Efficient Unique Ring Signatures From Lattices*

Tuong Ngoc Nguyen[1], Anh The Ta[3], Huy Quoc Le[1,2], Dung Hoang Duong[1],
Willy Susilo[1], Fuchun Guo[1], Kazuhide Fukushima[4], and Shinsaku Kiyomoto[4]

[1] School of Computing and Information Technology, University of Wollongong,
Northfields Avenue, Wollongong NSW 2522, Australia
{ntn807,qhl576}@uowmail.edu.au, {hduong,wsusilo,fuchun}@uow.edu.au
[2] CSIRO Data61, Sydney, NSW, Australia,
[3] AI Lab, FPT Software Ltd., Ho Chi Minh City, Vietnam
tatheanhdtvt@gmail.com
[4] Information Security Laboratory, KDDI Research, Inc.
2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502, Japan
{ka-fukushima,kiyomoto}@kddi-research.jp

Abstract. Unique ring signatures (URS) were introduced by Franklin and Zhang (FC 2012) as a unification of linkable and traceable ring signatures. In URS, each member within a ring can only produce, on behalf of the ring, at most one signature for a message.
Applications of URS potentially are e–voting systems and e–token systems. In blockchain technology, URS have been implemented for mixing contract. However, existing URS schemes are based on the Discrete Logarithm Problem, which is insecure in the post-quantum setting.

In this paper, we design a new lattice-based URS scheme where the signature size is logarithmic in number of ring members. The proposed URS exploits a Merkle tree-based accumulator as building block in the lattice setting. Our scheme is secure under the Short Integer Solution and Learning With Rounding assumptions in the random oracle model.

Key words: unique ring signatures, lattice-based cryptography, Merkle tree accumulator, zero knowledge argument of knowledge

1 Introduction

Ring signatures (RS) were firstly introduced by Rivest, Shamir, and Tauman [RST01]. An RS of a group of signers (called the *ring*) is designed in such a way that any member in the group can sign messages on behalf of the group, but no one can tell who the real signer is. The ring can be dynamically formed by a signer without the need of agreement from other members. Thus, RS schemes offer the *anonymity* property. However, the strong anonymity of RS may be uncontrollably overused in some scenarios, for instance, in the double-spending problem in cryptocurrencies [Nak08].

[*] A preliminary version of this paper appears in *the 27th European Symposium on Research in Computer Security* (ESORICS 2022). This is the full version.

Linkable ring signatures (LRS) [LWW04, LW05] is a variant of RS that not only preserves the anonymity of RS but also provides an additional important property in protecting RS from such problems as the double-spending. The property is that given two or more signatures, even on different messages, a third party can efficiently determine whether these signatures were produced by the same signer or not. We call this property the *linkability.* However, note that even if the same signer produced the signatures, that signer remains anonymous to third parties.

Another variant of RS is traceable ring signature (TRS) which was first introduced by Fujisaki and Suzuki in [FS07]. A TRS has to fulfil both of the following two cases: (i) if a signer signs on two different messages using the same ring and the same issue, then there is an efficient public procedure that will learn the signer's indentity; (ii) if a signer signs on the same message concerning the same ring and the same issue twice, then the two corresponding signatures can be efficiently determined to be produced by the same signer, but the signer is still anonymous. The paper [FZ12] discusses many applications of TRS and LRS, such as various e-voting systems, e-token systems and k-times anonymous authentication.

Franklin and Zhang [FZ12, FZ13] introduced unique ring signatures (URS) aiming to capture the features of both LRS and TRS. A URS signature has a part called *unique identifier.* A URS offers anonymity, unforgeability, and the so-called *uniqueness* property. The uniqueness property guarantees that k colluding signers in the same ring cannot produce *more than* k valid signatures for *the same* message. In general, URS is a special variant of LRS which enjoys a stronger security property: In LRS, two signatures are linked if they are signed with respect to the same ring, but in URS, two signatures are linked if they are signed with respect to the same ring AND the same message. Informally speaking, In a URS, the anonymity of an uncorrupted user should be preserved as long as he/she does not issue 2 signatures with respect to the same pair (*message*, *ring*). It is not the case with LRS, since any 2 signatures of the same user are linked.

URS schemes are potentially used in the e-voting systems, e-token systems and k-times anonymous authentication applications mentioned above. Moreover, in the blockchain technology, Mercer proposed a mixing contract based on the URS. The author implemented the Franklin– Zhang URS protocol using the secp256k1 elliptic curve (EC). The implementation makes URS compatible with Bitcoin and Ethereum's EC libraries [Mer16].

The rapid development of quantum algorithms, as well as the remarkable realization of quantum computers, not only offer more powerful computational devices but may also lead to severe threats to many modern cryptography schemes and protocols. Indeed, from a cryptographic point of view, Peter Shor [Sho94] showed that all cryptosystems, which are based on the hardness of classical (number-theoretic) assumptions, e.g., the Integer Factorization Problem and the Discrete Logarithm Problem (DLP), will be broken as soon as large-scale quantum computers realized. To address this issue, there have been many proposed alternative hard problems that are believed to be quantum-resistant. Among others, lattice-based cryptography, firstly introduced by Ajtai [Ajt96], is emerging as a promising direction as it has better asymptotic efficiency than others and supports many advanced functionalities

There exist several schemes in the literature over lattices regarding ring signatures [LLNW16,ESS$^+$19,BDH$^+$19,CGH$^+$21], linkable ring signatures [BLO18, LAZ19, BKP20], and traceable ring signatures [FLWL20]. So far, nevertheless, there have been only URS constructions [FZ12,FZ13] that all base their security on the Computational Diffie-Hellman (CDH) and/or Decisional Diffie-Hellman (DDH) assumptions. The CDH and DDH, in turn, rely on the difficulty of the DLP problem. Hence, these schemes would be no longer secure in the quantum era. Moreover, these URS schemes, unfortunately, have the signature size of $\mathcal{O}(\lambda N)$ which is linear in N, where λ is the security parameter, and N is the number of members in the corresponding ring. Ta *et al.* [TKN$^+$21] addressed this issue by proposing a URS scheme with logarithmic size. However, the signature scheme is only based on DDH and DLP problems. Therefore, it is essential to design new URS schemes that are not only based on alternative hard problems offering quantum safety but also more efficient in terms of signature size.

1.1 Contributions

We propose a unique ring signature scheme based on the hardness of the Short Integer Solution (SIS) and the Learning With Rounding (LWR) problems in lattices. The construction exploits the accumulator technique introduced by Libert et al. [LLNW16]. However, in order to obtain the uniqueness property, we add a *unique tag* to every node of the Merkle tree. Since the tag only needs to be computed once and then appended to every node, the extra computation cost is insignificant. Specifically, the signature size of the proposed URS scheme is $\mathcal{O}(\lambda \log N)$ where λ is the security parameter, and N is the number of members in the ring, versus $\mathcal{O}(\lambda N)$ in the schemes of Franklin and Zhang [FZ13]. To the best of our knowledge, the scheme is the *first* lattice-based unique ring signature. Thus, the scheme is the first that is secure against classical adversaries basing the security proofs on post-quantum assumptions. Moreover, in comparison with existing URS schemes, our signature size is much smaller since it is logarithmic in the number of ring members. In Table 1, we make a comparison of existing URS schemes with ours.

1.2 Overview of the Results

Our key idea is to transform lattice-based ring signatures with logarithmic size to get the desired unique ring signatures. To this end, we embed the so-called *unique tag* (a.k.a., *unique identifier*) into the ring signature schemes. A unique tag corresponding to a signer can be computed using a weak pseudorandom function. Namely, if we denote the hash function (modelled as a random oracle) being used here by H_{UT}, a weak pseudorandom function F, a message by μ, a ring of signers by R, and the secret key for a signer by sk, then the uniqueness tag for the signer is of the form $\mathbf{t} := \mathsf{F}_{sk}(H_{\mathsf{UT}}(\mu, R))$. Our starting point is a Merkle tree-based

Table 1: Comparison our URS with the [FZ12, FZ13] URSs

URS schemes	**Assumptions**	**Security model**	**Signature size**	**Based on post-quantum assumptions?**
Franklin & Zhang [FZ12]	CDH + DDH	ROM /SDM	linear	✗
Franklin & Zhang [FZ13]	DDH	ROM	linear	✗
Ta *et al.* [TKN$^+$21]	DDH + DLP	ROM	logarithmic	✗
Ours	SIS + LWR	ROM	logarithmic	✓

accumulator used in the LLNW ring signature in lattices [LLNW16]. Intuitively, the Merkle tree there looks like a binary tree but travelling via the bottom-up direction. The leaves' values are ones that we want to accumulate, while the root corresponds to the accumulator value. See Figure 1 for an illustration of how data values $\mathbf{p}_0, \cdots, \mathbf{p}_7$ are accumulated into the value $\mathbf{v}_\epsilon$. The associated hash function used to accumulate is denoted by $h_\mathbf{A}$, being indicated by a random matrix $\mathbf{A} := [\mathbf{A}_0|\mathbf{A}_1] \in \mathbb{Z}_q^{n\times m/2} \times \in \mathbb{Z}_q^{n\times m/2}$. Formally, $h_\mathbf{A}(\mathbf{v}) := \mathsf{bin}(\mathbf{A}_0\mathbf{v}_0 + \mathbf{A}_1\mathbf{v}_1 \pmod q) \in \{0,1\}^{m/2}$, for any vector $\mathbf{v} := (\mathbf{v}_0, \mathbf{v}_1) \in \{0,1\}^{m/2} \times \{0,1\}^{m/2}$. Here, bin denotes the binary decomposition operation. Such a hash function is proved to be collision-resistant under the hardness of the Short Integer Solution (SIS) problem.

To transform the ring signature in [LLNW16] to a URS, we modify the Merkle tree and the corresponding hash function. The modified Merke tree will also allow to accumulate the uniqueness tag $\mathbf{t}$. Specifically, for each hashing time in the modified Merkle tree, each of two inputs is also appended to the uniqueness tag $\mathbf{t}$. For instance, the inputs now are $(\mathbf{v}_0^\top|\mathbf{t}^\top)$, $(\mathbf{v}_1^\top|\mathbf{t}^\top)$ instead of $(\mathbf{v}_0, \mathbf{v}_1)$. (See Figure 2 for the modified Merkle tree illustration used in our URS.) Accordingly, the hash function $h_\mathbf{A}$ will be changed to be

$$h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0, \mathbf{v}_1) := \mathsf{bin}(\mathbf{A}_0\mathbf{v}_0 + \mathbf{A}_1\mathbf{v}_1 + \mathbf{B}_0\mathbf{t} + \mathbf{B}_1\mathbf{t} \pmod q)) \in \{0,1\}^{nk},$$

where $\mathbf{B} = [\mathbf{B}_0|\mathbf{B}_1] \leftarrow H_{\mathsf{UT}}(\mu, R)$. Since $\mathbf{B}$ and $\mathbf{t}$ are fixed as constants then by a simple reduction, we can prove that $h_{\mathbf{A},\mathbf{B},\mathbf{t}}$ is also collision-resistant assuming the hardness of the average case SIS assumption in lattices.

In this paper, we consider the following relation:

$$\begin{aligned}\mathcal{R}_{\mathsf{URS}} := \{&(\mathbf{A}, \mathbf{B}, \mathbf{v}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m} \times \{0,1\}^{nk};\\ &\mathbf{x} \in \{0,1\}^m, \mathbf{p} \in \{0,1\}^{nk}, \mathbf{t} \in \{0,1\}^{nk}, \mathsf{wit} \in \{0,1\}^\ell \times (\{0,1\}^{nk})^\ell :\\ &\mathsf{ACC.Verify}_\mathbf{A}(\mathbf{B}, \mathbf{t}, \mathbf{v}, \mathbf{p}, \mathsf{wit}) = 1 \ \wedge\ \mathbf{A}\mathbf{x} = \mathbf{G}\mathbf{p} \ \wedge\ \mathsf{F}_\mathbf{x}(\mathbf{B}) = \mathbf{G}\mathbf{t}\}.\end{aligned}$$

Here, the *gadget matrix* $\mathbf{G}$ is a special matrix with property that $\mathbf{G} \cdot \mathsf{bin}(\mathbf{a}) = \mathbf{a}$. We are therefore able to utilize the same "extend-then-permute" technique done

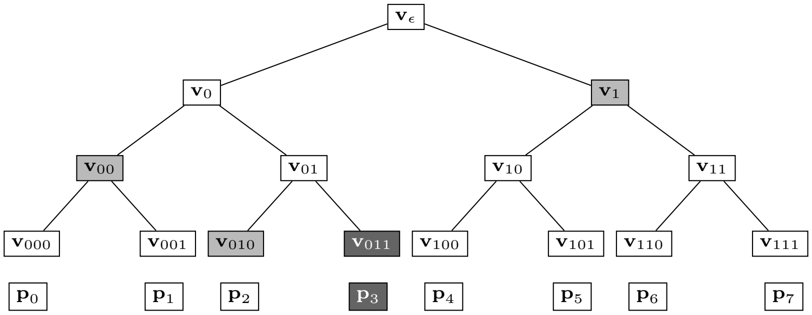

Fig. 1: An illustration for Merkle tree-based accumulator in [LLNW16], in which 2^3 data values $\mathbf{p}_0, \cdots, \mathbf{p}_7$ are accumulated into the value $\mathbf{v}_\epsilon$. It works as follows. First, it assigns data values $\mathbf{p}_i$'s to the leaves of the tree (at depth 3) and renames it as $\mathbf{v}_{i_1,i_2,i_3}$ where $(i_1, i_2, i_3) = \mathsf{bin}_3(i) \in \{0,1\}^3$, i.e., $\mathbf{v}_{i_1,i_2,i_3} \leftarrow \mathbf{p}_i$. At depth 3, it accumulates the pair ($\mathbf{v}_{000} := \mathbf{p}_0$, $\mathbf{v}_{001} := \mathbf{p}_1$) to get $\mathbf{v}_{00}$. Similarly for ($\mathbf{v}_{010} := \mathbf{p}_2$, $\mathbf{v}_{011} := \mathbf{p}_3$), ($\mathbf{v}_{010} := \mathbf{p}_4$, $\mathbf{v}_{011} := \mathbf{p}_5$), ($\mathbf{v}_{010} := \mathbf{p}_6$, $\mathbf{v}_{011} := \mathbf{p}_7$) to get $\mathbf{v}_{01}$, $\mathbf{v}_{10}$, $\mathbf{v}_{11}$, respectively. At depth 2, it continues to accumulate two pairs $(\mathbf{v}_{00}, \mathbf{v}_{01})$, $(\mathbf{v}_{10}, \mathbf{v}_{11})$ to get $\mathbf{v}_0$ and $\mathbf{v}_1$, respectively. Finally, at depth 3, $(\mathbf{v}_0, \mathbf{v}_1)$ is accumulated to $\mathbf{v}_\epsilon$ located at the root of the tree. The witness for the fact that $\mathbf{p}_3$ (i.e., $\mathbf{v}_{011}$) (dark grey-filled boxes) has been accumulated is $\mathsf{wit} = \{011, \{\mathbf{v}_{010}, \mathbf{v}_{00}, \mathbf{v}_1\}\}$ (light grey-filled boxes).

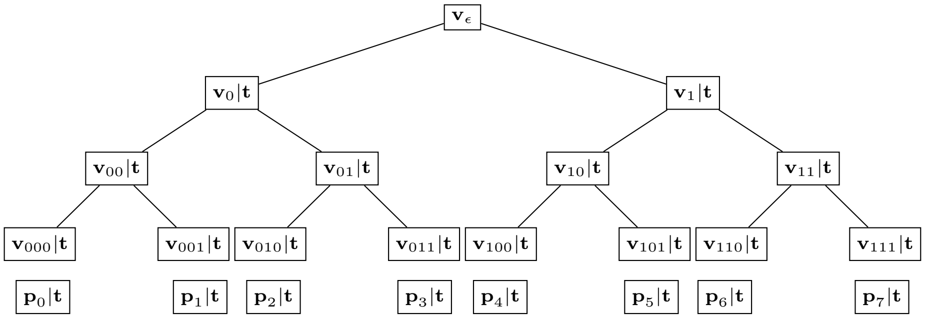
$\mathbf{v}_\epsilon$
$\mathbf{v}_0|\mathbf{t}$
$\mathbf{v}_1|\mathbf{t}$
$\mathbf{v}_{00}|\mathbf{t}$
$\mathbf{v}_{01}|\mathbf{t}$
$\mathbf{v}_{10}|\mathbf{t}$
$\mathbf{v}_{11}|\mathbf{t}$
$\mathbf{v}_{000}|\mathbf{t}$
$\mathbf{v}_{001}|\mathbf{t}$
$\mathbf{v}_{010}|\mathbf{t}$
$\mathbf{v}_{011}|\mathbf{t}$
$\mathbf{v}_{100}|\mathbf{t}$
$\mathbf{v}_{101}|\mathbf{t}$
$\mathbf{v}_{110}|\mathbf{t}$
$\mathbf{v}_{111}|\mathbf{t}$
$\mathbf{p}_0|\mathbf{t}$
$\mathbf{p}_1|\mathbf{t}$
$\mathbf{p}_2|\mathbf{t}$
$\mathbf{p}_3|\mathbf{t}$
$\mathbf{p}_4|\mathbf{t}$
$\mathbf{p}_5|\mathbf{t}$
$\mathbf{p}_6|\mathbf{t}$
$\mathbf{p}_7|\mathbf{t}$

Fig. 2: The modified Merkle tree-based accumulator for our unique ring signatures.

for $\mathcal{R}_{\mathsf{ring}}$ to handle the ZKAoK for $\mathcal{R}_{\mathsf{URS}}$. The details of the induced URS will be presented in Section 4.

1.3 Organization

In Section 2, we review some backgrounds neccessary for our work. Section 3 describes a parameterized accumulator in lattices, which will be applied to our URS construction. Section 4 gives in details the lattice-based construction of URS from accumulators and related proofs.

2 Preliminaries

Notation. Throughout this work, all vectors are in column form unless otherwise stated. A vector is written in bold-face small letter, e.g., $\mathbf{v}$, while a matrix in bold-face capital letter, e.g., $\mathbf{A}$. The transpose operation of a vector or a matrix denoted by the superscript $\top$; e.g., transpose of vector $\mathbf{v}$ is $\mathbf{v}^\top$. For $k \in \mathbb{N}$, the notation $[k]$ means $\{1, \ldots, k\}$. We denote by $|S|$ the cardinality of a discrete set S.

2.1 Framework of Unique Ring Signatures

We first recall the framework of URS as introduced by Franklin and Zhang in [FZ12, FZ13].

Syntax. A URS scheme consists of four algorithms $\mathsf{URS} = (\mathsf{URS.Setup}, \mathsf{URS.KeyGen}, \mathsf{URS.Sign}, \mathsf{URS.Verify})$ described as follows.

$\mathsf{URS.Setup}(1^\lambda)$. This probabilistic polynomial time (PPT) algorithm takes as input a security parameter λ to output public parameters pp.

$\mathsf{URS.KeyGen}(pp)$. This PPT algorithm takes as input a public parameters pp to generate a secret signing key sk and a public verification key pk. This algorithm will be used to generate a key pair for each user.

$\mathsf{URS.Sign}(pp, \mu, R, sk)$. This PPT algorithm outputs a signature σ on the message μ, the ring $R = (pk_1, pk_2, \ldots, pk_N)$ using the secret key sk of a member of R. Note that, for URS the signature σ can be parsed as $\sigma = (\tau, \pi)$ where τ is called the *unique identifier* or the *unique tag*.

$\mathsf{URS.Verify}(pp, \mu, R, \sigma)$. This deterministic polynomial time (DPT) algorithm takes as input public parameters pp, a message μ, a ring of signers R and a ring signature σ, returns 1 if the signature σ is valid, and 0 otherwise.

Correctness. The correctness of URS is defined in Definition 1.

Definition 1 (Correctness). *For any $pp \leftarrow \mathsf{URS.Setup}(1^\lambda)$, any integer N, $i = 1, 2, \ldots, N : (pk_i, sk_i) \leftarrow \mathsf{URS.KeyGen}(pp)$, and $R = \{pk_1, pk_2, \ldots, pk_N\}$, for any message μ and any member (pk_j, sk_j) of R, the correctness of a URS holds that*

$$\mathsf{URS.Verify}(pp, \mu, R, \mathsf{URS.Sign}(pp, \mu, R, sk_j)) = 1.$$

In order to formally define the security notions, we need some further definitions.

Queried Oracles. Given $\{(pk_i, sk_i)\}_{i=1}^N$ and a ring $S = \{pk_i\}_{i=1}^N$ for reference, the adversaries can have access to one or more of the following oracles depending the security they involve:

- **The user secret key oracle** $\mathcal{O}_{sk}(i)$: Output the secret key sk_i of some member i in R.

- **The ring signature oracle $\mathcal{O}_{Sign}(i, R, \mu)$:** Output the ring signature on message μ respective to a subring $R \subseteq S$, in which $pk_i \in R$ is the real signer.

In addition to the above oracles, we need some more notations below.

- $\mathsf{SIGNER}_{R,\mu}$ denotes a set of users (i.e., secret keys) that have been queried to $\mathcal{O}_{Sign}(\cdot, R, \mu)$ by the adversary.
- $\overrightarrow{\mathsf{SIGNER}}_{\mathcal{R},\mathcal{M}} := \{\mathsf{SIGNER}_{R,\mu} : R \in \mathcal{R}, \mu \in \mathcal{M}\}$ where $\mathcal{R}$ is a set of rings and $\mathcal{M}$ a set of messages.
- $\mathsf{Corrupt}$ denotes the set of all users whose secret keys are given to the adversary.
- Also, $\mathsf{Corrupt}_R$ denotes the set of all users *in the ring* R, whose secret keys are given to the adversary.

We now give formal definitions for the URS security notions.

$\underline{\mathsf{URS}^{\mathsf{Anom}}_{\mathcal{A},N}:}$

Setup. The challenger $\mathcal{C}$ runs $\mathsf{URS.Setup}(1^\lambda)$ on input a security parameter λ, to get public parameters pp. Now, the challenger runs $\mathsf{URS.KeyGen}(pp)$ to get a key pair $(pk_i, sk_i) \leftarrow \mathsf{URS.KeyGen}(pp)$ for each user $i = 1, 2, \ldots, N$. Then $\mathcal{C}$ sets $S = \{pk_i\}_{i=1}^N$ and hands (pp, S) to $\mathcal{A}$. The challenger also sets $\mathsf{Corrupt} \leftarrow \emptyset$ and $\overrightarrow{\mathsf{SIGNER}}_{\mathcal{R},\mathcal{M}} \leftarrow \emptyset$.

Query 1. $\mathcal{A}$ may make polynomially bounded number of queries to the oracles $\mathcal{O}_{sk}$ and $\mathcal{O}_{Sign}$, and $\mathcal{C}$ responses it in the way mentioned above. Meanwhile, $\mathcal{C}$ updates the sets $\mathsf{Corrupt}$ and $\overrightarrow{\mathsf{SIGNER}}_{\mathcal{R},\mathcal{M}}$.

Challenge. $\mathcal{A}$ chooses two indices i_0, i_1 together with a message μ^* and a ring $R^* \subseteq S$ such that $i_0, i_1 \notin \mathsf{Corrupt}$ and $i_0, i_1 \notin \mathsf{SIGNER}_{R^*,\mu^*}$, and sends the tuple (i_0, i_1, μ^*, R^*) to $\mathcal{C}$. Then $\mathcal{C}$ choosed uniformly at random a bit $b \leftarrow \{0, 1\}$ and runs $\sigma \leftarrow \mathsf{URS.Sign}(\mu^*, R^*, sk_{i_b})$. The challenger returns σ to $\mathcal{A}$.

Query 2. Same as **Query 1**, with restriction that $\mathcal{A}$ is not allowed to query to $\mathcal{O}_{sk}(i_0)$, $\mathcal{O}_{sk}(i_1)$, and signing queries $\mathcal{O}_{Sign}(i_0, \mu^*, R^*)$, $\mathcal{O}_{Sign}(i_1, \mu^*, R^*)$.

Guess. $\mathcal{A}$ outputs a guess b' for b. $\mathcal{A}$ wins if and only if $b' = b$.

Fig. 3: Anonymity experiment for URS

Definition 2 (Anonymity). *A URS is called anonymous if for any polynomial-time adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{URS,Anom}}_{\mathcal{A},N}(\lambda)$ of $\mathcal{A}$ in the Anonymity experiment $\mathsf{URS}^{\mathsf{Anom}}_{\mathcal{A},N}$ presented in Figure 3 is negligible. That is,* $\mathsf{Adv}^{\mathsf{URS,Anom}}_{\mathcal{A},N}(\lambda) := 2\left|\Pr[b' = b] - 1/2\right| = \mathsf{negl}(\lambda)$.

Definition 3 (Unforgeability). *A URS is called unforgeable under adaptive chosen-message attacks if for any PPT adversary $\mathcal{A}$, the advantage* $\mathsf{Adv}^{\mathsf{URS,Unforge}}_{\mathcal{A},N}(\lambda)$

$\mathsf{URS}^{\mathsf{Unforge}}_{\mathcal{A},N}$:

Setup. Same as **Setup** of $\mathsf{URS}^{\mathsf{Anom}}_{\mathcal{A},N}$.
Query. Same as **Query 1** of $\mathsf{URS}^{\mathsf{Anom}}_{\mathcal{A},N}$.
Forge. The adversary $\mathcal{A}$ outputs a ring signature σ^* on a message μ^* and a ring $R^* \subseteq S$, with the condition that R^* does not contain corrupted users (i.e., $R^* \subseteq S \setminus \mathsf{Corrupt}$), and $\mathcal{A}$ has never made queries $\mathcal{O}_{Sign}(\cdot, R^*, \mu^*)$ before. $\mathcal{A}$ wins the game if and only if $\mathsf{URS.Verify}(\mu^*, R^*, \sigma^*) = 1$.

Fig. 4: Unforgeability experiment for URS

of $\mathcal{A}$ in the Unforgeability experiment $\mathsf{URS}^{\mathsf{Unforge}}_{\mathcal{A},N}$ presented in Figure 4 is negligible. That is,

$$\mathsf{Adv}^{\mathsf{URS,Unforge}}_{\mathcal{A},N}(\lambda) := \Pr[\mathcal{A}\ wins] = \mathsf{negl}(\lambda).$$

$\mathsf{URS}^{\mathsf{Unique}}_{\mathcal{A},N}$:

Setup. Same as **Setup** of $\mathsf{URS}^{\mathsf{Anom}}_{\mathcal{A},N}$.
Query. Same as **Query 1** of $\mathsf{URS}^{\mathsf{Anom}}_{\mathcal{A},N}$.
Forge. The adversary $\mathcal{A}$ outputs $t := |\mathsf{Corrupt}_{R^*} \cup \mathsf{SIGNER}_{R^*,\mu^*}| + 1$ different valid signatures $\sigma_1, \ldots, \sigma_t$ on the same message μ^* in regards the same ring R^*. The challenger parses the signatures as $\sigma_j = (\tau_j, \pi_j)$, and checks whether the unique tags $\tau_k,\ k = 1, 2, \ldots, t$, are pairwise distinct. If this is the case, then the challenger returns 1 and $\mathcal{A}$ wins; otherwise, returns 0 and $\mathcal{A}$ loses.

Fig. 5: Uniqueness experiment for URS

Definition 4 (Uniqueness). *A URS is called unique if for any PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{URS,Unique}}_{\mathcal{A},N}(\lambda)$ of $\mathcal{A}$ in the Uniqueness experiment $\mathsf{URS}^{\mathsf{Unique}}_{\mathcal{A},N}$ presented in Figure 5 is negligible. That is,*

$$\mathsf{Adv}^{\mathsf{URS,Unique}}_{\mathcal{A},N}(\lambda) := \Pr[\mathcal{A}\ wins] = \mathsf{negl}(\lambda).$$

Additionally, a URS is also required to satisfy the *non-colliding property.* Note that the non-colliding property is not a security requirement.

Definition 5 (Non-colliding property). *For all $i \neq j$, a URS scheme is non-colliding when the probability*

$$\Pr[\sigma_i = (\tau_i, \pi_i) \leftarrow \mathsf{URS.Sign}(\mu, R, sk_i), \sigma_j = (\tau_j, \pi_j) \leftarrow \mathsf{URS.Sign}(\mu, R, sk_j) : \tau_i = \tau_j]$$

is negligible to the security parameter λ.

Definition 6 (Security of URS). *A URS is called secure if it satisfies the correctness and the non-colliding property, and it is unforgeable, anonymous as well as unique.*

2.2 Lattices and Hardness Assumptions

An integer lattice $\mathcal{L}$ is a discrete subgroup which can be represented as $\mathcal{L} = \mathcal{L}(\mathbf{B}) := \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^m\}$, where $\mathbf{B} \in \mathbb{Z}^{n \times m}$ is a basis of $\mathcal{L}$. The lattice $\mathcal{L}$ is called *full-rank* if $n = m$.

Definition 7 (SIS, [Ajt96,GPV08]). *Short Integer Solution problem* $\mathsf{SIS}^{\infty}_{m,n,q,\theta}$ *is, given matrix* $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}$*, to find* $\mathbf{x} \in \mathbb{Z}^m$ *such that* $\mathbf{Ax} = 0 \pmod q$ *and* $0 < \|\mathbf{x}\|_\infty \leq \theta$.

Definition 8 (Decision-LWR, [BPR12]). *For a vector* $\mathbf{s} \in \mathbb{Z}_q^n$*, define the LWR distribution* $\mathcal{L}_{\boldsymbol{s}}$ *to be the distribuiton over* $\mathbb{Z}_q^n \times \mathbb{Z}_p$ *obtained by choosing a vector* $\mathbf{a} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$ *and outputting* $(\mathbf{a}, b = \lfloor \langle \mathbf{a} \cdot \mathbf{s} \rangle \rceil_p)$*. The decision-*$\mathsf{LWR}_{n,m,q,p}$ *is to distinguish between* m *independent samples* $(\mathbf{a}_i, b_i) \leftarrow \mathcal{L}_{\mathbf{s}}$*, and* m *samples drawn uniformly and independently from* $\mathbb{Z}_q^n \times \mathbb{Z}_p$*. We denote the advanatge of an LWR solver* $\mathcal{S}$ *by* $\mathsf{Adv}^{\mathsf{LWR}}(\mathcal{S})$.

Lemma 1 (Leftover Hash Lemma). *Given* m, n *are positive integers,* $q \geq 2$ *is a prime such that* $m \geq 2n \log q$*, and that* $\mathbf{x} \overset{\$}{\leftarrow} \{0,1\}^m, \mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}, \mathbf{y} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$*, the distribution* $(\mathbf{A}, \mathbf{A} \cdot \mathbf{x})$ *is statistically close to the distribution* $(\mathbf{A}, \mathbf{y})$.

Proof. An unbounded adversary $\mathcal{A}$ can guess $\mathbf{x}$ correctly with the probability $\upsilon = 1/2^m \leq 1/(2^{2n \log q})$. Because the range size is $|\Omega| = q^n = 2^{n \log q}$, the distinguishing advantage of $\mathcal{A}$ is bounded by $\upsilon \cdot |\Omega| = 1/q^n = \mathsf{negl}(n)$.

2.3 Accumulator Schemes

Syntax. An accumulator consists of the following algorithms:

$pp \leftarrow$ **ACC.Setup**(1^λ)**:** It takes as input a security parameter λ to output public parameters pp.

$\mathbf{v} \leftarrow$ **ACC.Acc**(pp, R)**:** It takes as input public parameters pp, a list of N data values $R = (\mathbf{p}_1, \cdots, \mathbf{p}_{N-1})$ to output an accumulator value $\mathbf{v}$ for R.

wit $\leftarrow$ **ACC.Witness**$(pp, R, \mathbf{p})$**:** It takes as input public parameters pp, a list of N data values $R = (\mathbf{p}_1, \cdots, \mathbf{p}_{N-1})$ and a data value $\mathbf{p}$. It outputs $\perp$ if $\mathbf{p} \notin R$. Otherwise, it outputs a witness wit proving that $\mathbf{p}$ has been accumulated in $\mathsf{ACC.Acc}$.

$0/1 \leftarrow$ **ACC.Verify**$(pp, \mathbf{v}, \mathbf{p}, \mathbf{wit})$**:** It takes as input public parameters pp, a pair $(\mathbf{p}, \mathsf{wit})$. It outputs 1 if $(\mathbf{p}, \mathsf{wit})$ is valid for the accumulator value $\mathbf{v}$ and outputs 0 otherwise.

Correctness. It is required for ACC that for all $pp \leftarrow \mathsf{ACC.Setup}(1^\lambda)$, $\mathbf{v} \leftarrow \mathsf{ACC.Acc}(pp, R)$, $\mathsf{wit} \leftarrow \mathsf{ACC.Witness}(pp, R, \mathbf{p})$, it holds that

$$\textbf{ACC.Verify}(pp, \mathbf{v}, \mathbf{p}, \mathsf{wit}) = 1, \text{ for all } \mathbf{p} \in R.$$

Security. An accumulator scheme is called secure if for all PPT adversaries $\mathcal{A}$:

$$\begin{aligned}&\Pr[pp \leftarrow \mathsf{ACC.Setup}(1^\lambda); (R, \mathbf{p}^*, \mathsf{wit}^*) \leftarrow \mathcal{A}(pp) :\\ &\quad \mathbf{p}^* \notin R \wedge \mathsf{ACC.Verify}(pp, \mathsf{ACC.Acc}(pp, R), \mathbf{p}^*, \mathsf{wit}^*) = 1] = \mathsf{negl}(\lambda).\end{aligned}$$

2.4 String Commitment Schemes

In this work, we also exploit the so-called *string commitment function*. We need it to be *statistically hiding* and *computationally binding*. The first property ensures that any computationally unbounded adversarial receiver cannot distinguish two commitment strings generated from two distinct strings. The second property says that no polynomial-time algorithm can change the committed string after sending the commitment. See [HM96, KTX08] for more details.

In lattices, such a string commitment scheme comes from Kawachi et al. [KTX08]. It is statistically hiding and computationally binding if the $\mathsf{SIS}^\infty_{m,n,q,\theta}$ problem is hard. We will denote it by $\mathsf{COM} : \{0,1\}^* \times \{0,1\}^m \to \mathbb{Z}_q^n$ and use it for the ZKAoK, which is generally described later in Section 2.5.

2.5 Zero Knowledge Arguments of Knowledge (ZKAoK)

Let $\mathcal{R} := \{(\mathsf{stm}, \mathsf{wit}) \in \{0,1\}^* \times \{0,1\}^*\}$ be a polynomial time decidable binary relation for a language $\mathcal{L}$ in the NP class. We call wit a witness for a statement $\mathsf{stm} \in \mathcal{L}$ if $(\mathsf{stm}, \mathsf{wit}) \in \mathcal{R}$.

A *statistical* zero knowledge arguments (ZKA) system for the relation $\mathcal{R}$ with soundness error ϵ is an interactive system ($\mathcal{P}$, $\mathcal{V}$) between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ endowed with the following properties:

1. **Completeness:** If $(\mathsf{stm}, \mathsf{wit}) \in \mathcal{R}$ then $\Pr[(\mathcal{P}(\mathsf{stm}, \mathsf{wit}), \mathcal{V}(\mathsf{stm})) = 1] = 1$.
2. **ϵ-Soundness:** If $(\mathsf{stm}, \mathsf{wit}) \notin \mathcal{R}$ then for all PPT $\mathcal{P}^*$, $\mathsf{Adv}^{\mathsf{sound}}_{(\mathcal{P},\mathcal{V})}(\mathcal{P}^*) := \Pr[(\mathcal{P}^*(\mathsf{stm}, \mathsf{wit}), \mathcal{V}(\mathsf{stm})) = 1] \leq \epsilon$. Here, note that $\mathcal{P}^*$ is a *computationally bounded* cheating prover.
3. **Statistical zero-knowledge:** For any $\mathcal{V}^*(\mathsf{stm})$, there exists a PPT simulator $\mathcal{S}(\mathsf{stm}))$ who is able to simulate a transcript statistically close to the transcript produced by the real interaction between $\mathcal{P}$ and $\mathcal{V}^*$. We define the advantage of $\mathcal{V}^*$ who can break the statistical zero-knowledge by $\mathsf{Adv}^{\mathsf{zk}}_{(\mathcal{P},\mathcal{V})}(\mathcal{V}^*)$.

The notion of *Argument of Knowledge* is related to the so-called withness-extended emulation [Lin03]. Informally stating, the withness-extended emulation

Roughly speaking, transcript is what the prover and the verifier have exchanged in a complete interaction.

requires that given an adversary that produces an acceptable argument with some probability, there exists an emulator that produces a similar argument with the same probability together with a witness wit. Note that the emulator can rewind the prover and verifier's interaction to any previous move. See [BCC$^+$16, Def. 7] for a formal definition.

2.6 Weak Pseudorandom Function (wPRF)

In [YAL$^+$17], a weak pseudorandom function under the LWR hardness assumtion was proposed. More precisely, let n, m, p, q are positive integers, $p \geq 2, \gamma = q/p$ is an odd integer and $m \geq n(\log q + 1)(\log p - 1)$, a wPRF F is described as below:

- $\mathsf{KeyGen}(1^\lambda)$. The algorithm takes as input a security parameter λ, and outputs $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_q^m$.
- $\mathsf{Eval}(\mathbf{x}, \mathbf{A})$. The algorithm takes as input $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and outputs $\mathsf{F}_{\mathbf{x}}(\mathbf{A}) = \lfloor \mathbf{A}\mathbf{x} \rceil_p$.

Lemma 2 ([YAL$^+$17]). *If the LWR$_{n,p,q}$ assumption holds, and $m \geq n(\log q + 1)(\log p - 1)$, then F is a secure wPRF.*

We adapt the domain of $\mathbf{x}$ to $\{0, 1\}^m$ in this paper. A wRPF F has the following properties:

- Weak Pseudorandomness. Let $\mathbf{x}_1 \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{y}_1 = \mathsf{F}_{\mathbf{x}}(\mathbf{A})$ and $\mathbf{x}_2 \xleftarrow{\$} \{0, 1\}^m, \mathbf{y}_2 \xleftarrow{\$} \mathbb{Z}_p^n$, any PPT adversary $\mathcal{A}$ successfully distinguishes $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$ with negligible probability.
- Strong Uniqueness. Let $\mathbf{x}_1, \mathbf{x}_2 \leftarrow \mathsf{KeyGen}(1^\lambda)$ be two secret keys, and two random matrices $\mathbf{A}_1, \mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, if we require $m \geq 2n(\log q + 1)(\log p - 1)$, we have:

$$\Pr[\exists \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1 \neq \mathbf{x}_2 \wedge \mathsf{F}_{\mathbf{x}_1}(\mathbf{A}_1) = \mathsf{F}_{\mathbf{x}_2}(\mathbf{A}_2)] \leq \mathsf{negl}(\lambda).$$

3 The Underlying Accumulator for Our URS

Aiming to apply the lattice-based accumulator to constructing a unique ring signature, we will build a modified Merkle tree and a corresponding family of hash functions. Now the Merkle tree also allows us to accumulate the unique tag $\mathbf{t} := \mathsf{bin}(H_{\mathsf{UT}}(\mu, R) \cdot \mathbf{x})$, where μ is a message, R is a ring of signers and $\mathbf{x}$ is the secret key for the real signer.

(See Figure 2 for the modified Merkle tree used in our URS). Accordingly, we take into account the family of hash functions formally defined in Definition 9 below.

Definition 9. *Let $k := \lceil \log q \rceil$, and $m := 2nk$. Fix a message M and a ring R, we define a family of hash functions as follows:*

$\mathcal{H}_{\mathbf{B},\mathbf{t}} = \{h_{\mathbf{A},\mathbf{B},\mathbf{t}} : \mathbf{A} := [\mathbf{A}_0|\mathbf{A}_1] \xleftarrow{\$} \mathbb{Z}_q^{n\times nk} \times \mathbb{Z}_q^{n\times nk}, \mathbf{B} := [\mathbf{B}_0|\mathbf{B}_1] \in \mathbb{Z}_q^{n\times nk} \times \mathbb{Z}_q^{n\times nk}, \mathbf{t} \in \{0,1\}^{nk}\}$ mapping from $\{0,1\}^{nk} \times \{0,1\}^{nk}$ to $\{0,1\}^{nk}$ such that

$$h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0, \mathbf{v}_1) := \mathsf{bin}(\mathbf{A}_0\mathbf{v}_0 + \mathbf{A}_1\mathbf{v}_1 + \mathbf{B}_0\mathbf{t} + \mathbf{B}_1\mathbf{t} \pmod q) \in \{0,1\}^{nk}.$$

Note that $h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0, \mathbf{v}_1) = \mathbf{v}$ is equivalent to $\mathbf{A}_0\mathbf{v}_0 + \mathbf{A}_1\mathbf{v}_1 + \mathbf{B}_0\mathbf{t} + \mathbf{B}_1\mathbf{t} = \mathbf{G}\mathbf{v} \pmod q$.

Lemma 3. *The family $\mathcal{H}_{\mathbf{B},\mathbf{t}}$ defined in Definition 9 is collision-resistant assuming the hardness of the $\mathsf{SIS}^{\infty}_{m,n,q,\theta}$.*

Proof. Given a matrix $\mathbf{B} = [\mathbf{B}_0|\mathbf{B}_1] \in \mathbb{Z}_q^{n\times nk} \times \mathbb{Z}_q^{n\times nk}$ and vector $\mathbf{t} \in \{0,1\}^{nk}$. Also given an instance $\mathsf{SIS}^{\infty}_{n,m,q,1}$ defined by $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1] \xleftarrow{\$} \mathbb{Z}_q^{n\times nk} \times \mathbb{Z}_q^{n\times nk}$, where $k = \lceil \log q \rceil$ and $m = 2nk$. Let $h_{\mathbf{A},\mathbf{B},\mathbf{t}}$ be a hash function as defined in Definition 9.

Suppose that there are two disticnt $(\mathbf{v}_0, \mathbf{v}_1) \neq (\mathbf{v}_0', \mathbf{v}_1') \in \{0,1\}^{nk} \times \{0,1\}^{nk}$ such that $h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0, \mathbf{v}_1) = h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0', \mathbf{v}_1')$. Then the following equalities are equivalent.

$$\begin{gathered} h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0, \mathbf{v}_1) = h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0', \mathbf{v}_1'), \\ \mathbf{G} \cdot h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0, \mathbf{v}_1) = \mathbf{G} \cdot h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0', \mathbf{v}_1') \pmod q, \\ \mathbf{A}_0\mathbf{v}_0 + \mathbf{A}_1\mathbf{v}_1 + \mathbf{B}_0\mathbf{t} + \mathbf{B}_1\mathbf{t} \pmod q = \mathbf{A}_0\mathbf{v}_0' + \mathbf{A}_1\mathbf{v}_1' + \mathbf{B}_0\mathbf{t} + \mathbf{B}_1\mathbf{t} \pmod q, \\ \mathbf{A}_0\mathbf{v}_0 + \mathbf{A}_1\mathbf{v}_1 \pmod q = \mathbf{A}_0\mathbf{v}_0' + \mathbf{A}_1\mathbf{v}_1' \pmod q. \end{gathered}$$

This equality is equivalent to $\mathbf{A}\begin{pmatrix}\mathbf{v}_0 - \mathbf{v}_0' \\ \mathbf{v}_1 - \mathbf{v}_1'\end{pmatrix} = \mathbf{0} \pmod q$. Let $\mathbf{z} := \begin{pmatrix}\mathbf{v}_0 - \mathbf{v}_0' \\ \mathbf{v}_1 - \mathbf{v}_1'\end{pmatrix}$, then $\mathbf{z} \neq \mathbf{0}$ and $\mathbf{z} \in \{0,1\}^m$. Therefore, we have shown that $\mathbf{z}$ is a non-zero solution to $\mathsf{SIS}^{\infty}_{n,m,q,1}$ problem. □

3.1 The Parameterized Lattice-based Accumulator Scheme

Let $N = 2^{\ell}$ be a positive integer for some $\ell \in \mathbb{N}$. Let $\mathsf{bin}_{\ell}(\cdot)$ be the binary decomposition mapping an integer $i \in 0, \cdots, 2^{\ell} - 1$ to a bit string in $\{0,1\}^{\ell}$. For example, $011 \leftarrow \mathsf{bin}_3(3)$, while $11 \leftarrow \mathsf{bin}_2(3)$.

The accumulator we consider in this work is parametrized by a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times 2nk}$ and a vector $\mathbf{t} \in \{0,1\}^{nk}$, where $k = \lceil \log q \rceil$. We call it the *parameterized accumulator* (or PACC for short). The PACC works as follows:

$pp \leftarrow$ **PACC.Setup**(n)**:** On input a security parameter n, do:

1. Choose q. Let $k := \lceil \log_2 q \rceil$, $m := 2nk$, $N = 2^{\ell}$.
2. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and output $pp := \mathbf{A}$.

$\mathbf{v} \leftarrow$ **PACC.Acc**$_{pp}(\mathbf{B}, \mathbf{t}, R)$**:** On input public parameters pp and a list $R := \{\mathbf{p}_0, \cdots, \mathbf{p}_{N-1}\}$ with $N = 2^{\ell}$ for some ℓ, do:

1. For $i \in \{0, N-1\}$, assign $\mathbf{v}_{i_1,\cdots,i_\ell} \leftarrow \mathbf{p}_i$, where $(i_1,\cdots,i_\ell) \in \{0,1\}^\ell \leftarrow \mathsf{bin}_\ell(i)$.
2. Build a Merkle tree of depth ℓ whose leaves are $\mathbf{v}_{0,0,\cdots,0},\cdots,\mathbf{v}_{1,1,\cdots,1}$.
3. At depth $i \in [\ell]$, for $j \in \{0, i-1\}$, the value of the $(j+1)$-th node denoted by $\mathbf{v}_{j_1,\cdots,j_i}$, where $(j_1,\cdots,j_i) \in \{0,1\}^i \leftarrow \mathsf{bin}_i(j)$, can be computed as

$$\mathbf{v}_{j_1,\cdots,j_i} \leftarrow h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_{j_1,\cdots,j_i,0}, \mathbf{v}_{j_1,\cdots,j_i,1}).$$

4. At depth 0, the root $\mathbf{v} := \mathbf{v}_\epsilon \leftarrow h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{v}_0, \mathbf{v}_1)$.
5. Output the accumulator value $\mathbf{v}$.

$\textsf{wit} \leftarrow$ **$\mathsf{PACC.Witness}_{pp}(\mathbf{B}, \mathbf{t}, R, \mathbf{p})$:** On input public parameters pp, a ring $R := \{\mathbf{p}_0,\cdots,\mathbf{p}_{N-1}\}$, and $\mathbf{p}$, perform:

1. If $\mathbf{p} \notin R$, return $\perp$. Otherwise, we have $\mathbf{p} = \mathbf{p}_i$ for some $i \in \{0,\cdots,N-1\}$. Now, let $(i_1,\cdots,i_\ell) \in \{0,1\}^\ell \leftarrow \mathsf{bin}_\ell(i)$.
2. The witness for the fact $\mathbf{p} \in R$ is

$$\mathsf{wit} := \{(i_1,\cdots,i_\ell), (\mathbf{v}_{i_1,\cdots,i_{\ell-1},\overline{i_\ell}},\cdots,\mathbf{v}_{i_1,\overline{i_2}},\mathbf{v}_{\overline{i_1}})\},$$

where $\mathbf{v}_{i_1,\cdots,i_{\ell-1},\overline{i_\ell}},\cdots,\mathbf{v}_{i_1,\overline{i_2}},\mathbf{v}_{\overline{i_1}}$ are computed using $\mathsf{PACC.Acc}_{pp}(\mathbf{B},\mathbf{t},R)$.

$0/1 \leftarrow$ **$\mathsf{PACC.Verify}_{pp}(\mathbf{B},\mathbf{v},\mathbf{t},\mathbf{p},\mathsf{wit})$:** On input public parameters pp, an accumulator value, a witness $\mathsf{wit} := \{(i_1,\cdots,i_\ell),(\mathbf{w}_\ell,\cdots,\mathbf{w}_1)\}$ for $\mathbf{p}$, compute:

1. Assign $\mathbf{z}_\ell \leftarrow \mathbf{p}$. For $j \in \{\ell-1,\cdots,0\}$, compute

$$\mathbf{z}_j := \overline{i_{j+1}} \cdot h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{z}_{j+1},\mathbf{w}_{j+1}) + i_{j+1} \cdot h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{w}_{j+1},\mathbf{z}_{j+1}). \tag{1}$$

2. If $\mathbf{z}_0 = \mathbf{v}$, return 1. Otherwise, return 0.

The following theorem guarantees the security of the PACC.

Theorem 1. *Provided the hardness of the* $\mathsf{sda}_{\tilde{\mathcal{O}}(n)}$ *problem, the accumulator scheme* PACC *is secure.*

Proof. Suppose by contradiction that there is a PPT adversary $\mathcal{A}$ such that the probablity of the following event is non-negligible:

$$\begin{aligned}&\mathbf{A} \leftarrow \mathsf{PACC.Setup}(1^\lambda); (R, \mathbf{p}^*, \mathsf{wit}^*) \leftarrow \mathcal{A}(\mathbf{A}) :\\ &\quad \mathbf{p}^* \notin R \wedge \mathsf{PACC.Verify}_{pp}(\mathbf{B}, \mathsf{PACC.Acc}_{pp}(\mathbf{B},\mathbf{t},R), \mathbf{t}, \mathbf{p}^*, \mathsf{wit}^*) = 1.\end{aligned}$$

Here matrix $\mathbf{B}$ and vector $\mathbf{t}$ parametrise the PACC. Now, we construct an algorithm $\mathcal{B}$ that can break an SIS instance and hence break the $\mathsf{as}_{\tilde{\mathcal{O}}(n)}$ problem. Assume that $\mathcal{B}$ wants to solve the SIS instance given by matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. Now $\mathcal{B}$ sets $\mathbf{A}$ as the output of $\mathsf{PACC.Setup}(1^\lambda)$ then sends it to $\mathcal{A}$. Finaly $\mathcal{A}$ returns $(R, \mathbf{p}^*, \mathsf{wit}^*)$. Here, the witness $\mathsf{wit}^* := \{(i_1^*,\cdots,i_\ell^*),(\mathbf{w}_\ell,\cdots,\mathbf{w}_1)\}$ in which $(i_1^*,\cdots,i_\ell^*)$ is the binary expansion of some integer $i^* \in \{0,\cdots N-1\}$. Let $\mathbf{v}^* := \mathsf{PACC.Acc}_{pp}(\mathbf{B},\mathbf{t},R)$. Accordingly, we will have a path $[\mathbf{v}_{i_1^*,\cdots,i_\ell^*} = \mathbf{p}_{i^*} \rightarrow \mathbf{v}_{i_1^*,\cdots,i_{\ell-1}^*} \rightarrow \cdots \rightarrow \mathbf{v}_{i_1^*} \rightarrow \mathbf{v}^*]$ from the leave $\mathbf{p}_{i^*}$ to the root $\mathbf{v}^*$ of the

Merkle tree formed through the execution of $\mathbf{v}^* \leftarrow \mathsf{PACC.Acc}_{pp}(\mathbf{B}, \mathbf{t}, R)$. However, through the execution of $\mathsf{PACC.Verify}_{pp}(\mathbf{B}, \mathbf{v}^*, \mathbf{t}, \mathbf{p}^*, \mathsf{wit}^*) = 1$, we will have the path $[\mathbf{z}_\ell = \mathbf{p}^* \to \mathbf{z}_{\ell-1} \to \cdots \to \mathbf{z}_1 \to \mathbf{z}_0 = \mathbf{v}^*]$. Notice that $\mathbf{p}^* \notin R$. Thus, $\mathbf{p}^* \neq \mathbf{p}_{i^*}$. This implies by comparing these above two paths that there is the smallest integer $k \in [\ell]$ satisfying that $\mathbf{z}_k \neq \mathbf{v}_{i_1^*,\cdots,i_k^*}$. Therefore, there will be a collision for the hash function $h_{\mathbf{A},\mathbf{B},\mathbf{t}}$ at the parent node of $\mathbf{v}_{i_1^*,\cdots,i_k^*}$. At this point, the theorem follows from Lemma 3 □

4 Lattice-based Unique Ring Signature from Accumulator

4.1 The Unique Ring Signature Construction

We present a construction of URS from ring signatures based on accumulators. The key idea is to merge the unique tag (which is produced using the message, the ring, and the real signer's secret key) into the accumulator. The unique tag defined in our work is the binary decomposition of $H_{\mathsf{UT}}(\mu, R) \cdot \mathbf{x}$, in which $\mathbf{x}$ is a secret key. Note that we model the hash function H_{UT} as a random oracle. The URS construction is described below.

$urs.pp \leftarrow$ **URS.Setup**(n)**:** On input a security parameter n, do:

1. Choose q. Let $k := \lceil \log_2 q \rceil$, $m = 2nk$.
2. Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, choose hash functions H_{UT} and H_{FS}, and output
$$urs.pp := ((n, m, q, k, \mathbf{A}), H_{\mathsf{UT}}, H_{\mathsf{FS}}).$$
Here, $H_{\mathsf{UT}} : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ which will be modeled as a random orcale.
3. Consider the weak pseudorandom function F presented in Section 2.6.

$(\mathbf{x}, \mathbf{p}) \leftarrow$ **URS.Key**$(urs.pp)$**:** On input public parameters $urs.pp$, choose $\mathbf{x} \xleftarrow{\$} \{0,1\}^m$ then compute $\mathbf{p} = \mathsf{bin}(\mathbf{A}\mathbf{x} \pmod{q}) \in \{0,1\}^{nk}$, and output $(sk, pk) = (\mathbf{x}, \mathbf{p})$.

sig $\leftarrow$ **URS.Sign**$(urs.pp, sk, \mu, R)$**:** On input public parameters $urs.pp := \mathbf{A}$, the secret key $\mathsf{sk} = \mathbf{x}$ for the real signer (with respect to the public key $\mathbf{p} := \mathsf{bin}(\mathbf{A}\mathbf{x} \pmod{q}))$ belonging to the ring $R := \{\mathbf{p}_1, \cdots, \mathbf{p}_{N-1}\}$, a message μ, perform:

1. Compute $\mathbf{B} = [\mathbf{B}_0|\mathbf{B}_1] \leftarrow H_{\mathsf{UT}}(\mu, R)$.
2. Compute the unique tag $\mathbf{t} := \mathsf{bin}(\mathsf{F}_{\mathbf{x}}(\mathbf{B}) \pmod{p}))$.
3. Let $acc.pp := \mathbf{A}$.
4. Run $\mathbf{v} \leftarrow \mathsf{PACC.Acc}_{acc.pp}(\mathbf{B}, \mathbf{t}, R)$ using the hash function $h_{\mathbf{A},\mathbf{B}}$.
5. Run $\mathsf{wit} \leftarrow \mathsf{PACC.Witness}_{acc.pp}(\mathbf{B}, \mathbf{t}, R, \mathbf{p})$ where
$$\mathsf{wit} := \{(i_1, \cdots, i_\ell) \in \{0,1\}^\ell, (\mathbf{w}_\ell, \cdots, \mathbf{w}_1) \in (\{0,1\}^{nk})^\ell\}.$$
6. Use the Fiat-Shamir Heuristic with the hash function H_{FS} to transform the ZKAoK in Figure 6 to a non-interactive ZKAoK protocol $\mathsf{NIZKAoK}$.

The $\mathsf{NIZKAoK}$ protocol is repeated up to $\kappa = \omega(\log n)$ (to get a negligible soundness error) on input $(\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t})$ and the prover's witness $(\mathbf{x}, \mathbf{p}, \mathsf{wit})$ to produce a transcript $\mathit{\Pi}_{\mathsf{urs}} := (\{\mathsf{CMT}_j\}_{j=1}^{\kappa}, \{\mathsf{CH}_j\}_{j=1}^{\kappa}, \{\mathsf{RSP}_j\}_{j=1}^{\kappa}, \mathbf{t})$, where

$$\mathsf{CH}_j := H_{\mathsf{FS}}(\mu, \mathsf{CMT}_j, \mathbf{A}, \mathbf{v}, R, \mathbf{B}, \mathbf{t}) \in \{1, 2, 3\}.$$

7. Output $\mathsf{sig} := \mathit{\Pi}_{\mathsf{urs}}$.

$0/1 \leftarrow$ **URS.Verify**$(urs.pp, \mu, R, \mathsf{sig})$**:** On input public parameters $\mathsf{urs.pp}$, a message μ, a ring of signers R and a signature sig, compute:

1. Compute $\mathbf{B} = [\mathbf{B}_0|\mathbf{B}_1] \leftarrow H_{\mathsf{UT}}(\mu, R)$. Let $acc.pp := (n, m, q, k, \mathbf{A})$.
2. Run $\mathbf{v} \leftarrow \mathsf{PACC.Acc}_{acc.pp}(\mathbf{B}, \mathbf{t}, R)$ using the hash function $h_{\mathbf{A},\mathbf{B},\mathbf{t}}$.
3. Parse $\mathsf{sig} = \mathit{\Pi}_{\mathsf{urs}} := (\{\mathsf{CMT}_i\}, \mathsf{CH}, \{\mathsf{RSP}_i\}, \mathbf{t})$.
   Return 0 if $\mathsf{CH} \neq H_{\mathsf{FS}}(\mu, \mathsf{CMT}, \mathbf{A}, \mathbf{v}, R, \mathbf{B}, \mathbf{t})$..
4. Run $\mathsf{ZKAoK.Verify}$ to check the validity of each tuple $(\mathsf{CMT}_i, \mathsf{CH}_i, \mathsf{RSP}_i)$. If any of them does not hold then return 0. Otherwise, return 1.

4.2 A ZKAoK for the Unique Ring Signatures

For the accumulator-based unique ring signatures from lattices, we consider the following relation:

$$\begin{aligned}\mathcal{R}_{\mathsf{URS}} := \{&(\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m} \times \{0,1\}^{nk} \times \{0,1\}^{nk};\\ &\mathbf{x} \in \{0,1\}^m, \mathbf{p} \in \{0,1\}^{nk}, \mathsf{wit} \in \{0,1\}^{\ell} \times (\{0,1\}^{nk})^{\ell} :\\ &\mathsf{PACC.Verify}_{\mathbf{A}}(\mathbf{B}, \mathbf{v}, \mathbf{t}, \mathbf{p}, \mathsf{wit}) = 1 \ \wedge \ \mathbf{A}\mathbf{x} = \mathbf{G}\mathbf{p} \ \wedge \ \mathsf{F}_{\mathbf{x}}(\mathbf{B}) = \mathbf{G}\mathbf{t}\}.\end{aligned}$$

We will design a ZKAoK for the relation $\mathcal{R}_{\mathsf{URS}}$. That is, the ZKAoK is to prove that a prover $\mathcal{P}$ knows a witness $(\mathbf{x}, \mathbf{p}, \mathsf{wit})$ for a given statement $(\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t})$ such that $((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}); \mathbf{x}, \mathbf{p}, \mathsf{wit}) \in \mathcal{R}_{\mathsf{URS}}$. Note that $\mathbf{v} = \mathbf{z}_0$, and $\mathbf{p} = \mathbf{z}_{\ell}$.

We introduce here some new notations:

- Let $\mathcal{B}_m^{nk} := \{\mathbf{x} = (x_1, \cdots, x_m) : \mathbf{x} \in \{0,1\}^m \wedge \|\mathbf{x}\|_1 = nk\}$ be the set of vectors in $\{0,1\}^m$ having Hamming weight nk. Here $\|\mathbf{x}\|_1 := \sum_{i=1}^m |x_i|$.
- Let $\mathcal{S}_m$ be the set of all permutations of m elements.
- Let $\mathsf{ext}(b, \mathbf{z}) := \begin{pmatrix} \overline{b} \cdot \mathbf{z} \\ b \cdot \mathbf{z} \end{pmatrix}$, and $\mathsf{dbl}(\mathbf{t}) := \begin{pmatrix} \mathbf{t} \\ \mathbf{t} \end{pmatrix}$.
- For $b \in \{0,1\}, \pi \in \mathcal{S}_m$, we denote by $T_{b,\pi}$ the permutation that transforms $\mathbf{w} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \end{pmatrix}$, where $\mathbf{y}_i \in \mathbb{Z}_q^m$, into $T_{b,\pi}(\mathbf{y}) = \begin{pmatrix} \pi(\mathbf{y}_b) \\ \pi(\mathbf{y}_{\overline{b}}) \end{pmatrix}$, where $\mathbf{z}_i \in \mathbb{Z}_q^m$.

Note that, for all $b, c \in \{0,1\}, \pi, \phi \in \mathcal{S}_m$ and all $\mathbf{z}, \mathbf{w} \in \{0,1\}^m$, the following equivalences hold:

$$\begin{cases} \widehat{\mathbf{z}} := \mathsf{ext}(c, \mathbf{z}) \wedge \mathbf{z} \in \mathcal{B}_m^{nk} & \Longleftrightarrow T_{b,\pi}(\widehat{\mathbf{z}}) = \mathsf{ext}(c \oplus b, \pi(\mathbf{z})) \wedge \pi(\mathbf{z}) \in \mathcal{B}_m^{nk}; \\ \widehat{\mathbf{w}} := \mathsf{ext}(c, \mathbf{w}) \wedge \mathbf{w} \in \mathcal{B}_m^{nk} & \Longleftrightarrow T_{b,\pi}(\widehat{\mathbf{w}}) = \mathsf{ext}(c \oplus b, \phi(\mathbf{w})) \wedge \phi(\mathbf{w}) \in \mathcal{B}_m^{nk}. \end{cases}$$

Now, we analyze the relation $\mathcal{R}_{\mathsf{URS}}$. We start with the condition

$$\mathsf{PACC.Verify}_{\mathbf{A}}(\mathbf{B}, \mathbf{v}, \mathbf{t}, \mathbf{p}, \mathsf{wit}) = 1.$$

From Equation (1), we have

$$\mathbf{G}\mathbf{z}_j := \overline{i_{j+1}} \cdot (\mathbf{A}_0\mathbf{z}_{j+1} + \mathbf{A}_1\mathbf{w}_{j+1} + \mathbf{B}_0\mathbf{t} + \mathbf{B}_1\mathbf{t}) + i_{j+1} \cdot (\mathbf{A}_0\mathbf{w}_{j+1} + \mathbf{A}_1\mathbf{z}_{j+1} + \mathbf{B}_0\mathbf{t} + \mathbf{B}_1\mathbf{t}),$$

which is equivalent to $\mathbf{G}\mathbf{z}_j := \mathbf{A} \cdot \mathsf{ext}(i_{j+1}, \mathbf{z}_{j+1}) + \mathbf{A} \cdot \mathsf{ext}(\overline{i_{j+1}}, \mathbf{w}_{j+1}) + \mathbf{B} \cdot \mathsf{dbl}(\mathbf{t})$. Let $\widehat{\mathbf{z}}_{j+1} := \mathsf{ext}(i_{j+1}, \mathbf{z}_{j+1})$, $\widehat{\mathbf{w}}_{j+1} := \mathsf{ext}(\overline{i_{j+1}}, \mathbf{w}_{j+1})$, and $\widehat{\mathbf{t}} := \mathsf{dbl}(\mathbf{t})$ we have

$$\begin{cases} \mathbf{G}\mathbf{z}_j & = \mathbf{A} \cdot \widehat{\mathbf{z}}_{j+1} + \mathbf{A} \cdot \widehat{\mathbf{w}}_{j+1} + \mathbf{B} \cdot \widehat{\mathbf{t}}, \forall j \in [\ell - 1] \\ \mathbf{G}\mathbf{v} & = \mathbf{A} \cdot \widehat{\mathbf{z}}_1 + \mathbf{A} \cdot \widehat{\mathbf{w}}_1 + \mathbf{B} \cdot \widehat{\mathbf{t}} \end{cases}.$$

We exploit the “extend-then-permute” technique for the ZKAoK. For doing that, we

- extend $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1]$ to $\mathbf{A}^* = [\mathbf{A}_0|\mathbf{0}_{n\times nk}|\mathbf{A}_1|\mathbf{0}_{n\times nk}]$, $\mathbf{A}$ to $\widehat{\mathbf{A}} := [\mathbf{A}|\mathbf{0}_{n\times m}]$, $\mathbf{B}$ to $\widehat{\mathbf{B}} := [\mathbf{B}|\mathbf{0}_{n\times m}]$, $\mathbf{G}$ to $\mathbf{G}^* := [\mathbf{G}|\mathbf{0}_{n\times nk}]$
- extend $\mathbf{z}_1, \cdots, \mathbf{z}_\ell$, $\mathbf{w}_1, \cdots, \mathbf{w}_\ell$ to $\mathbf{z}_1^*, \cdots, \mathbf{z}_\ell^*$, $\mathbf{w}_1^*, \cdots, \mathbf{w}_\ell^* \in \mathcal{B}_m^{nk}$, respectively. These vectors are extended by concatenating a length-nk vector of suitable Hamming weight.
- also, extend $\mathbf{x}$, $\widehat{\mathbf{t}}$ to $\mathbf{x}^*$, $\widehat{\mathbf{t}}^*$ by appending vector $\{0\}^m$, respectively.

We will a brief description of the ZAKoK for $\mathcal{R}_{\mathsf{URS}}$ aiming to the goals and the strategies that a prover $\mathcal{P}$ would like to perform.

Common inputs: $(\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t})$, where $\mathbf{A}$ is extended to $\mathbf{A}^*$ and $\widehat{\mathbf{A}}$, while $\mathbf{B}$ is extended to $\widehat{\mathbf{B}}$ as above.

$\mathcal{P}$'s inputs: $(\mathbf{x}, \mathbf{p}, \mathsf{wit})$, where $\mathsf{wit} := \{(i_1, \cdots, i_\ell), (\mathbf{w}_\ell, \cdots, \mathbf{w}_1))\}$.

$\mathcal{P}$'s goal: $\mathcal{P}$ proves in zero knowledge that it knows that

Goal 1. $\mathbf{z}_i^*$, $\mathbf{w}_i^* \in \mathcal{B}_m^{nk}$, $\widehat{\mathbf{z}}_j^* = \mathsf{ext}(i_j, \mathbf{z}_j^*)$, $\widehat{\mathbf{w}}_j^* = \mathsf{ext}(\overline{i_j}, \mathbf{w}_j^*)$ for all $i \in [\ell]$; and that

Goal 2. the following equations hold:

$$\begin{cases} \forall j \in [\ell - 1], \mathbf{A}^* \cdot \widehat{\mathbf{z}}_{j+1}^* + \mathbf{A}^* \cdot \widehat{\mathbf{w}}_{j+1}^* + \mathbf{B} \cdot \widehat{\mathbf{t}} & = \mathbf{G}^*\mathbf{z}_j^* \pmod q \\ \mathbf{A}^* \cdot \widehat{\mathbf{z}}_1^* + \mathbf{A}^* \cdot \widehat{\mathbf{w}}_1^* + \mathbf{B} \cdot \widehat{\mathbf{t}} & = \mathbf{G}\mathbf{v} \pmod q \\ \widehat{\mathbf{A}} \cdot \mathbf{x}^* & = \mathbf{G}^*\mathbf{z}_\ell^* = \mathbf{G}\mathbf{p} \pmod q \\ \mathsf{F}_{\mathbf{x}^*}(\widehat{\mathbf{B}}) & = \mathbf{G}\mathbf{t} \pmod p \end{cases} \tag{2}$$

Techniques/Strategies for Prover $\mathcal{P}$:

For Goal 1: For each $j \in [\ell]$, $\mathcal{P}$ samples permutations $\pi_j, \phi_j \xleftarrow{\$} \mathcal{S}_m$ and $b_j \xleftarrow{\$} \{0, 1\}$ then it shows that

$$\pi_j(\mathbf{z}_j^*) \in \mathcal{B}_m^{nk} \wedge T_{b_j,\pi_j}(\widehat{\mathbf{z}}_j^*) = \mathsf{ext}(i_j \oplus b_j, \pi_j(\mathbf{z}_j^*))$$
$$\phi_i(\mathbf{w}_j^*) \in \mathcal{B}_m^{nk} \wedge T_{b_j,\pi_j}(\widehat{\mathbf{w}}_j^*) = \mathsf{ext}(i_j \oplus b_j, \phi_j(\mathbf{w}_j^*))$$

For Goal 2: $\mathcal{P}$ uniformly samples random masking vectors $\mathbf{r}_{\mathbf{z}}^{(1)}, \cdots, \mathbf{r}_{\mathbf{z}}^{(\ell)}, \xleftarrow{\$} \mathbb{Z}_q^m$; $\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{z}}}^{(\ell)}$; $\mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{w}}}^{(\ell)}$; $\mathbf{r}_{\mathbf{x}}^{(A)}$; $\mathbf{r}_{\mathbf{x}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^{2m}$; $\mathbf{r}_{\mathbf{e}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^n$.
We can transform last equation in Equation (2) as $\lfloor \widehat{\mathbf{B}}\mathbf{x}^* \rceil_p = \mathbf{G}\mathbf{t} \pmod p$. Given $\gamma = p/q$, let $\mathbf{e} = \gamma \cdot \mathbf{G}\mathbf{t} - \widehat{\mathbf{B}}\mathbf{x}^*$, we have $\widehat{\mathbf{B}}\mathbf{x}^* + \mathbf{e} = \gamma \cdot \mathbf{G}\mathbf{t} \pmod q$, or equivalently

$$\widehat{\mathbf{B}}(\mathbf{x}^* + \mathbf{r}_{\mathbf{x}}^{(B)}) + (\mathbf{e} + \mathbf{r}_{\mathbf{e}}^{(B)}) = \gamma \cdot \mathbf{G}\mathbf{t} + \widehat{\mathbf{B}}\mathbf{r}_{\mathbf{x}}^{(B)} + \mathbf{r}_{\mathbf{e}}^{(B)} \pmod q.$$

That ís, $\mathcal{P}$ proves $\mathcal{V}$ that

$$\begin{cases} \mathbf{A}^*(\widehat{\mathbf{z}}_1^* + \mathbf{r}_{\widehat{\mathbf{z}}}^{(1)}) + \mathbf{A}^*(\widehat{\mathbf{w}}_1^* + \mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}) - \mathbf{G}\mathbf{v} + \mathbf{B} \cdot \widehat{\mathbf{t}} = \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{w}}}^{(1)} \pmod q; \\ \forall j \in [\ell - 1], \mathbf{A}^*(\widehat{\mathbf{z}}_{j+1} + \mathbf{r}_{\widehat{\mathbf{z}}}^{(j+1)}) + \mathbf{A}^*(\widehat{\mathbf{w}}_{j+1} + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j+1)}) + \mathbf{B} \cdot \widehat{\mathbf{t}} - \mathbf{G}^*(\mathbf{z}_j^* + \mathbf{r}_{\mathbf{z}}^{(j+1)}) \\ = \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{w}}}^{(j+1)} - \mathbf{G}^*\mathbf{r}_{\mathbf{z}}^{(j)} \pmod q; \\ \widehat{\mathbf{A}}(\mathbf{x}^* + \mathbf{r}_{\mathbf{x}}^{(A)}) - \mathbf{G}^*(\mathbf{z}_\ell^* + \mathbf{r}_{\mathbf{z}}^{(\ell)}) = \widehat{\mathbf{A}}\mathbf{r}_{\mathbf{x}}^{(A)} - \mathbf{G}^*\mathbf{r}_{\mathbf{z}}^{(\ell)} \pmod q; \\ \widehat{\mathbf{B}}(\mathbf{x}^* + \mathbf{r}_{\mathbf{x}}^{(B)}) + (\mathbf{e} + \mathbf{r}_{\mathbf{e}}^{(B)}) = \gamma \cdot \mathbf{G}\mathbf{t} + \widehat{\mathbf{B}}\mathbf{r}_{\mathbf{x}}^{(B)} + \mathbf{r}_{\mathbf{e}}^{(B)} \pmod q. \end{cases} .$$

The following lemma says that there exists a ZKAoK for the relation $\mathcal{R}_{\mathsf{URS}}$. The ZKAoK given in Figure 6 is a Stern type one [Ste96] which is a 2-Sigma protocol enjoying 3-special soundness. That is, we need up to 3 transcripts in order to be able to extract the witness.

Lemma 4. *Assume that the* $\mathsf{SIS}_{m,n,q,\theta}^{\infty}$ *problem is hard. Then there exists a statistical* ZKAoK *for the relation* $\mathcal{R}_{\mathsf{URS}}$ *with perfect completeness and communication cost* $\tilde{O}(\ell \cdot n)$*. In particular:*

- *There exists an efficient simulator that, on input* $(\mathbf{A}, \mathbf{v})$*, outputs an accepting transcript which is statistically close to that produced by the real prover.*
- *There exists an efficient knowledge extractor that, on input 3 valid responses* $(\mathsf{RSP}_1, \mathsf{RSP}_2, \mathsf{RSP}_3)$ *to the same commitment* CMT*, outputs* $(\mathbf{x}', \mathbf{p}', \mathsf{wit}')$ *such that* $((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}), \mathbf{x}', \mathbf{p}', \mathsf{wit}') \in \mathcal{R}_{\mathsf{URS}}$.

4.3 Analysis of the ZKAoK for the Relation $\mathcal{R}_{\mathsf{URS}}$

Theorem 2 (Completeness and Communication Cost). *The interactive protocol described in Figure 6 is perfectly complete and costs* $\tilde{\mathcal{O}}(\ell \cdot n)$ *bits for communication. It is a statistical zero-knowledge argument of knowledge if the string commitment* COM *is statistically hiding and computationally binding.*

Follows section 3.1, one can easilly check that:

$$\Pr[\mathcal{P}((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}), \mathbf{x}, \mathbf{p}, \mathsf{wit}), \mathcal{V}(\mathbf{B}, \mathbf{v}, \mathbf{t}, \mathbf{p}, \mathsf{wit}) = 1] = 1,$$

where $((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}), \mathbf{x}, \mathbf{p}, \mathsf{wit}) \in \mathcal{R}_{\mathsf{URS}}$, that means an honest the prover $\mathcal{P}$ always successfully convinces the verifier $\mathcal{V}$. To compare to the accumulator in [LLNW16], our approach adds extra information of $(\eta, \mathbf{r}_{\mathbf{x}}^{(B)}, \mathbf{B}, \mathbf{t})$ which just costs marginally larger. Hence, the communication cost of our protocol is of order $\tilde{\mathcal{O}}(\ell \cdot m \cdot \log_q) = \tilde{\mathcal{O}}(\ell \cdot n)$ bits.

Commitment. $\mathcal{P}$ performs:

1. Samples randomnesses ρ_1, ρ_2, ρ_3 for COM
2. For $j \in [\ell]$, sample $\pi_j, \phi_j \xleftarrow{\$} \mathcal{S}_m$; $\tau, \eta \xleftarrow{\$} \mathcal{S}_{2m}; \zeta \xleftarrow{\$} \mathcal{S}_n; b_j \xleftarrow{\$} \{0,1\}$.
3. Sample random masking vectors $\mathbf{r}_{\mathbf{z}}^{(1)}, \cdots, \mathbf{r}_{\mathbf{z}}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^m$;
   $\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{z}}}^{(\ell)}; \mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{w}}}^{(\ell)}; \mathbf{r}_{\mathbf{x}}^{(A)}; \mathbf{r}_{\mathbf{x}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^{2m}; \mathbf{r}_{\mathbf{e}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^n$.
4. Compute comitment $\mathsf{CMT} = (C_1, C_2, C_3)$, where
   (i) $C_1 := \mathsf{COM}(\{b_j, \pi_j, \phi_j\}_{j=1}^{\ell}; \tau; \eta; \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}; \widehat{\mathbf{A}}\mathbf{r}_{\mathbf{x}}^{(A)} - \mathbf{G}^*\mathbf{r}_{\mathbf{z}}^{(\ell)};$
   $\gamma \cdot \mathbf{G}\mathbf{t} + \widehat{\mathbf{B}}\mathbf{r}_{\mathbf{x}}^{(B)} + \mathbf{r}_{\mathbf{e}}^{(B)}; \{\mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{w}}}^{(j+1)} - \mathbf{G}^*\mathbf{r}_{\mathbf{z}}^{(j+1)}\}_{j=1}^{\ell-1}; \rho_1)$
   (ii) $C_2 := \mathsf{COM}(\{\pi_j(\mathbf{r}_{\mathbf{z}}^{(j)}); T_{b_j,\pi_j}(\mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}); T_{\bar{b}_j,\phi_j}(\mathbf{r}_{\widehat{\mathbf{z}}}^{(j)})\}_{j=1}^{\ell}; \tau(\mathbf{r}_{\mathbf{x}}^{(A)}), \eta(\mathbf{r}_{\mathbf{x}}^{(B)}),$
   $\zeta(\mathbf{r}_{\mathbf{e}}^{(B)}); \rho_2)$
   (iii) $C_3 := \mathsf{COM}(\{\pi_j(\mathbf{z}_j^* + \mathbf{r}_{\mathbf{z}}^{(j)}); T_{b_j,\pi_j}(\widehat{\mathbf{z}_j}^* + \mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}); T_{\bar{b}_j,\phi_j}(\widehat{\mathbf{w}_j}^* + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)})\}_{j=1}^{\ell};$
   $\tau(\mathbf{r}_{\mathbf{x}}^{(A)} + \mathbf{x}^*); \eta(\mathbf{r}_{\mathbf{x}}^{(B)} + \mathbf{x}^*); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)} + \mathbf{e}); \rho_3)$

Challenge. $\mathcal{V}$ chooses a challenge $\mathsf{CH} \xleftarrow{\$} \{1,2,3\}$ and sends back to $\mathcal{P}$.

Response. What $\mathcal{P}$ responds will depend on the value of CH. Namely,

1. If $\mathsf{CH} = 1$: Let $\mathbf{a}_{\mathbf{x}}^{(A)} := \tau(\mathbf{x}^*)$, $\mathbf{a}_{\mathbf{x}}^{(B)} := \eta(\mathbf{x}^*)$, $\mathbf{b}_{\mathbf{x}}^{(A)} := \tau(\mathbf{r}_{\mathbf{x}}^{(A)})$, $\mathbf{b}_{\mathbf{x}}^{(B)} := \eta(\mathbf{r}_{\mathbf{x}}^{(B)})$, $\mathbf{a}_{\mathbf{e}}^{(B)} := \zeta(\mathbf{e}); \mathbf{b}_{\mathbf{e}}^{(B)} := \zeta(\mathbf{r}_{\mathbf{e}}^{(B)})$ and for each $j \in [\ell]$, compute:
   $$\begin{cases} a_j := i_j \oplus b_j; \mathbf{a}_{\mathbf{z}}^{(j)} := \pi_j(\mathbf{z}_j^*); \mathbf{a}_{\mathbf{w}}^{(j)} := \phi_j(\mathbf{w}_j^*); \\ \mathbf{b}_{\mathbf{z}}^{(j)} := \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)}); \mathbf{b}_{\widehat{\mathbf{z}}}^{(j)} := T_{b_j,\pi_j}(\mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}); \mathbf{b}_{\widehat{\mathbf{w}}}^{(j)} := T_{\overline{b_j},\phi_j}(\mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}). \end{cases} .$$
   Set $\mathsf{RSP} := (\{a_j; \mathbf{a}_{\mathbf{z}}^{(j)}; \mathbf{a}_{\mathbf{w}}^{(j)}; \mathbf{b}_{\mathbf{z}}^{(j)}; \mathbf{b}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{b}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \mathbf{a}_{\mathbf{x}}^{(A)}; \mathbf{b}_{\mathbf{x}}^{(A)}; \mathbf{a}_{\mathbf{x}}^{(B)}; \mathbf{b}_{\mathbf{x}}^{(B)};$
   $\mathbf{a}_{\mathbf{e}}^{(B)}, \mathbf{b}_{\mathbf{e}}^{(B)}; \rho_2; \rho_3)$
2. If $\mathsf{CH} = 2$: Let $\hat{\tau} := \tau; \hat{\eta} := \eta; \hat{\zeta} := \zeta; \mathbf{c}_{\mathbf{x}}^{(A)} := \mathbf{x}^* + \mathbf{r}_{\mathbf{x}}^{(A)}; \mathbf{c}_{\mathbf{x}}^{(B)} := \mathbf{x}^* + \mathbf{r}_{\mathbf{x}}^{(B)}; \mathbf{c}_{\mathbf{e}}^{(B)} := \mathbf{e} + \mathbf{r}_{\mathbf{e}}^{(B)}$ and for each $j \in [\ell]$, compute:
   $c_j = b_j; \hat{\pi}_j := \pi_j; \hat{\phi}_j := \phi_j; \mathbf{c}_{\mathbf{z}}^{(j)} := \mathbf{z}_j^* + \mathbf{r}_{\mathbf{z}}^{(j)}; \mathbf{c}_{\widehat{\mathbf{z}}}^{(j)} := \widehat{\mathbf{z}}_j^* + \mathbf{r}_{\widehat{\mathbf{z}}}^{(j)};$
   $\mathbf{c}_{\widehat{\mathbf{w}}}^{(j)} := \widehat{\mathbf{w}}_j^* + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}$.
   Set $\mathsf{RSP} := (\{c_j; \hat{\pi}_j; \hat{\phi}_j; \mathbf{c}_{\mathbf{z}}^{(j)}; \mathbf{c}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{c}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \hat{\tau}; \mathbf{c}_{\mathbf{x}}^{(A)}; \hat{\eta}; \mathbf{c}_{\mathbf{x}}^{(B)}; \hat{\zeta}; \mathbf{c}_{\mathbf{e}}^{(B)}; \rho_1; \rho_3)$.
3. If $\mathsf{CH} = 3$ Let $\tilde{\tau} := \tau$; $\tilde{\eta} := \eta$; $\tilde{\zeta} := \zeta$; $\mathbf{g}_{\mathbf{x}}^{(A)} := \mathbf{r}_{\mathbf{x}}^{(A)}$; $\mathbf{g}_{\mathbf{x}}^{(B)} := \mathbf{r}_{\mathbf{x}}^{(B)}$; $\mathbf{g}_{\mathbf{e}}^{(B)} := \mathbf{r}_{\mathbf{e}}^{(B)}$ and for each $j \in [\ell]$, compute:
   $g_j := b_j; \tilde{\pi}_j := \pi_j; \tilde{\phi}_j := \phi_j; \mathbf{g}_{\mathbf{z}}^{(j)} := \mathbf{r}_{\mathbf{z}}^{(j)}; \mathbf{g}_{\widehat{\mathbf{z}}}^{(j)} := \mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{g}_{\widehat{\mathbf{w}}}^{(j)} := \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)};$
   Set $\mathsf{RSP} := (\{g_j; \tilde{\pi}_j; \tilde{\phi}_j; \mathbf{g}_{\mathbf{z}}^{(j)}; \mathbf{g}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{g}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \tilde{\tau}; \mathbf{g}_{\mathbf{x}}^{(A)}; \tilde{\eta}; \mathbf{g}_{\mathbf{x}}^{(B)}; \tilde{\zeta}; \mathbf{g}_{\mathbf{e}}^{(B)}; \rho_1; \rho_2)$.

ZKAoK.Verify. Upon receiving RSP, $\mathcal{V}$ verifies it following below cases:

1. If $\mathsf{CH} = 1$: Parse RSP. Check that $\mathbf{a}_{\mathbf{x}}^{(A)} \in \mathcal{B}_{2m}^m$; $\mathbf{a}_{\mathbf{x}}^{(B)} \in \mathcal{B}_{2m}^m$. Check that:
   (i) $C_2 := \mathsf{COM}(\{\mathbf{b}_{\mathbf{z}}^{(j)}; \mathbf{b}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{b}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \mathbf{b}_{\mathbf{x}}^{(A)}; \mathbf{b}_{\mathbf{x}}^{(B)}; \mathbf{b}_{\mathbf{e}}^{(B)}; \rho_2)$
   (ii) $C_3 := \mathsf{COM}(\{\mathbf{a}_{\mathbf{z}}^{(j)} + \mathbf{b}_{\mathbf{z}}^{(j)}; \mathbf{a}_{\widehat{\mathbf{z}}}^{(j)} + \mathbf{b}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{a}_{\widehat{\mathbf{w}}}^{(j)} + \mathbf{b}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \mathbf{a}_{\mathbf{x}}^{(A)} + \mathbf{b}_{\mathbf{x}}^{(A)};$
   $\mathbf{a}_{\mathbf{x}}^{(B)} + \mathbf{b}_{\mathbf{x}}^{(B)}; \mathbf{a}_{\mathbf{e}}^{(B)} + \mathbf{b}_{\mathbf{e}}^{(B)}; \rho_3)$
2. If $\mathsf{CH} = 2$ Parse RSP. Check that:
   (i) $C_1 := \mathsf{COM}(\{c_j, \hat{\pi}_j, \hat{\phi}_j\}_{j=1}^{\ell}; \hat{\tau}; \hat{\eta}; \mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{w}}}^{(1)} - \mathbf{G}\mathbf{v} + \widehat{\mathbf{B}}\mathbf{t};$
   $\{\mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{w}}}^{(j+1)} - \mathbf{G}^*\mathbf{c}_{\mathbf{z}}^{(j+1)}\}_{j=1}^{\ell-1}; \widehat{\mathbf{A}}\mathbf{c}_{\mathbf{x}}^{(A)} - \mathbf{G}^*\mathbf{c}_{\mathbf{z}}^{(\ell)}; \widehat{\mathbf{B}}\mathbf{c}_{\mathbf{x}}^{(B)} + \mathbf{c}_{\mathbf{e}}^{(B)}; \rho_1)$
   (ii) $C_3 := \mathsf{COM}(\{\hat{\pi}_j(\mathbf{c}_{\mathbf{z}}^{(j)}); T_{c_j,\hat{\pi}_j}(\mathbf{c}_{\widehat{\mathbf{z}}}^{(j)}); T_{\overline{c}_j,\hat{\phi}_j}(\mathbf{c}_{\widehat{\mathbf{w}}}^{(j)})\}_{j=1}^{\ell}; \hat{\tau}(\mathbf{c}_{\mathbf{x}}^{(A)}); \hat{\eta}(\mathbf{c}_{\mathbf{x}}^{(B)});$
   $\hat{\zeta}(\mathbf{c}_{\mathbf{e}}^{(B)}); \rho_3)$
3. If $\mathsf{CH} = 3$ Parse RSP. Check that:
   (i) $C_1 := \mathsf{COM}(\{g_j; \tilde{\pi}_j; \tilde{\phi}_j\}_{j=1}^{\ell}; \tilde{\tau}; \tilde{\eta}; \mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{w}}}^{(1)};$
   $\{\mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{w}}}^{(i+1)} - \mathbf{G}^*\mathbf{g}_{\mathbf{z}}^{(j)}\}_{j=1}^{\ell-1}; \widehat{\mathbf{A}}\mathbf{g}_{\mathbf{x}}^{(A)} - \mathbf{G}^*\mathbf{g}_{\mathbf{z}}^{(\ell)}; \gamma \cdot \mathbf{G}\mathbf{t} + \widehat{\mathbf{B}}\mathbf{g}_{\mathbf{x}}^{(B)} + \mathbf{g}_{\mathbf{e}}^{(B)}; \rho_1)$
   (ii) $C_2 := \mathsf{COM}(\{\hat{\pi}_j(\mathbf{g}_{\mathbf{z}}^{(j)}); T_{g_j,\hat{\pi}_j}(\mathbf{g}_{\widehat{\mathbf{z}}}^{(j)}); T_{\overline{g}_j,\hat{\phi}_j}(\mathbf{g}_{\widehat{\mathbf{w}}}^{(j)})\}_{j=1}^{\ell}; \tilde{\tau}(\mathbf{g}_{\mathbf{x}}^{(A)}); \tilde{\eta}(\mathbf{g}_{\mathbf{x}}^{(B)});$
   $\tilde{\zeta}(\mathbf{g}_{\mathbf{e}}^{(B)}); \rho_2)$

If all conditions hold, $\mathcal{V}$ returns *accepted*. Otherwise, $\mathcal{V}$ returns *rejected*.

Fig. 6: The ZKAoK for the relation $\mathcal{R}_{\mathsf{URS}}$. Here COM is the string commitment scheme introduced in Section 2.4

Lemma 5 (Zero-Knowledge Property). *The interactive protocol described in Figure 6 is a statistical zero-knowledge argument, that is* $\mathsf{Adv}^{\mathsf{zk}}_{(\mathcal{P},\mathcal{V})}(\mathcal{V}^*) \leq \mathsf{negl}(\lambda)$*, if the string commitment* COM *is statistically hiding.*

Proof. The proof is given in Appendix A.1.

Lemma 6 (Argument of Knowledge (i.e., Soundness)). *Suppose the string commitment* COM *is computationally biding, there exists a knowledge extractor* $\mathcal{K}$ *that takes input as a commitment* CMT *and its valid reponses* $(\mathsf{RSP}_1, \mathsf{RSP}_2, \mathsf{RSP}_3)$ *then outputs* $(\mathbf{x}^*, \mathbf{p}^*, \mathsf{wit}^*)$ *such that* $((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}), \mathbf{x}^*, \mathbf{p}^*, \mathsf{wit}^*) \in \mathcal{R}_{\mathsf{URS}}$*. That is,* $\mathsf{Adv}^{\mathsf{sound}}_{(\mathcal{P},\mathcal{V})}(\mathcal{P}^*) \leq \mathsf{negl}(\lambda)$.

Proof. The proof is given in Appendix A.2.

4.4 Analysis of the Unique Ring Signature Scheme

Correctness. The completeness of the underlying ZKAoK protocol described in Figure 6 directly implies the correctness of the corresponding unique ring signature. An honest ring member's signature is always accepted by the verification algorithm since he can efficiently produce a tuple $(\mathbf{x}, \mathbf{p}, \mathsf{wit})$ such that

$$((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}), \mathbf{x}, \mathbf{p}, \mathsf{wit}) \in \mathcal{R}_{\mathsf{URS}}.$$

Efficiency. The signature bit-size of the given unique ring signature is of order $\tilde{\mathcal{O}}(\log N \cdot n)$ as the communication cost of the underlying ZKAoK protocol is of order $\tilde{\mathcal{O}}(\ell \cdot n)$.

Theorem 3 (Unforgeability). *In the random oracle model, the unique ring signature scheme given in Section 4 is unforgeable with respect to insider corruption under the hardness of the* $\mathsf{SIS}^{\infty}_{m,n,q,\theta}$ *problem.*

Theorem 4 (Anonymity). *In the random oracle model, the unique ring signature scheme is statistically anonymous under the zero-knowledge of the underlying ZKAoK protocol and the hardness of the decision-*$\mathsf{LWR}_{n,m,q,p}$ *problem.*

Theorem 5 (Uniqueness). *In the random oracle model, the unique ring signature scheme provides uniqueness against a probabilistic polynomially bounded adversary under the zero-knowledge property, the soundness of the underlying ZKAoK protocol, and the hardness of the decision-*$\mathsf{LWR}_{n,m,q,p}$ *problem.*

Theorem 6 (Non-colliding property). *The unique ring signature scheme given in Section 4 is non-colliding under the hardness of decisional-*$\mathsf{LWR}_{n,q,p}$ *problem with* $m \geq n(\log q + 1)(\log p - 1)$.

We provide the proof of these theorems in Appendix B.

5 Concrete Parameters

We choose the parameters m, n, q, p such that $m \geq n(\log q + 1)(\log p - 1)$ and $\theta = 1$ to ensure that the SIS and LWR problems are computationally hard. The security level for all parameter sets is for the root Hermite factor $\delta \approx 1.007$.

The concrete parameters for our URS scheme are provided in Table 2. Since our URS scheme has a signature size that is logarithmic to the number of ring members, the signature size gradually grows when this number increases.

Table 2: Concrete instantiations of URS scheme.

	Parameters					Size in MB	
Number of ring users (N)	m	n	q	θ	p	Public key	Signature
16	4608	128	2^{18}	1	4	2.4	7.85
32	4608	128	2^{18}	1	4	2.4	9.53
64	4608	128	2^{18}	1	4	2.4	11.2
128	4608	128	2^{18}	1	4	2.4	12.89
256	4608	128	2^{18}	1	4	2.4	14.56

6 Conclusions

In this paper, we present the first URS based on post-quantum hardness assumptions with logarithmic signature size. We showed that our scheme enjoys anonymity, unforgeability and unique properties in the random oracle model under the SIS and LWR assumptions. Since we only prove our URS scheme in ROM, we leave the proof in the quantum random oracle model as an open problem.

7 Acknowledgements

We are grateful to the ESORICS 2022 anonymous reviewers for their helpful comments. This work is partially supported by the Australian Research Council Linkage Project LP190100984. Dung Duong is also partially suported by the RevITAlise (RITA) Research Grants from University of Wollongong. Huy Quoc Le has been sponsored by a CSIRO Data61 PhD Scholarship and CSIRO Data61 Top-up Scholarship.

References

Ajt96. M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM.

BCC$^+$16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 327–357, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

BDH$^+$19. Michael Backes, Nico Döttling, Lucjan Hanzlik, Kamil Kluczniak, and Jonas Schneider. Ring signatures: Logarithmic-size, no setup—from standard assumptions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 281–311, Cham, 2019. Springer International Publishing.

BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 464–492, Cham, 2020. Springer International Publishing.

BLO18. Carsten Baum, Huang Lin, and Sabine Oechsner. Towards practical lattice-based one-time linkable ring signatures. In David Naccache, Shouhuai Xu, Sihan Qing, Pierangela Samarati, Gregory Blanc, Rongxing Lu, Zonghua Zhang, and Ahmed Meddahi, editors, *Information and Communications Security*, pages 303–322, Cham, 2018. Springer International Publishing.

BPR12. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 719–737, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

BPVY01. Ernest Brickell, David Pointcheval, Serge Vaudenay, and Moti Yung. Design validations for discrete logarithm based signature schemes. volume 1751, 09 2001.

CGH$^+$21. Rohit Chatterjee, Sanjam Garg, Mohammad Hajiabadi, Dakshita Khurana, Xiao Liang, Giulio Malavolta, Omkant Pandey, and Sina Shiehian. Compact ring signatures from learning with errors. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 282–312, Cham, 2021. Springer International Publishing.

ESS$^+$19. Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 67–88, Cham, 2019. Springer International Publishing.

FLWL20. Hanwen Feng, Jianwei Liu, Qianhong Wu, and Ya-Nan Li. Traceable ring signatures with post-quantum security. In Stanislaw Jarecki, editor, *Topics in Cryptology – CT-RSA 2020*, pages 442–468, Cham, 2020. Springer International Publishing.

FS07. Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography – PKC 2007*, pages 181–200, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

FZ12. M. Franklin and Haibin Zhang. A framework for unique ring signatures. *IACR Cryptol. ePrint Arch.*, 2012:577, 2012.

FZ13. Matthew Franklin and Haibin Zhang. Unique ring signatures: A practical construction. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, pages 162–170, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

HM96. Shai Halevi and Silvio Micali. Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. In Neal Koblitz, editor, *Advances in Cryptology — CRYPTO '96*, pages 201–215, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

KTX08. Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008*, pages 372–389, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

LAZ19. Xingye Lu, Man Ho Au, and Zhenfei Zhang. Raptor: A practical lattice-based (linkable) ring signature. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 110–130, Cham, 2019. Springer International Publishing.

Lin03. Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. *Journal of Cryptology*, 16:143–184, 2003.

LLNW16. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 1–31, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

LW05. Joseph K. Liu and Duncan S. Wong. Linkable ring signatures: Security models and new schemes. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, pages 614–623, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

LWW04. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *Information Security and Privacy*, pages 325–335, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

Lyu08. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In Ronald Cramer, editor, *Public Key Cryptography – PKC 2008*, pages 162–179, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

Lyu12. Vadim Lyubashevsky. Lattice Signatures Without Trapdoors. Cryptology ePrint Archive, Report 2011/537, Full version of paper appearing at Eurocrypt 2012, last revised 18 Oct 2017, 2012.

Mer16. Rebekah Mercer. Privacy on the blockchain: Unique ring signatures, 2016.

Nak08. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

RST01. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to Leak a Secret. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

Sho94. P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Nov 1994.

Ste96. J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

TKN+21. Anh The Ta, Thanh Xuan Khuc, Tuong Ngoc Nguyen, Huy Quoc Le, Dung Hoang Duong, Willy Susilo, Kazuhide Fukushima, and Shinsaku Kiyomoto. Efficient unique ring signature for blockchain privacy protection. In Joonsang Baek and Sushmita Ruj, editors, *Information Security and Privacy*, pages 391–407, Cham, 2021. Springer International Publishing.

YAL+17. Rupeng Yang, Man Ho Au, Junzuo Lai, Qiuliang Xu, and Zuoxia Yu. Lattice-based techniques for accountable anonymity: Composition of abstract stern's protocols and weak prf with efficient protocols from lwr. Cryptology ePrint Archive, Report 2017/781, 2017. https://ia.cr/2017/781.

A Proof of The Underlying Accumulator

A.1 Proof of Lemma 5

Proof. Let $\mathcal{S}$ be a PPT simulator which interacts with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$. $\mathcal{S}$ will choose a random value $\overline{\mathsf{CH}} \in \{1,2,3\}$. Depending on the challenge value chosen by $\widehat{\mathcal{V}}$ (possibly dishonest) and $\overline{\mathsf{CH}}$, we simulate the protocol as following scenarios. We denote by $\mathit{\Pi}_{\mathsf{urs}}^{(\mathsf{sim})} := (\mathsf{CMT}^{(\mathsf{sim})}, \mathsf{CH}, \mathsf{RSP}^{(\mathsf{sim})}, \mathbf{t})$ the simulated transcript.

Case $\overline{\mathsf{CH}} = 1$: $\mathcal{S}$ follows below steps:

1. Sample $\mathbf{x}'^*, \{\widehat{\mathbf{z}}'_j{}^*, \widehat{\mathbf{w}}'_j{}^*\}_{j=1}^{\ell} \in \mathbb{Z}_q^{2m}$, $\{\mathbf{z}'^*_j\}_{j=1}^{\ell} \in \mathbb{Z}_q^m$, $\mathbf{e}' \in \mathbb{Z}_q^n$ such that:

$$\begin{cases} \forall j \in [\ell-1], \mathbf{A}^* \cdot \widehat{\mathbf{z}}'^*_{j+1} + \mathbf{A}^* \cdot \widehat{\mathbf{w}}'^*_{j+1} + \mathbf{B} \cdot \widehat{\mathbf{t}} & = \mathbf{G}^* \mathbf{z}'^*_j \pmod q \\ \mathbf{A}^* \cdot \widehat{\mathbf{z}}'^*_1 + \mathbf{A}^* \cdot \widehat{\mathbf{w}}'^*_1 + \mathbf{B} \cdot \widehat{\mathbf{t}} & = \mathbf{G}\mathbf{v} \pmod q \\ \widehat{\mathbf{A}} \cdot \mathbf{x}'^* & = \mathbf{G}^* \mathbf{z}'^*_{\ell} = \mathbf{G}\mathbf{p} \pmod q \\ \mathsf{F}_{\mathbf{x}'^*}(\widehat{\mathbf{B}}) & = \mathbf{G}\mathbf{t} \pmod p \\ \widehat{\mathbf{B}}\mathbf{x}^* + \mathbf{e} & = \gamma \cdot \mathbf{G}\mathbf{t} \pmod q \end{cases}$$

2. Sample randomnesses ρ_1, ρ_2, ρ_3 for COM.
3. For $j \in [\ell]$, sample permutations $\pi_j, \phi_j \xleftarrow{\$} \mathcal{S}_m$; $\tau, \eta \xleftarrow{\$} \mathcal{S}_{2m}$; $\zeta \xleftarrow{\$} \mathcal{S}_n$ and $b_j \xleftarrow{\$} \{0,1\}$.
4. Sample random masking vectors $\mathbf{r}_{\mathbf{z}}^{(1)}, \cdots, \mathbf{r}_{\mathbf{z}}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^m$; $\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{z}}}^{(\ell)}; \mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{w}}}^{(\ell)}; \mathbf{r}_{\mathbf{x}}^{(A)}; \mathbf{r}_{\mathbf{x}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^{2m}; \mathbf{r}_{\mathbf{e}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^n$.
5. Send $\widehat{\mathcal{V}}$ a commiment $\mathsf{CMT}^{(\mathsf{sim})} := (C'_1, C'_2, C'_3)$ computed as below:

$$\begin{cases} C'_1 := \mathsf{COM}(\{b_j, \pi_j, \phi_j\}_{j=1}^{\ell}; \tau; \eta; \zeta; \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}; \widehat{\mathbf{A}}\mathbf{r}_{\mathbf{x}}^{(A)} - \mathbf{G}^*\mathbf{r}_{\mathbf{z}}^{(\ell)}; \\ \widehat{\mathbf{B}}\mathbf{r}_{\mathbf{x}}^{(B)} + \gamma \cdot \mathbf{G}\mathbf{t} + \mathbf{r}_{\mathbf{e}}^{(B)}; \{\mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^*\mathbf{r}_{\widehat{\mathbf{w}}}^{(j+1)} - \mathbf{G}^*\mathbf{r}_{\mathbf{z}}^{(j+1)}\}_{j=1}^{\ell-1}; \rho_1) \\ C'_2 := \mathsf{COM}(\{\pi_j(\mathbf{r}_{\mathbf{z}}^{(j)}); F_{b_j,\pi_j}(\mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}); F_{\bar{b}_j,\phi_j}(\mathbf{r}_{\widehat{\mathbf{z}}}^{(j)})\}_{j=1}^{\ell}; \tau(\mathbf{r}_{\mathbf{x}}^{(A)}); \eta(\mathbf{r}_{\mathbf{x}}^{(B)}); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)}); \rho_2) \\ C'_3 := \mathsf{COM}(\{\pi_j(\mathbf{z}'_j{}^* + \mathbf{r}_{\mathbf{z}}^{(j)}); F_{b_j,\pi_j}(\widehat{\mathbf{z}'_j}^* + \mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}); F_{\bar{b}_j,\phi_j}(\widehat{\mathbf{w}'_j}^* + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)})\}_{j=1}^{\ell}; \\ \tau(\mathbf{r}_{\mathbf{x}}^{(A)} + \mathbf{x}'^*); \eta(\mathbf{r}_{\mathbf{x}}^{(B)} + \mathbf{x}'^*); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)} + \mathbf{e}'); \rho_3); \end{cases}$$

6. Upon receving a challenge $\mathsf{CH} \in \{1,2,3\}$ from $\widehat{\mathcal{V}}$, $\mathcal{S}$ responds as follows:

- If $\mathsf{CH} = 1$: Output $\perp$ and abort.
- If $\mathsf{CH} = 2$: Send reponse $\mathsf{RSP} := (\{c_j; \hat{\pi}_j; \hat{\phi}_j; \mathbf{z}'^{\,*}_j + \mathbf{r}_{\mathbf{z}}^{(j)}; \widehat{\mathbf{z}}'^{*}_j + \mathbf{r}'^{(j)}_{\widehat{\mathbf{z}}}; \widehat{\mathbf{w}}'^{*}_j + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \hat{\tau}; \hat{\eta}; \hat{\zeta}; \mathbf{x}'^{*} + \mathbf{r}_{\mathbf{x}}^{(A)}; \mathbf{x}'^{*} + \mathbf{r}_{\mathbf{x}}^{(B)}; \mathbf{e}' + \mathbf{r}_{\mathbf{e}}^{(B)}; \rho_1; \rho_3)$.
- If $\mathsf{CH} = 3$: Send response $\mathsf{RSP} := (\{b_j; \tilde{\pi}_j; \tilde{\phi}_j; \mathbf{r}_{\mathbf{z}}^{(j)}; \mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \tilde{\tau}; \mathbf{r}_{\mathbf{x}}^{(A)}; \tilde{\eta}; \mathbf{r}_{\mathbf{x}}^{(B)}; \tilde{\zeta}; \mathbf{r}_{\mathbf{e}}^{(B)} \rho_1; \rho_2)$.

Case $\overline{\mathsf{CH}} = 2$: $\mathcal{S}$ follows below steps:

1. Sample $x'^{*} \xleftarrow{\$} \mathcal{B}_{2m}^{m}$, $\mathbf{e}' \xleftarrow{\$} \mathbb{Z}_q^n$.
2. For $j \in [\ell]$, sample $\{i'_j, b'_j\} \xleftarrow{\$} \{0,1\}$; $\pi_j, \phi_j \xleftarrow{\$} \mathcal{S}_m$; $\tau, \eta \xleftarrow{\$} \mathcal{S}_{2m}$; $\zeta \xleftarrow{\$} \mathcal{S}_n$ and $\{\mathbf{z}'^{*}_j, \mathbf{w}'^{*}_j\} \xleftarrow{\$} \mathcal{B}_m^{nk}$.
3. Sample random masking vectors $\mathbf{r}_{\mathbf{z}}^{(1)}, \cdots, \mathbf{r}_{\mathbf{z}}^{(\ell)} \xleftarrow{\$} \mathbb{Z}_q^m$; $\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{z}}}^{(\ell)}$; $\mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}, \cdots, \mathbf{r}_{\widehat{\mathbf{w}}}^{(\ell)}; \mathbf{r}_{\mathbf{x}}^{(A)}; \mathbf{r}_{\mathbf{x}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^{2m}; \mathbf{r}_{\mathbf{e}}^{(B)} \xleftarrow{\$} \mathbb{Z}_q^n$.
4. Let $\widehat{\mathbf{z}}'^{*}_j := \mathsf{ext}(i'_j, \mathbf{z}'^{*}_j)$, $\widehat{\mathbf{w}}'^{*}_j := \mathsf{ext}(\overline{i'_j}, \mathbf{w}'^{*}_j)$.
5. Send $\widehat{\mathcal{V}}$ a commiment $\mathsf{CMT}^{(\mathsf{sim})} := (C'_1, C'_2, C'_3)$ computed as below:

$$\begin{cases} C'_1 := \mathsf{COM}(\{b_j, \pi_j, \phi_j\}_{j=1}^{\ell}; \tau; \eta; \zeta; \mathbf{A}^{*}\mathbf{r}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^{*}\mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}; \widehat{\mathbf{A}}\mathbf{r}_{\mathbf{x}}^{(A)} - \mathbf{G}^{*}\mathbf{r}_{\mathbf{z}}^{(\ell)}; \\ \widehat{\mathbf{B}}\mathbf{r}_{\mathbf{x}}^{(B)} + \gamma \cdot \mathbf{G}\mathbf{t} + \mathbf{r}_{\mathbf{e}}^{(B)}; \{\mathbf{A}^{*}\mathbf{r}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^{*}\mathbf{r}_{\widehat{\mathbf{w}}}^{(j+1)} - \mathbf{G}^{*}\mathbf{r}_{\mathbf{z}}^{(j+1)}\}_{j=1}^{\ell-1}; \rho_1); \\ C'_2 := \mathsf{COM}(\{\pi_j(\mathbf{r}_{\mathbf{z}}^{(j)}); F_{b_j,\pi_j}(\mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}); F_{\bar{b_j},\phi_j}(\mathbf{r}_{\widehat{\mathbf{z}}}^{(j)})\}_{j=1}^{\ell}; \tau(\mathbf{r}_{\mathbf{x}}^{(A)}); \eta(\mathbf{r}_{\mathbf{x}}^{(B)}); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)}); \rho_2); \\ C'_3 := \mathsf{COM}(\{\pi_j(\mathbf{z}'^{\,*}_j + \mathbf{r}_{\mathbf{z}}^{(j)}); F_{b_j,\pi_j}(\widehat{\mathbf{z}'_j}^{*} + \mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}); F_{\bar{b_j},\phi_j}(\widehat{\mathbf{w}'_j}^{*} + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)})\}_{j=1}^{\ell}; \\ \tau(\mathbf{r}_{\mathbf{x}}^{(A)} + \mathbf{x}'^{*}); \eta(\mathbf{r}_{\mathbf{x}}^{(B)} + \mathbf{x}'^{*}); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)} + \mathbf{e}'); \rho_3). \end{cases}$$

6. Upon receving a challenge $\mathsf{CH} \in \{1,2,3\}$ from $\widehat{\mathcal{V}}$, $\mathcal{S}$ responds as follows:
   - If $\mathsf{CH} = 1$: Send response $\mathsf{RSP}^{(\mathsf{sim})} := (\{i'_j \oplus b'_j; \pi_j(\mathbf{z}'^{\,*}_j); \phi(\mathbf{w}'^{\,*}_j); \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)}); F_{b_j,\pi_j}(\mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}); F_{\overline{b_j},\phi_j}(\mathbf{r}_{\widehat{\mathbf{w}}}^{(j)})\}_{j=1}^{\ell}; \tau(\mathbf{x}'^{*}); \eta(\mathbf{x}'^{*}); \zeta(\mathbf{e}'); \tau(\mathbf{r}_{\mathbf{x}}^{(A)}); \eta(\mathbf{r}_{\mathbf{x}}^{(B)}); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)}); \rho_2; \rho_3)$
   - If $\mathsf{CH} = 2$: Output $\perp$ and abort.
   - If $\mathsf{CH} = 3$: Send response $\mathsf{RSP}^{(\mathsf{sim})}$ as in the case $\{\overline{\mathsf{CH}} = 1, \mathsf{CH} = 3\}$.

Case $\overline{\mathsf{CH}} = 3$: $\mathcal{S}$ follows steps $(1,2,3,4)$ as in the case $\overline{\mathsf{CH}} = 2$ and then follows:

5. Send $\widehat{\mathcal{V}}$ a commiment $\mathsf{CMT}^{(\mathsf{sim})} := (C'_1, C'_2, C'_3)$ computed as below:

$$\begin{cases} C'_1 := \mathsf{COM}(\{b'_j, \hat{\pi}_j, \hat{\phi}_j\}_{j=1}^{\ell}; \hat{\tau}; A^{*}(\widehat{\mathbf{z}}'_1 + \mathbf{r}_{\widehat{\mathbf{z}}}^{(1)}) + A^{*}(\widehat{\mathbf{w}}'_1 + \mathbf{r}_{\widehat{\mathbf{w}}}^{(1)}) - \mathbf{G}\mathbf{p}; \\ \{A^{*}(\widehat{\mathbf{z}}'_{j+1} + \mathbf{r}_{\widehat{\mathbf{z}}}^{(j+1)}) + A^{*}(\widehat{\mathbf{w}}'_{j+1} + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j+1)}) - \mathbf{G}^{*}(\mathbf{z}'^{\,*}_j + \mathbf{r}_{\mathbf{z}}^{(j+1)})\}_{j=1}^{\ell-1}; \\ \widehat{\mathbf{A}}(\mathbf{x}'^{*} + \mathbf{r}_{\mathbf{x}}^{(A)}) - \mathbf{G}^{*}(\mathbf{z}'^{\,*}_{\ell} + \mathbf{r}_{\mathbf{z}}^{(\ell)}); \widehat{\mathbf{B}}(\mathbf{x}'^{*} + \mathbf{r}_{\mathbf{x}}^{(B)}) + \mathbf{e}' + \mathbf{r}_{\mathbf{e}}^{(B)}; \rho_1); \\ C'_2 := \mathsf{COM}(\{\pi_j(\mathbf{r}_{\mathbf{z}}^{(j)}); F_{b_j,\pi_j}(\mathbf{r}_{\widehat{\mathbf{w}}}^{(j)}); F_{\bar{b_j},\phi_j}(\mathbf{r}_{\widehat{\mathbf{z}}}^{(j)})\}_{j=1}^{\ell}; \tau(\mathbf{r}_{\mathbf{x}}^{(A)}); \eta(\mathbf{r}_{\mathbf{x}}^{(B)}); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)}); \rho_2) \\ C'_3 := \mathsf{COM}(\{\pi_j(\mathbf{z}'^{\,*}_j + \mathbf{r}_{\mathbf{z}}^{(j)}); F_{b_j,\pi_j}(\widehat{\mathbf{z}'_j}^{*} + \mathbf{r}_{\widehat{\mathbf{z}}}^{(j)}); F_{\bar{b_j},\phi_j}(\widehat{\mathbf{w}'_j}^{*} + \mathbf{r}_{\widehat{\mathbf{w}}}^{(j)})\}_{j=1}^{\ell}; \\ \tau(\mathbf{r}_{\mathbf{x}}^{(A)} + \mathbf{x}'^{*}); \eta(\mathbf{r}_{\mathbf{x}}^{(B)} + \mathbf{x}'^{*}); \zeta(\mathbf{r}_{\mathbf{e}}^{(B)} + \mathbf{e}'); \rho_3). \end{cases}$$

6. Upon receving a challenge $\mathsf{CH} \in \{1,2,3\}$ from $\widehat{\mathcal{V}}$, $\mathcal{S}$ responds as follows:
   - If $\mathsf{CH} = 1$: Send response $\mathsf{RSP}^{(\mathsf{sim})}$ as in the case $\{\overline{\mathsf{CH}} = 2, \mathsf{CH} = 1\}$.
   - If $\mathsf{CH} = 2$: Send response $\mathsf{RSP}^{(\mathsf{sim})}$ as in the case $\{\overline{\mathsf{CH}} = 1, \mathsf{CH} = 2\}$.

– If $\mathsf{CH} = 3$: Output $\perp$ and abort.

In the aforementioned cases, in comparison with the commitment CMT and the challenge CH in the real interaction, the distribution of $\mathsf{CMT}^{(\mathsf{sim})}$ and the distribution of $\overline{\mathsf{CH}}$ are statistically close since COM is statistically hiding. Let ϵ denoted a negligible probability, one can check that the simulator $\mathcal{S}$ aborts with probability equal to $(\frac{1}{3} + \epsilon)$. In the case that $\mathcal{S}$ does not halt, compared to the distribution of the transcript of the prover in the real interaction, that of the one generated by $\mathcal{S}$ is also statistically close. As a result, $\mathcal{S}$ successfully simulates the honest prover with probability equal to $(\frac{2}{3} + \epsilon)$. □

A.2 Proof of Lemma 6

Proof. We now prove the soundness property of the protocol described in Figure 6 in order to show that it is a zero knowledge argument of knowledge for the relation $\mathcal{R}_{\mathsf{URS}}$. Let $\mathcal{K}$ be a knowledge extractor which takes input as 3 valid responses of CMT, where:

$$\begin{cases} \mathsf{RSP}_1 := (\{a_j; \mathbf{a}_{\mathbf{z}}^{(j)}; \mathbf{a}_{\mathbf{w}}^{(j)}; \mathbf{b}_{\mathbf{z}}^{(j)}; \mathbf{b}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{b}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \mathbf{a}_{\mathbf{x}}^{(A)}; \mathbf{b}_{\mathbf{x}}^{(A)}; \mathbf{a}_{\mathbf{x}}^{(B)}; \mathbf{b}_{\mathbf{x}}^{(B)}; \mathbf{a}_{\mathbf{e}}^{(B)}; \mathbf{b}_{\mathbf{e}}^{(B)}; \rho_2; \rho_3) \\ \mathsf{RSP}_2 := (\{c_j; \hat{\pi}_j; \hat{\phi}_j; \mathbf{c}_{\mathbf{z}}^{(j)}; \mathbf{c}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{c}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \hat{\tau}; \mathbf{c}_{\mathbf{x}}^{(A)}; \hat{\eta}; \mathbf{c}_{\mathbf{x}}^{(B)}; \hat{\zeta}; \mathbf{c}_{\mathbf{e}}^{(B)}; \rho_1; \rho_3) \\ \mathsf{RSP}_3 := (\{g_j; \tilde{\pi}_j; \tilde{\phi}_j; \mathbf{g}_{\mathbf{z}}^{(j)}; \mathbf{g}_{\widehat{\mathbf{z}}}^{(j)}; \mathbf{g}_{\widehat{\mathbf{w}}}^{(j)}\}_{j=1}^{\ell}; \tilde{\tau}; \mathbf{g}_{\mathbf{x}}^{(A)}; \tilde{\eta}; \mathbf{g}_{\mathbf{x}}^{(B)}; \tilde{\zeta}; \mathbf{g}_{\mathbf{e}}^{(B)}; \rho_1; \rho_2), \end{cases}$$

and outputs $(\mathbf{x}', \mathbf{p}', \mathsf{wit}')$ such that:

$$((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}), \mathbf{x}', \mathbf{p}', \mathsf{wit}') \in \mathcal{R}_{\mathsf{URS}}.$$

Following the verification steps in Figure 6, by comparing each item in the same commitments between different cases of CH the verification algorithm, we have:

$$\begin{cases} \mathbf{a}_{\mathbf{x}}^{(A)} \in \mathcal{B}_{2m}^{m}, \mathbf{a}_{\mathbf{x}}^{(B)} \in \mathcal{B}_{2m}^{m}, \tilde{\tau} = \hat{\tau}, \tilde{\eta} = \hat{\eta}, \tilde{\zeta} = \hat{\zeta} \\ \widehat{\mathbf{B}}\mathbf{c}_{\mathbf{x}}^{(B)} = \hat{\eta}(\mathbf{c}_{\mathbf{x}}^{(B)}) \bmod q, \\ \mathbf{b}_{\mathbf{x}}^{(A)} = \tilde{\tau}(\mathbf{g}_{\mathbf{x}}^{(A)}), \mathbf{b}_{\mathbf{x}}^{(B)} = \tilde{\eta}(\mathbf{g}_{\mathbf{x}}^{(B)}), \mathbf{b}_{\mathbf{e}}^{(B)} = \tilde{\eta}(\mathbf{g}_{\mathbf{e}}^{(B)}), \mathbf{a}_{\mathbf{x}}^{(A)} + \mathbf{b}_{\mathbf{x}}^{(A)} = \hat{\tau}(\mathbf{c}_{\mathbf{x}}^{(A)}), \\ \mathbf{a}_{\mathbf{x}}^{(B)} + \mathbf{b}_{\mathbf{x}}^{(B)} = \hat{\eta}(\mathbf{c}_{\mathbf{x}}^{(B)}), \mathbf{a}_{\mathbf{e}}^{(B)} + \mathbf{b}_{\mathbf{e}}^{(B)} = \hat{\zeta}(\mathbf{c}_{\mathbf{e}}^{(B)}) \\ \mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{w}}}^{(1)} - \mathbf{G}\mathbf{p} = \mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{z}}}^{(1)} + \mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{w}}}^{(1)} \bmod q, \\ \widehat{\mathbf{A}}\mathbf{c}_{\mathbf{x}}^{(A)} - \mathbf{G}^*\mathbf{c}_{\mathbf{z}}^{(\ell)} = \widehat{\mathbf{A}}\mathbf{g}_{\mathbf{x}}^{(A)} - \mathbf{G}^*\mathbf{g}_{\mathbf{z}}^{(\ell)} \bmod q, \\ \{\mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^*\mathbf{c}_{\widehat{\mathbf{w}}}^{(j+1)} - \mathbf{G}^*\mathbf{c}_{\mathbf{z}}^{(j+1)}\}_{j=1}^{\ell-1} = \{\mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{z}}}^{(j+1)} + \mathbf{A}^*\mathbf{g}_{\widehat{\mathbf{w}}}^{(i+1)} - \mathbf{G}^*\mathbf{g}_{\mathbf{z}}^{(j+1)}\}_{j=1}^{\ell-1} \bmod q, \end{cases}$$

and for all $j \in [\ell]$:

$$\begin{cases} c_j = g_j, \hat{\pi}_j = \tilde{\pi}_j, \hat{\phi}_j = \tilde{\phi}_j \\ \mathbf{a}_{\mathbf{z}}^{(j)} + \mathbf{b}_{\mathbf{z}}^{(j)} = \hat{\pi}_j(\mathbf{c}_{\mathbf{z}}^{(j)}), \mathbf{a}_{\widehat{\mathbf{z}}}^{(j)} + \mathbf{b}_{\widehat{\mathbf{z}}}^{(j)} = T_{c_j, \hat{\pi}_j}(\mathbf{c}_{\widehat{\mathbf{z}}}^{(j)}), \mathbf{a}_{\widehat{\mathbf{w}}}^{(j)} + \mathbf{b}_{\widehat{\mathbf{w}}}^{(j)} = T_{\overline{c}_j, \hat{\phi}_j}(\mathbf{e}_{\widehat{\mathbf{w}}}^{(j)}), \\ \mathbf{b}_{\mathbf{z}}^{(j)} = \hat{\pi}_j(\mathbf{g}_{\mathbf{z}}^{(j)}), \mathbf{b}_{\widehat{\mathbf{z}}}^{(j)} = T_{g_j, \hat{\pi}_j}(\mathbf{g}_{\widehat{\mathbf{z}}}^{(j)}), \mathbf{b}_{\widehat{\mathbf{w}}}^{(j)} = T_{\overline{g}_j, \hat{\phi}_j}(\mathbf{g}_{\widehat{\mathbf{w}}}^{(j)}). \end{cases}$$

Let $\mathbf{x}^* = \hat{\tau}^{-1}(\mathbf{a}_{\mathbf{x}}^{(A)})$, $\forall j \in [\ell]$, we have: $i_j = a_j \oplus c_j; \mathbf{z}_j^* = \widehat{\pi}_j^{-1}(\mathbf{a}_{\mathbf{z}}^{(j)}); \mathbf{w}_j^* = \widehat{\phi}_j^{-1}(\mathbf{a}_{\mathbf{w}}^{(j)}); \widehat{\mathbf{z}}_j = \mathbf{c}_{\widehat{\mathbf{z}}}^{(j)} - \mathbf{p}_{\widehat{\mathbf{z}}}^{(j)}; \widehat{\mathbf{w}}_j = \mathbf{c}_{\widehat{\mathbf{w}}}^{(j)} - \mathbf{p}_{\widehat{\mathbf{w}}}^{(j)}$. We observe that:

$$\begin{cases} \mathbf{x}^* \in \mathcal{B}_{2m}^m; \{\mathbf{z}_j^*, \mathbf{w}_j^*\}_{j=1}^{\ell} \in \mathcal{B}_m^{nk}; \\ \widehat{\mathbf{z}}_j = \mathsf{ext}(i_j, \mathbf{z}_j^*); \widehat{\mathbf{w}}_j = \mathsf{ext}(\bar{i}_j, \mathbf{w}_j^*). \end{cases}$$

The knowledge extractor $\mathcal{K}$ now computes:

$$\begin{cases} \mathbf{G}\mathbf{z}_j^* = \mathbf{A}^* \cdot \mathsf{ext}(i_{j+1}, \mathbf{z}_{j+1}^*) + \mathbf{A}^* \cdot \mathsf{ext}(\overline{i_{j+1}}, \mathbf{w}_{j+1}^*) + \mathbf{B} \cdot \widehat{\mathbf{t}} \bmod q, \forall \text{j} \in [\ell - 1] \\ \mathbf{G}\mathbf{v} = \mathbf{A}^* \cdot \mathsf{ext}(i_1, \mathbf{z}_1^*) + \mathbf{A}^* \cdot \mathsf{ext}(\overline{i_1}, \mathbf{w}_1^*) + \mathbf{B} \cdot \widehat{\mathbf{t}} \bmod q \end{cases}$$

Next, $\mathcal{K}$ performs:

- Reduce $\mathbf{x}^* \in \mathcal{B}_{2m}^m$ to $\mathbf{x}' \in \{0,1\}^m$ by dropping the last m coordinates.
- Reduce $\{\mathbf{z}_j^*, \mathbf{w}_j^*\}_{j=1}^{\ell} \in \mathcal{B}_m^{nk}$ to $\{\mathbf{z}_j'^*, \mathbf{w}_j'^*\}_{j=1}^{\ell} \in \{0,1\}^{nk}$ by dropping the last nk coordinates.

Now we have:

$$\begin{cases} \mathbf{G}\mathbf{z}_j' = \mathbf{A} \cdot \mathsf{ext}(i_{j+1}, \mathbf{z}_{j+1}') + \mathbf{A} \cdot \mathsf{ext}(\overline{i_{j+1}}, \mathbf{w}_{j+1}') + \mathbf{B} \cdot \widehat{\mathbf{t}} \bmod q, \forall \text{j} \in [\ell - 1] \\ \mathbf{G}\mathbf{v} = \mathbf{A} \cdot \mathsf{ext}(i_1, \mathbf{z}_1') + \mathbf{A} \cdot \mathsf{ext}(\overline{i_1}, \mathbf{w}_1') + \mathbf{B} \cdot \widehat{\mathbf{t}} \bmod q \end{cases}$$

These above equations hold when:

$$\begin{cases} \mathbf{z}_0' = \mathbf{v} \\ \mathbf{z}_j' := \overline{i_{j+1}} \cdot h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{z}_{j+1}', \mathbf{w}_{j+1}') + i_{j+1} \cdot h_{\mathbf{A},\mathbf{B},\mathbf{t}}(\mathbf{w}_{j+1}', \mathbf{z}_{j+1}'), \forall j \in [\ell - 1] \end{cases}$$

Let $\mathbf{p}' = \mathbf{z}_\ell'$, $\mathsf{wit}' = ((i_j)_{j=1}^{\ell}, (\mathbf{w}_j')_{j=1}^{\ell})$, we have $\mathsf{PACC.Verify}_{pp}(\mathbf{B}, \mathbf{v}, \mathbf{t}, \mathbf{p}', \mathsf{wit}') = 1$. Hence, $\mathcal{K}$ can successfully outputs a tuple $(\mathbf{x}', \mathbf{p}', \mathsf{wit}')$ such that:

$$((\mathbf{A}, \mathbf{B}, \mathbf{v}, \mathbf{t}), \mathbf{x}', \mathbf{p}', \mathsf{wit}') \in \mathcal{R}_{\mathsf{URS}}.$$

This completes the proof. □

B Proof of The Lattice-based URS Scheme

B.1 Proof of Theorem 3

The following lemma will be helpful for the proof of the unforgeability of our proposed URS.

Lemma 7 ([Lyu08, Lemma 8]). *For any matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a uniformly random* $\mathbf{x} \xleftarrow{\$} \{0,1\}^m$*, the probability that there exists another* $\mathbf{x}' \in \{0,1\} \setminus \{\mathbf{x}\}$ *such that* $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' \bmod q$ *is at least* $1 - 2^{n \cdot \log q - m}$.

Proof. Suppose that there is an adversary $\mathcal{A}$ that can break the unforgeability of the URS given in Section 4 with non-negligible advantage ϵ. We now construct an algorithm $\mathcal{B}$ with a non-negligible advantage that either breaks the soundness of the ZKAoK protocol given in Lemma 6, breaks the security of the accumulator presented in Theorem 1, or solves the SIS instance given by $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.

To simulate the Unforgeability game for $\mathcal{A}$, $\mathcal{B}$ sets the public parameter $urs.pp$ to the matrix $\mathbf{A}$. To answer $\mathcal{A}$'s corruption query on the i-th member to $\mathcal{O}_{sk}$, $\mathcal{B}$ faithfully returns pairs of (sk, pk) which are distributed exactly as the real scheme. Namely, $\mathcal{B}$ chooses a secret key $\mathbf{x} \xleftarrow{\$} \{0,1\}^m$ and then returns the public key computed as $\mathbf{p} := \mathsf{bin}(\mathbf{A}\mathbf{x} \bmod q)$. Here, $\mathcal{B}$ models H_{FS} as a random oracle. To answer each of q_{FS} hash queries to H_{FS} from $\mathcal{A}$, $\mathcal{B}$ returns a value chosen uniformly from $\{1,2,3\}^\kappa$. Notice that each hash query to H_{FS} consists of a message μ, κ commitments CMT_j's, matrix $\mathbf{A}$, vector $\mathbf{v}$,, matrix $\mathbf{B}$, ring R and unique tag $\mathbf{t}$. In order to answer a query (μ, R) to $\mathcal{O}_{Sign}$ from $\mathcal{A}$, $\mathcal{B}$ returns a valid triple (μ, R, sig).

Assuming that after querying, $\mathcal{A}$ successfully forges a valid triple $(\mu^{(\mathsf{for})}, R^{(\mathsf{for})}, \mathsf{sig}^{(\mathsf{for})})$ and wins the game defined in Figure 4 with a non-negligible probability, in which

$$\mathsf{sig}^{(\mathsf{for})} := (\{\mathsf{CMT}_j^{(\mathsf{for})}\}_{j=1}^{\kappa}, \{\mathsf{CH}_j^{(\mathsf{for})}\}_{j=1}^{\kappa}, \{\mathsf{RSP}_j^{(\mathsf{for})}\}_{j=1}^{\kappa}, \mathbf{t}^{(\mathsf{for})}).$$

Note that, with probability $\epsilon - 3^{-\kappa}$, there is a hash query made by the adversary $\mathcal{A}$ at the **Query** phase, which is indexed by some $t^* \in \{1, \cdots, q_{FS}\}$ such that $(\mu^{(\mathsf{for})}, \mathsf{CMT}_j^{(\mathsf{for})}, \mathbf{A}, \mathbf{v}^{(\mathsf{for})}, R^{(\mathsf{for})}, \mathbf{B}^{(\mathsf{for})}, \mathbf{t}^{(\mathsf{for})})_{j\in[\kappa]}$ has been the t^*-th hash query of $\mathcal{A}$, where $\mathbf{B}^{(\mathsf{for})} := H_{\mathsf{UT}}(\mu^{(\mathsf{for})}, R^{(\mathsf{for})})$, $\mathbf{t}^{(\mathsf{for})} := \mathsf{bin}(\mathsf{F}_{\mathbf{x}^{(\mathsf{for})}}(\mathbf{B}^{(\mathsf{for})})(\bmod q))$, $\mathbf{v}^{(\mathsf{for})} := \mathsf{PACC.Acc}_{\mathbf{A}}(\mathbf{B}^{(\mathsf{for})}, \mathbf{t}^{(\mathsf{for})}, R^{(\mathsf{for})})$ for some $\mathbf{x}^{(\mathsf{for})} \in \{0,1\}^m$.

To extract the solution to the given SIS instance, $\mathcal{B}$ needs to re-run up to $32 \cdot q_{FS}/(\epsilon - 3^{-\kappa})$ executions of $\mathcal{A}$ with the same random tape and the same input as in the first $\mathcal{A}$'s execution. In each new execution, the first $t^* - 1$ hash queries to H_{FS} get the same responses as in the first execution. However, from the t^*-th hash query to the q_{FS}-th one, the responses are freshly sampled at random from $\{1,2,3\}^\kappa$. Using the forking lemma [BPVY01], at the t^*-th hash query where a forking occurs, with the probability at least $1/2$, the algorithm $\mathcal{B}$ can obtain a 3-fork with distinct reponses $\mathsf{CH}^{(t^*,1)}, \mathsf{CH}^{(t^*,2)}, \mathsf{CH}^{(t^*,3)} \in \{1,2,3\}^\kappa$ regarding to the same hash query input $(\mu^{(\mathsf{for})}, \mathsf{CMT}_j^{(\mathsf{for})}, \mathbf{A}, \mathbf{v}^{(\mathsf{for})}, R^{(\mathsf{for})}, \mathbf{B}^{(\mathsf{for})}, \mathbf{t}^{(\mathsf{for})})_{j\in[\kappa]}$. By the result of [BPVY01], with probability $1 - (7/9)^\kappa$, there exists an index $j \in \{1, \cdots, \kappa\}$ such that $(\mathsf{CH}_j^{(t^*,1)}, \mathsf{CH}_j^{(t^*,2)}, \mathsf{CH}_j^{(t^*,3)}) = (1,2,3)$. Following the proof of soundness property in Section 4.3 (i.e., the proof of Lemma 6), with the responses $(\mathsf{RSP}^{(t^*,1)}, \mathsf{RSP}^{(t^*,2)}, \mathsf{RSP}^{(t^*,3)})$, $\mathcal{B}$ can output $(\mathbf{x}^{(\mathsf{for})}, \mathbf{p}^{(\mathsf{for})}, \mathsf{wit}^{(\mathsf{for})})$, where $\mathsf{wit}^{(\mathsf{for})} := \{(i_1^{(\mathsf{for})}, \cdots, i_\ell^{(\mathsf{for})}), (\mathbf{w}_\ell^{(\mathsf{for})}, \cdots, \mathbf{w}_1^{(\mathsf{for})})\}$ such that:

$$\begin{cases} \mathbf{A} \cdot \mathbf{x}^{(\mathsf{for})} & = \mathbf{G}\mathbf{p}^{(\mathsf{for})} \pmod q \\ \mathsf{PACC.Verify}_{\mathbf{A}}(\mathbf{B}^{(\mathsf{for})}, \mathbf{v}^{(\mathsf{for})}, \mathbf{t}^{(\mathsf{for})}, \mathbf{p}^{(\mathsf{for})}, \mathsf{wit}^{(\mathsf{for})}) & = 1 \end{cases} . \quad (3)$$

Here, $(i_1^{(\mathsf{for})}, \cdots, i_\ell^{(\mathsf{for})})$ is a binary expansion of some index $i^{(\mathsf{for})} \in \{0, \cdots, |R^{(\mathsf{for})}|-1\}$. If $\mathsf{p}^{(\mathsf{for})} \notin R^{(\mathsf{for})}$, then $\mathcal{B}$ successfully breaks the security of the accumulator with the tripple $(\mathbf{x}^{(\mathsf{for})}, \mathbf{p}^{(\mathsf{for})}, \mathsf{wit}^{(\mathsf{for})})$. Otherwise, i.e., $\mathsf{p}^{(\mathsf{for})} = \mathbf{p}_{i^{(\mathsf{for})}} \in R^{(\mathsf{for})}$, then we have the corresponding secret key $\mathbf{x}_{i^{(\mathsf{for})}} \in \{0,1\}^m$ satisfying that $\mathsf{p}^{(\mathsf{for})} = \mathsf{bin}(\mathbf{A} \cdot \mathbf{x}_{i^{(\mathsf{for})}})$. Hence,

$$\mathbf{A} \cdot \mathbf{x}_{i^{(\mathsf{for})}} = \mathbf{G}\mathbf{p}^{(\mathsf{for})} \pmod{q}. \tag{4}$$

Because no any member in $R^{(\mathsf{for})}$ (hence, the $i^{(\mathsf{for})}$-th signer) is corrupted, with probability at least 1/2 we have $\mathbf{x}^{(\mathsf{for})} \neq \mathbf{x}_{i^{(\mathsf{for})}}$. This is because if $\mathbf{x}^{(\mathsf{for})} = \mathbf{x}_{i^{(\mathsf{for})}}$, then we can apply Lemma 7 to get a new $\mathbf{x}_{i^{(\mathsf{for})}}$ such that $\mathbf{x}^{(\mathsf{for})} \neq \mathbf{x}_{i^{(\mathsf{for})}}$, following a standard *witness indistinguishability* argument (see, e.g., [Lyu08, Lyu12]). From Equations (3)-(4), we have $\mathbf{A} \cdot (\mathbf{x}_{i^{(\mathsf{for})}} - \mathbf{x}^{(\mathsf{for})}) = \mathbf{0} \pmod{q}$, which shows $(\mathbf{x}_{i^{(\mathsf{for})}} - \mathbf{x}^{(\mathsf{for})}) \in \{-1,0,1\}^m$ is a solution to the SIS instance. Therefore, algorithm $\mathcal{B}$ can solve the SIS problem with a non-negligible probability. Hence, assuming $\mathsf{SIS}^\infty_{m,n,q,\theta}$ problem is hard, our scheme is unforgeable with respect to insider corruption. □

B.2 Proof of Theorem 4

Proof. We proceed the proof with a sequence of games. Let $\Pr[\mathsf{Win}_i]$ be the probability of $\mathcal{A}$ to win Game i.

Game 0. This is the original anonymity experiment defined in Figure 3, we have

$$\mathsf{Adv}^{\mathsf{URS},\mathsf{Anon}}_{\mathcal{A},N} = |\Pr[\mathsf{Win}_0] - 1/2|.$$

Game 1. This game is similar to Game 0, except in order to produce the challenge signature sig^* with respect to the choice (μ^*, R^*, sk_{i_b}), $\mathcal{C}$ returns the simulated transcript $\Pi^{(\mathsf{sim})}$ produced as in the proof of Lemma 5 (see Appendix A.1).In this game, based on the zero-knowledge property of the proposed ZKAoK, $\mathcal{A}$ can only successfully distinguish the real proof and the simulated one with negligible probability, we have

$$|\Pr[\mathsf{Win}_1] - \Pr[\mathsf{Win}_0]| \leq \mathsf{Adv}^{\mathsf{zk}}_{(\mathcal{P},\mathcal{V})}(\mathcal{V}^*).$$

Game 2. This game is similar to Game 1, except that in the challenge phase, the challenger $\mathcal{C}$ also chooses the unique tag randomly over $\{0,1\}^{nk}$. Game 2 and Game 1 are indistinguishable under the pseudorandomness of F, which in turn relies on the hardness of the decision-LWR problem, we have $|\Pr[W_2] - \Pr[W_1]| \leq \mathsf{Adv}^{\mathsf{LWR}}(\mathcal{S})$ for some LWR solver $\mathcal{S}$.

Game 3. This game is similar to Game 2, except $\mathcal{C}$ first guesses two target indices i'_0, i'_1 in advance and randomly chooses $pk_{i'_0}, pk_{i'_1} \xleftarrow{\$} \{0,1\}^{nk}$. After that, if $\mathcal{A}$ queries for these two indices to $\mathcal{O}_{sk}, \mathcal{O}_{sign}$, $\mathcal{C}$ aborts and and restarts the game. Similarly, if $\mathcal{A}$ chooses $i_0 \neq i'_0$ or $i_1 \neq i'_1$ in the challenge phase, $\mathcal{C}$ also aborts and restarts the game. In this game, because the integers i'_0, i'_1 are independent of the view of $\mathcal{A}$ until $\mathcal{C}$ aborts. Also in Game 3, using the leftover hash lemma (i.e., Lemma 1), we have $(\mathbf{A}, pk_{i'_b})$ is statistically close to the real public keys $(\mathbf{A}, pk_{i_b} = \mathbf{A}\mathbf{x}_{i_b} \pmod q)$ for some $\mathbf{x}_{i_b} \in \{0,1\}^m$, then $\mathcal{A}$ also only can distinguish with negligible probability. Thus, the probability of $\mathcal{A}$ in Game 3 is just negligibly changed compared with that in Game 2. Therefore, we have $|\Pr[W_3] - \Pr[W_2]| \leq \mathsf{negl}(\lambda)$. Notice that, in Game 3, bit b is independent of the $\mathcal{A}$'s view, and the public key pk_{i_b} is random, the unique tag is also random, and the signature is the simulated transcript. Thus, we have $\Pr[W_3] = 1/2$. Overall, the advantage of $\mathcal{A}$ in the Anonymity experiment is

$$\mathsf{Adv}^{\mathsf{URS,Anon}}_{\mathcal{A},N} = |\Pr[\mathsf{Win}_0] - 1/2| \leq \mathsf{Adv}^{\mathsf{zk}}_{(\mathcal{P},\mathcal{V})}(\mathcal{V}^*) + \mathsf{Adv}^{\mathsf{LWR}}(\mathcal{S}) + \mathsf{negl}(\lambda).$$

This completes the proof of anonymity. □

B.3 Proof of Theorem 5

Proof. In order to prove the uniqueness for the scheme, we proceed a sequence of hybrid security games. Let Win_i be the event that the adversary $\mathcal{A}$ wins Game i. Assume that, $\mathcal{A}$ can makes up to q_{cor} corruption queries.

Game 0. This is the original uniqueness experiment presented in Figure 5. The challenger $\mathcal{C}$ interacts with the adversary $\mathcal{A}$ as below:

- Setup. $\mathcal{C}$ runs $urs.pp \leftarrow \mathsf{URS.Setup}(1^\lambda)$. Then, for eachh user $i \in [N]$ it runs $(pk_i, sk_i) \leftarrow \mathsf{URS.KeyGen}(urs.pp)$. The challenger $\mathcal{C}$ gives $S = \{pk_i\}_{i=1}^N$ to $\mathcal{A}$. It also initializes $\mathsf{Corrupt} \leftarrow \emptyset$ and $\mathsf{SIGNER}_{R,M} \leftarrow \emptyset$.
- Hash Queries to H_{FS}.Upon receiving a random oracle query with input $(\mu, \mathsf{CMT}, \mathbf{A}, \mathbf{v}, \mu, R, \mathbf{B}, \mathbf{t})$ from the adversary $\mathcal{A}$, if it has been queried before, the challenger returns the associated hash value h_{FS}, else the challenger computes $h'_{\mathsf{FS}} = H_{\mathsf{FS}}(\mu, \mathsf{CMT}, \mathbf{A}, \mathbf{v}, R, \mathbf{B}, \mathbf{t})$, then returns h'_{FS} to $\mathcal{A}$ and adds the new tuple $((\mu, \mathsf{CMT}, \mathbf{A}, \mathbf{v}, R, \mathbf{B}, \mathbf{t}), h'_{\mathsf{FS}})$ to the list L_{FS}.
- Signing Queries. Upon receiving a signing query with input (sk_i, μ, R), the challenger runs $\mathsf{URS.Sign}(urs.pp, sk_i, \mu, R)$ to obtain sig and returns to the adversary.
- Corupption Queries. Upon receiving a corruption query with input pk_i, the challenger returns sk_i to the adversary.
- Forge. The adversary $\mathcal{A}$ outputs $\zeta := |\mathsf{Corrupt}_{R^*} \cup \mathsf{SIGNER}_{R^*,\mu^*}| + 1$ different valid signatures $\mathsf{sig}_1, \ldots, \mathsf{sig}_\zeta$ on the same message μ^* in regards the same ring R^*. $\mathcal{A}$ wins if all the signatures are valid and their corresponding *unique identifiers* $\mathbf{t}_1, \ldots, \mathbf{t}_\zeta$ are pairwise distinct. Note that $\zeta \leq N + 1$.

Acording to the definition, we have

$$\mathsf{Adv}^{\mathsf{URS,Unique}}_{\mathcal{A},N} = \Pr[\mathsf{Win}_0].$$

Game 1. This game is similar to Game 0 except that for each signature sig_j, the challenger $\mathcal{C}$ takes one further action which is to check that if there exists at least one $i^*_j \in [N]$ and $\mathbf{x}_{i^*_j}$ such that for $j \in [\zeta]$,

$$\begin{cases} pk_{i^*_j} \in R^*, \\ \mathsf{F}_{\mathbf{x}_{i^*_j}}(\mathsf{H}_{\mathsf{UT}}(\mu^*, R^*)) = \mathbf{t}_j. \end{cases}$$

To this end, the challenger must also run the knowledge extractor described in the proof of Lemma 6 (given in Appendix A.2), up to ζ times, to get $\mathbf{x}_{i_j^*}$'s. Then, we have

$$|\Pr[\mathsf{Win}_1] - \Pr[\mathsf{Win}_0]| \leq \zeta \cdot \mathsf{Adv}^{\mathsf{sound}}_{(\mathcal{P},\mathcal{V})}(\mathcal{P}^*) \leq \mathsf{negl}(\lambda).$$

Game 2. This game is similar to Game 1 except that the challenger $\mathcal{C}$ uses a simulated transscript $\Pi^{(\mathsf{sim})}_{\mathsf{urs}}$ instead of a real one to reponse to a signing query. The simulated proof is formed as in the security proof of Lemma 5 (given in Appendix A.1) for the underlying ZKAoK. We have,

$$|\Pr[\mathsf{Win}_2] - \Pr[\mathsf{Win}_1]| \leq \mathsf{Adv}^{\mathsf{zk}}_{(\mathcal{P},\mathcal{V})}(\mathcal{V}^*) \leq \mathsf{negl}(\lambda).$$

Game 3. Game 3 should be the same as Game 2 with the following difference. When answering signing queries regarding any user in $R^* \setminus \mathsf{Corrupt}_{R^*}$ (i.e., uncorrupted users in R^*), the challenger chooses a unique tag uniformly at random over $\{0,1\}^{nk}$. Note that $|R^* \setminus \mathsf{Corrupt}_{R^*}| \leq N - q_{\mathsf{cor}}$, in which the right-hand side term $N - q_{\mathsf{cor}}$ is the total number of uncorrupted users in the system. The indistinguishability of Game 3 and Game 2 is guaranteed by the hardness of the decision-LWR problem. Therefore,

$$|\Pr[\mathsf{Win}_3] - \Pr[\mathsf{Win}_2]| \leq (N - q_{\mathsf{cor}}) \cdot \mathsf{Adv}^{\mathsf{LWR}}(\mathcal{S}) \leq N \cdot \mathsf{Adv}^{\mathsf{LWR}}(\mathcal{S}),$$

for some LWR solver $\mathcal{S}$. Suppose that $\mathcal{A}$ wants to produce a valid triple $(\mu^*, R^*, \mathsf{sig}^*)$ whose unique identifier differs from those of queried signatures. Then, $\mathcal{A}$ can use secret keys received when making corruption queries, and may forge some valid signatures on the same messsage μ^* and the same ring R^*, each having a new unique tag.

However, in this game, $\mathcal{A}$ only knows $|\mathsf{Corrupt}_{R^*}|$ secret keys in the ring R^*. Thus, $\mathcal{A}$ must guess the rest of them (i.e., $\zeta - |\mathsf{Corrupt}_{R^*}|$ secret keys). Therefore, the probability that $\mathcal{A}$ wins Game 3 is

$$\Pr[\mathsf{Win}_3] \leq (\zeta - |\mathsf{Corrupt}_{R^*}|) \cdot N/2^{nk} \leq \zeta \cdot N/2^{nk}.$$

Overall, we have

$$\Pr[\mathsf{Win}_0] \leq \zeta \cdot \mathsf{Adv}^{\mathsf{sound}}_{(\mathcal{P},\mathcal{V})}(\mathcal{P}^*) + N \cdot \mathsf{Adv}^{\mathsf{zk}}_{(\mathcal{P},\mathcal{V})}(\mathcal{V}^*) + \mathsf{Adv}^{\mathsf{LWR}}(\mathcal{A}') + \zeta \cdot N/2^{nk}.$$

□

B.4 Proof of Theorem 6

Proof. Recall that the non-colliding property ensures that two *unique identifiers* computed by two different signers on the same message and the same ring are the same only with a negligible probability. Since in our URS scheme, *unique identifiers* are computed by the weak pseudorandom function F presented in Section 2.6, the non-colliding property follows by the strong uniqueness property of F that is guaranteed by the hardness of decision-$\mathsf{LWR}_{n,q,p}$ problem. □