. Lai, Y.F., Dobson, S.: Collusion resistant revocable ring signatures and group signatures from hard homogeneous spaces. Cryptology ePrint Archive (2021)
25. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography **75**(3), 565–599 (2015)
26. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35. pp. 1–31. Springer (2016)
27. Lu, X., Au, M.H., Zhang, Z.: Raptor: a practical lattice-based (linkable) ring signature. In: Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17. pp. 110–130. Springer (2019)

28. Nguyen, T.N., Ta, A.T., Le, H.Q., Duong, D.H., Susilo, W., Guo, F., Fukushima, K., Kiyomoto, S.: Efficient unique ring signatures from lattices. In: Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part II. pp. 447–466. Springer (2022)
29. Peikert, C.: He gives c-sieves on the csidh. In: Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30. pp. 463–492. Springer (2020)
30. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Advances in CryptologyASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7. pp. 552–565. Springer (2001)
31. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. Cryptology ePrint Archive (2006)
32. Scafuro, A., Zhang, B.: One-time traceable ring signatures. In: Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part II 26. pp. 481–500. Springer (2021)
33. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
34. Stolbunov, A.: Cryptographic schemes based on isogenies (2012)

A Isogenies and Ideal class group action

Let p be a prime number with $p \geq 5$. E and E' denote elliptic curves defined over a finite field F_p. An isogeny ϕ: $E \to E'$ is a non-constant rational map defined over F_p that maps the identity element of E to the identity element of E'. $End_p(E)$ is the subring of the endomorphism ring $End(E)$ consisting the endomorphisms defined over F_p. In particular, for a supersingular curve over F_p, its full endomorphism ring $End(E)$ is an order in quaternion algebra, $End_p(E)$ is an order in the imaginary quadratic field. Define the set of isomorphism classes of elliptic curves $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$. In the following, let order $\mathcal{O} = End_p(E)$.

Definition 3. (Ideal Class Group Action) *Let $\mathcal{O}$ be an order in an imaginary quadratic field and $\pi \in \mathcal{O}$ such that $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ is non-empty. The ideal-class group $\mathcal{C}\ell(\mathcal{O})$ acts freely and transitively on the set $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$ via the map ($\mathfrak{a}$ is chosen as an integral representative):*

$$\mathcal{C}\ell(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}, \pi) \longrightarrow \mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$$

$$([\mathfrak{a}], E) \longmapsto E/\mathfrak{a}$$

We write $\mathfrak{a} \star E$ to denote $E/\mathfrak{a}$, then an ideal class action can be defined by $\mathfrak{a} \star E = E'$ such that there exists an isogeny $\phi : E \to E'$ with $ker(\phi) = \cap_{\alpha \in \mathfrak{a}}\{P \in E(\overline{F_p}) | \alpha(P) = O\}$. The main hardness assumption underlying ideal group actions based on isogenies is that it is infeasible to invert the group action.

Definition 4. (Group Action Inverse Problem(GAIP)) *Let E_0 be an element in set $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, and E is sampled uniformly from $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$. Given a pair (E_0, E), the GAIP problem is to find an ideal $\mathfrak{a} \in \mathcal{C}\ell(\mathcal{O})$, such that $E = \mathfrak{a} \star E_0$.*

Definition 5. (Squaring Decisional CSIDH Problem (sdCSIDH)) *Let E_0 be an element in set $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$. and $\mathfrak{a}$ is sampled from $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$. Given (E, E') sampled from $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi)$, the sdCSIDH problem is to distinguish the two distributions $(\mathfrak{a} \star E_0, \mathfrak{a}^2 \star E_0)$ and (E, E').*

B Lattices

For security parameter λ, let $F(X) = X^n + 1$ where $n = n(\lambda)$ is a power of 2, for integer $q = q(\lambda) \geq 2$. Let $R = \mathbb{Z}[X]/(F(X))$ and $R_q = R/qR$. Norms of R lie over $\mathbb{Z}^n$. Norms of R_q are defined by representing coefficients of elements over R_q in the range $(-q/2, q/2]$ when q is even and $[-(q-1)/2, (q-1)/2]$ when q is odd.

Definition 6. (Module Short Integer Solution Problem(MSIS) *Let k, ℓ and γ be integers. The $\mathsf{MSIS}_{n,q,k,\ell,\gamma}$ problem is to find a nonzero polynomial vector $\mathbf{v}$ of norm $0 < ||\mathbf{v}||_\infty \leq \gamma$ such that $[\mathbf{A} \mid \mathbf{I}] \cdot \mathbf{v} = 0$ where $\mathbf{A} \leftarrow R_q^{k \times \ell}$.*

Definition 7. (Module Learning with Errors Problem (MLWE)) *Let k, ℓ be integers, and D is a probability distribution over R_q. The $\mathsf{MLWE}_{n,q,k,\ell,D}$ problem is to distinguish the following two distributions: In the first distribution $(\mathbf{A}, \mathbf{b})$, one samples $\mathbf{A}$ uniformly from $R_q^{k\times\ell}$ and $\mathbf{b}$ from R_q^k. In the second distribution $(\mathbf{A}', \mathbf{b}')$, one first samples $\mathbf{s}$ uniformly from D^l, then samples $\mathbf{A}'$ uniformly from $R_q^{k\times\ell}$ and $\mathbf{e}$ from D^k, and setting $\mathbf{b}' = \mathbf{A}' \cdot \mathbf{s} + \mathbf{e}$.*

C Analysis of Lattice-based TRS

C.1 Correctness

The correctness of our TRS scheme Π_{LAT} is composed of *completeness* and *traceability*.

Completeness. The output range of the group action in lattice-based instantiation is a vector space R_q^m, $1/\pi$ is an element in R_q, aux can be seen as the result of scalar multiplication on $(T_\pi - a)$ by $1/\pi$. The signer generates signature σ and tag set TagSet through the $\mathsf{RSign\_LAT}$ algorithm, the verifier reconstructs TagSet. Due to the completeness of the underlying OR sigma protocol, the $\mathsf{RVer\_LAT}$ algorithm always outputs accept for an honest signer.

Traceability. We demonstrate the *traceability* of the scheme in all possible situations under the same tag L.

- **Situation 1** ($\pi = \pi' \wedge M = M'$). From the $\mathsf{RSign\_LAT}$ algorithm we have $T_0 = T_0'$, $T_\pi = T_\pi'$, $a = \mathcal{H}_5(L, M) = a' = \mathcal{H}_5(L, M')$, so that $T_\pi = T_\pi'$ and $\mathsf{aux} = \mathsf{aux}'$, and there is always $T_i = T_i'$ for $i \in N, i \neq \pi$. In this situation, the $\mathsf{TRace\_LAT}$ algorithm outputs linked.
- **Situation 2** ($\pi = \pi' \wedge M \neq M'$). When $x \neq x'$, $\mathcal{H}_5(x) = \mathcal{H}_5(x')$ means we find a collision of $\mathcal{H}_5$, hence we can conclude that $a \neq a'$ with overwhelming probability. For $i = \pi$, we have $T_\pi = sk_\pi \star T_0, T_\pi' = sk_\pi \star T_0$. For all $i \in N$, $i \neq \pi$, it holds that: $k \neq k'$, since the occurrence of four elements $a, \mathsf{aux}, a', \mathsf{aux}' \in R_q^n$ that satisfy $a + \mathsf{aux} \cdot i = a' + \mathsf{aux}' \cdot i$ is negligible, with a probability of at most q^{2-n}. Therefore only one pair of tags are equal in the two tag sets: $T_\pi = T_\pi'$ and the $\mathsf{RTrace\_LAT}$ algorithm outputs pk_π.
- **Situation 3** ($\pi \neq \pi'$). Since $\pi \neq \pi'$, we have $T_\pi \neq T_\pi'$. When $M = M'$, we have $a = a'$ and $\mathsf{aux} \neq \mathsf{aux}'$, thus $T_i \neq T_i'$ for all $i \in N, i \neq \pi$. When $M \neq M'$, we have $a \neq a'$, as proof in situation 2, it holds $T_i \neq T_i'$ for all $i \in N, i \neq \pi$. Consequently, it is difficult to find $i \in N$ such that $T_i = T_i'$ and the $\mathsf{RTrace\_LAT}$ algorithm outputs indep.

C.2 Security

The security proof for Π_{LAT} scheme follows a similar structure to the Π_{LAT} scheme, except for the generation of tag set.

Theorem 2. *Our TRS scheme* Π_{LAT} *provides tag-linkability, anonymity and exculpability, as long as the OR sigma protocol is soundness and zero-knowledge, the hash function* $\mathcal{H}_1, \mathcal{H}_4, \mathcal{H}_5$ *are collision-resistant, the* ResPGA *is a restricted pair of group actions.*

Proof. **Tag-Linkability.** Conversely, if there exists an adversary $\mathcal{A}$ that makes at most B random oracle queries, the probability of $\mathcal{A}$ winning the game is not negligible. Then we can use $\mathcal{A}$ to construct an algorithm $\mathcal{B}$, which breaks the Item 5 of ResPGA and collision resistance of $\mathcal{H}_2$. The simulation of $\mathcal{B}$ under random oracle is as follows:

- $\mathsf{Sim}^1_{\mathcal{B}}$: The winning output of $\mathcal{A}$ in the tag-linkability game serves as the input of $\mathcal{B}$. Assuming that $\{(L,(M_1,\sigma_1)),\ldots,(L,(M_{N+1},\sigma_{N+1}))\}$ are the outputs of $\mathcal{A}$, $\sigma_i = (\mathsf{aux}_i, \mathsf{com}_i, \mathsf{chall}_i, \mathsf{rsp}_i)$, chall_i is the output of $\mathcal{H}_3$ on input $(M_i, \mathsf{rpk_i}, \mathsf{TagSet}_i, \mathsf{com}_i)$, $\mathcal{A}$ records these transcripts into list $\mathsf{List} = \{i, \mathsf{aux}_i, (\mathsf{com}_i, \mathsf{chall}_i, \mathsf{rsp}_i), M_i\}_{i\in[N+1]}$.
- $\mathsf{Sim}^2_{\mathcal{B}}$: $\mathcal{B}$ re-invokes $\mathcal{A}$ until $\mathcal{A}$ wins the tag-linkability game. Different from $\mathsf{Sim}^1_{\mathcal{B}}$, $\mathcal{B}$ controls the randomness used in the underlying main OR sigma protocol to generate signatures in each query. For responding to the j-th signing query $\mathcal{B}$ uses the same randomness as in $\mathsf{Sim}^1_{\mathcal{B}}$ before q_i-th ($q_i \in [B]$) random oracle query, then $\mathcal{B}$ uses fresh randomness to interacti with $\mathcal{A}$. Let $\sigma' = \left(\mathsf{aux}'_j, (\mathsf{com}'_j, \mathsf{chall}'_j, \mathsf{rsp}'_j), M'_j, L\right)_{j\in[N+1]}$ be the output of $\mathcal{A}$. If σ'_j does not appear in the q_j-th random oracle query, then $\mathcal{B}$ restarts the simulation from $\mathsf{Sim}^2_{\mathcal{B}}$, $\mathcal{B}$ updates list $\mathsf{List} = \mathsf{List} \cup \{j, \mathsf{aux}'_j, (\mathsf{com}'_j, \mathsf{chall}'_j, \mathsf{rsp}'_j), M'_j\}$, chall'_j is the result of the q_j-th random oracle query. Due to the randomness used by both simulations is fixed before the q_j-th random oracle query, we have $(M_j, \mathsf{aux}_j, \mathsf{com}_j) = (M'_j, \mathsf{aux}'_j, \mathsf{com}'_j)$ for all the entries in the list.
- $\mathsf{Sim}^3_{\mathcal{B}}$: $\mathcal{B}$ extracts $(\mathsf{aux}_j, (\mathsf{com}_j, \mathsf{chall}_j, \mathsf{rsp}_j), M_j)$ from List generated in $\mathsf{Sim}^1_{\mathcal{B}}$, and extracts $(\mathsf{aux}_j, (\mathsf{com}_j, \mathsf{chall}'_j, \mathsf{rsp}'_j), M_j)$ generated in $\mathsf{Sim}^1_{\mathcal{B}}$ that satisfy $(M_j, \mathsf{aux}_j, \mathsf{com}_j) = (M'_j, \mathsf{aux}'_j, \mathsf{com}'_j)$, If $\mathsf{chall}_j = \mathsf{chall}'_j$, $\mathcal{B}$ aborts the simulation, otherwise, $\mathcal{B}$ invokes the underlying main OR sigma protocol extraction algorithm, on input the statement $(\mathsf{rpk}_j, \mathsf{TageSet}_j)$ and two accepted transcripts $(\mathsf{com}_j, \mathsf{chall}_j, \mathsf{rsp}_j), (\mathsf{com}_j, \mathsf{chall}'_j, \mathsf{rsp}'_j)$, where TagSet_j is generated by the public input aux_j, M_j and L, it outputs a witness w_j.
- $\mathsf{Sim}^4_{\mathcal{B}}$: For $j, j' \in [N+1]$, if there exists $w_j = (sk_j, \pi), w'_j = (sk'_j, \pi)$, then $\mathcal{B}$ outputs (sk_j, sk'_j), if w_j forms a collision of $\mathcal{H}_2$, $\mathcal{B}$ outputs w_j, otherwise, $\mathcal{B}$ aborts the simulation.

The witness $(w_j)_{j\in[N+1]}$ has the form: $w_j = (sk_j, \pi_j)$ such that $S_{\pi_j} = sk_j \star S_0, T_{\pi_j} = sk_j \star T_0$ or a collision of $\mathcal{H}_2$. If no collision occurs, due to the pigeonhole principle, there must have two indexes $j', j \in [N+1]$ such that $w_j = (sk_j, \pi), w'_j = (sk'_j, \pi)$. Further, the winning conditions of tag-linkability game indicate that $sk_j \star S_0 = sk'_j \star S_0$ but $sk_j \star T_0 \neq sk'_j \star T_0$, this violates the Item 5 of the restricted pair of group actions. Otherwise w_j is a collision of $\mathcal{H}_2$. □

Anonymity. The anonymity of scheme can be deduced from a sequence of games. The first game is the same as the true anonymity game, where $c = 0$. The last game is exactly like the original anonymity game, where $c = 1$. For for any PPT adversary $\mathcal{A}$, the probability that he distinguishes between any two games is negligible. Let $Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_i}$ denotes the advantage of adversary $\mathcal{A}$ in Game_i.

- Game_1 and Game_2: The process for both games is the same as the anonymity proof in Π_{LAT}. Then we have that output distribution of Game_1 and Game_2 is indistinguishable.
- Game_3: The challenger $\mathcal{C}$ simulates N public-secret key pairs $\{(sk_i, pk_i)\}_{i\in[N]}$ instead of running the Setup algorithm, then $\mathcal{C}$ guesses that the pair of public keys $\{pk_j^*, pk_k^*\}$ sent by the adversary exactly j-th and k-th public keys, then generates aux and two tag sets from $\{sk_j^*, pk_j^*\}$ and $\{sk_k^*, pk_k^*\}$:

$$\mathsf{aux}_0 = \frac{sk_j^* \star T_0 - a}{j}, \mathsf{aux}_1 = \frac{sk_k^* \star T_0 - a}{k}$$
$$\mathsf{TagSet}_0 = \Big(T_0, T_j = sk_j^* \star T_0, (T_i = (a + \mathsf{aux}_0 \cdot i) \star T_0)_{i\in[N]\setminus j}\Big)$$
$$\mathsf{TagSet}_1 = \Big(T_0, T_k = sk_k^* \star T_0, (T_i = (a + \mathsf{aux}_1 \cdot i) \star T_0)_{i\in[N]\setminus k}\Big)$$

  where $a = \mathcal{H}_5(L, M), T_0 = \mathcal{H}_4(L)$. If the guess is incorrect, the challenger randomly samples a bit as the output of $\mathcal{A}$ and terminates the game. Otherwise, it responds to the signing queries using a pre-computed TagSet_0, TagSet_1, aux_0 and aux_1 at the beginning of Game_2. Since the probability of the challenger correctly guessing the two public keys is at most $1/N^2$, thus we have :

$$Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_3}(\lambda) \approx \frac{1}{N^2} Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_2}(\lambda)$$

- Game_4: The challenger $\mathcal{C}$ randomly samples $\{i_0, i_1\} \leftarrow [N]$, then he simulates $N-2$ public-secret key pairs $\{(sk_i, pk_i)\}_{i\in[N]\setminus\{i_0,i_1\}}$. $\mathcal{C}$ samples (E_0, E_1) uniformly from $\mathcal{T}$ and computes:

$$\mathsf{aux}_0 = \frac{E_0 - a}{i_0}, \mathsf{TagSet}_0 = \Big(T_0, (T_i = (a + \mathsf{aux}_0 \cdot i) \star T_0)_{i\in[N]}\Big)$$
$$\mathsf{aux}_1 = \frac{E_1 - a}{i_1}, \mathsf{TagSet}_1 = \Big(T_0, (T_i = (a + \mathsf{aux}_1 \cdot i) \star T_0)_{i\in[N]}\Big)$$

  where $T_0 = \mathcal{H}_4(L)$, the rest is the same as Game_3. From the *weak-pseudrandom* of the restricted pair of group actions, we have:

$$(sk \star T_0 : sk \leftarrow \boldsymbol{\mathcal{G}_1}) \approx (E : E \leftarrow \boldsymbol{\mathcal{T}})$$

  Thus Game_4 is computationally indistinguishable from Game_3: $Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_1}(\lambda)$ $\approx Adv^{\mathsf{anon}}_{\mathcal{A},\mathsf{Game}_3}(\lambda)$. Now, the secret key is no longer used to generate the signature, i.e., the output of signing query does not reveal any information about the bit c in Game_4.

– Game_5: Game_5 is the same as an actual anonymous game, where $c = 1$.

There is no such adversary $\mathcal{A}$ that can distinguish any two games with a non-negligible probability, that is, the probability of the adversary winning the real anonymity game is negligible. □

Exculpability. If there exists an adversary $\mathcal{A}$ wins $\mathsf{Game}_{\mathcal{A}}^{\mathsf{excu}}$ with non-negligible probability, then we can construct an algorithm $\mathcal{B}$ from $\mathcal{A}$ that breaks the property Item 6 of restricted pair of group actions.

First, we simulate a game Game_1 which is indistinguishable from the real game $\mathsf{Game}_{\mathcal{A}}^{\mathsf{excu}}$. In Game_1, the challenger generates N public-secret key pairs $\{(sk_i, pk_i)\}_{i\in[N]}$, then he computes N tag sets $\mathsf{TagSet}_i = (T_{0,i} = \mathcal{H}_4(L), (T_{j,i} = (a + \mathsf{aux}_i) \star T_0)_{j\in[N]})_{i\in[N]}$ before the game starts where $\mathsf{aux}_i = \frac{sk_i \star T_0 - a}{\pi}$. If the signing query performed by the adversary contains index i, then the challenger uses N public-secret key pairs, precomputed N tag sets TagSet_i and N elements aux_i to generate the response. From the zero-knowledge of OR sigma protocol, indistinguishable of tag sets and collision resistance of $\mathcal{H}_4, \mathcal{H}_5$, we have $Adv^{\mathsf{excu}}_{\mathcal{A},\mathsf{Game}_1} \approx Adv^{\mathsf{excu}}_{\mathcal{A},\mathsf{Game}^{\mathsf{excu}}}$. When $\mathcal{A}$ wins Game_1, the simulation of $\mathcal{B}$ on the input (S, T) as follows:

– $\mathsf{Sim}^1_{\mathcal{B}}$: $\mathcal{B}$ randomly samples index $j \leftarrow [N]$, sets $pk_j = S$, $\mathsf{aux}_j = \frac{T-a}{j}$, and computes $\mathsf{TagSet}_j = \big(T_0 = \mathcal{H}_4(L), (T_i = (a + \mathsf{aux}_j \cdot i) \star T_0)_{i\in[N]}\big)$, then generates the remaining $N - 1$ public-secret key pairs $\{(pk_i, sk_i)\}_{i\in[N]\setminus j}$.

– $\mathsf{Sim}^2_{\mathcal{B}}$: Since Game_1 does not reveal any information of secret key, $\mathcal{B}$ simulates the view of Game_1. After interacting with $\mathcal{B}$, $\mathcal{A}$ outputs a forgery $(M, \mathsf{rpk}^*, \sigma^* = (\mathsf{aux}^*, \mathsf{com}^*, \mathsf{chall}^*, \mathsf{rsp}^*))$. To make sure the signature σ^* wins the Game_1, $\mathcal{B}$ must have responded to the signing query (i, M, rpk) with signature $\sigma = (\mathsf{aux}', \mathsf{com}', \mathsf{chall}', \mathsf{rsp}')$. If $i \neq j$, $\mathcal{B}$ terminates the simulation, otherwise, we have $\mathsf{aux}' = \mathsf{aux}_j$ and $\mathsf{aux}^* = \mathsf{aux}_j$, then $\mathcal{B}$ can extract witness w from the signature σ^* by rerunning $\mathcal{A}$. It is the same as what we have shown in the proof for tag-linkability.

– $\mathsf{Sim}^3_{\mathcal{B}}$: If w does not constitute a collision of $\mathcal{H}_2$, then we have $w = (sk, \pi)$ such that $sk \star T_0 = T_\pi$, $\mathcal{B}$ outputs $w = (sk, \pi)$, which violates the Item 6 of the underlying restricted pair of group actions. □