Lattice-Based Linkable Ring Signature in the Standard Model

Mingxing Hu and Zhen Liu

Shanghai Jiao Tong University, Shanghai, China
{mxhu2018,liuzhen}@sjtu.edu.cn

Abstract. Ring signatures enable a user to sign messages on behalf of an arbitrary set of users, called the ring. The anonymity of the scheme guarantees that the signature does not reveal which member of the ring signed the message. The notion of linkable ring signatures (LRS) is an extension of the concept of ring signatures such that there is a public way of determining whether two signatures have been produced by the same signer. Lattice-based LRS is an important and active research line, since lattice-based cryptography has attracted more attention due to its distinctive features especially the quantum resistant. However, *all the existing lattice-based LRS relied on random oracle heuristics*, i.e., no lattice-based LRS in the standard model has been introduced so far.

In this paper, we present a lattice-based LRS scheme based on the well-studied standard lattice assumptions (SIS and LWE) in the standard model.

Keywords: Lattice-based · Linkable ring signature · Standard model.

1 Introduction

Ring signatures, introduced by Rivest et al. [43], allow a signer to hide in a *ring* of potential signers of which the user is a member, without revealing which member actually produced the signature. However, the signer-anonymous that ring signatures provide may be too strong in some scenarios. For example, regular ring signatures cannot be used for anonymous e-voting since any double votes remain undetectable, which means no one is able to find out whether any two signatures (with two votes) are submitted by the same voter or not. Similar concerns should be aroused in cryptocurrency where a double-spent payment should be discarded. Linkable ring signatures (LRS) [35] provide the remedy to this problem by allowing the public to detect any signer who has produced two or more signatures (i.e., votes, payments). Thereafter, LRS has been studied extensively [1,2,3,9,22,23,29,44,45,49,50,51] especially in recent years, driven by the rapid development of cryptocurrencies.

Another important line of research is constructing LRS schemes from lattices [11,12,30,46,52,34,48,7,47,15], since lattice-based cryptography has attracted more attention due to its distinctive features such as efficient, simple, highly parallelizable and the potentially quantum resistant. However, these works have so far required the random oracle (ROM) model [16] (or similar heuristics) for their security analysis. Katz (Sect. 6.2.1 of [28]) mentioned that existing some negative results concerning the relying on ROM. Canetti et al. [18] and Dodis et al. [20] shown that a proof in ROM can only serve as a heuristic argument and, admittedly using quite contrived constructions, has been shown to possibly lead to insecure schemes when the ROM are implemented in the practical scenarios. Furthermore, Leurent and Nguyen [33] presented the attacks extracting the secret keys on several hash-then-sign type signature schemes (includes the lattice based signature [25]) and identity-based encryption schemes if the underlying hash functions are modeled as random oracle. Quantum Random Oracle Model (QROM) is a generalized notion of ROM [8]. Though a proof of security in the QROM is stronger than one in the ROM, it does not means the security in the QROM implies standard-model security [21]. Furthermore, Grilo et al. [24] shown that the proofs in QROM lacking of conceptual simplicity and tightness.

1.1 Our Results

To address the above concerns, we present a lattice-based LRS scheme provably secure from the well-studied standard lattice assumptions (SIS and LWE) in the standard model. It is worth to mention that we employ the strongest security model that is strong unforgeability w.r.t. insider corruption (An important realistic attack presented by Bender et al. [10]). In other words, our construction provides strong confidence on security in threefold: provably secure without relying on any random oracle heuristics, and instantiated under standard lattice assumptions make our work being quantum-resistant, and satisfies the strongest security model that capture the realistic attacks i.e., strongly unforgeable w.r.t. insider corruption that make our system more applicable in practical scenarios.

Moreover, we present two new lattice basis extending algorithms that may be of independent interest. The algorithms are the key ingredients in our construction, which break the obstacle in building the 'key image' of LRS without the help of cryptographic hash functions that are modeled as random oracles.

2 Definitions

In this section, we review the definitions of linkable ring signatures: syntax, correctness, unforgeability, anonymity, linkability, and non-slanderability.

Definition 1 (Linkable Ring Signature). *A linkable ring signature LRS consists of the following algorithms:*

- $\mathsf{Setup}(1^n) \rightarrow \mathsf{PP}$. *This is a probabilistic algorithm. On input the security parameter n, outputs the public parameter* PP.

  The public parameters PP are common parameters used by all ring members in the system, for example, the message space $\mathcal{M}$, the modulo, etc. In the following, PP is implicit input parameter to every algorithm.

- $\mathsf{KeyGen}() \rightarrow (\mathsf{vk},\mathsf{sk})$. *This is a probabilistic algorithm. The algorithm outputs a verification key* vk *and a signing key* sk.

  Any ring member can run this algorithm to generate a pair of verification key and signing key.

- $\mathsf{Sign}(\mathsf{sk}, \mu, \mathsf{R}) \rightarrow \Sigma$. *This is a probabilistic algorithm. On input a signing key* sk, *a message* $\mu \in \mathcal{M}$, *and a ring of verification keys* $\mathsf{R} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$[1]. *Assume that (1) the input signing key* sk *and the corresponding verification key* vk *is a valid key pair output by* KeyGen *and* $\mathsf{vk} \in \mathsf{R}$, *(2) the ring size* $|\mathsf{R}| \geq 2$, *(3) each verification key in ring* R *is distinct. This algorithm outputs a signature* Σ.

- $\mathsf{Ver}(\mathsf{R}, \mu, \Sigma) \rightarrow 1/0$. *This is a deterministic algorithm. On input a ring of verification keys* $\mathsf{R} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$, *a message* $\mu \in \mathcal{M}$, *and a signature* Σ, *outputs* 1 *if the signature is valid, or* 0 *if the signature is invalid.*

- $\mathsf{Link}(\mathsf{R}_0, \mu_0, \Sigma_0, \mathsf{R}_1, \mu_1, \Sigma_1) \rightarrow 1/0$. *This is a deterministic algorithm. On input two valid signature tuples* $(\mathsf{R}_0, \mu_0, \Sigma_0), (\mathsf{R}_1, \mu_1, \Sigma_1)$, *the algorithm outputs* 1 *if the two signatures* linked, *or* 0 *if* unlinked.

Remark: Note that it is open on whether the Sign algorithm is probabilistic or deterministic, which may depend on the concrete constructions.

Correctness. *A LRS scheme is correct, if for all* $n \in \mathbb{N}$, *any* $N = \mathrm{poly}(n)$, *any* $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n)$ *as implicit input parameter to every algorithm, any* N *pairs* $(\mathsf{vk}^{(1)}, \mathsf{sk}^{(1)}), \ldots, (\mathsf{vk}^{(N)}, \mathsf{sk}^{(N)}) \leftarrow \mathsf{KeyGen}()$, *let* $\mathsf{R} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$, *it holds that*

- *For any messages* $\mu \in \mathcal{M}$, *and any* $s \in [N]$, *it holds that*

$$\Pr\left[\mathsf{Ver}(\mathsf{R}, \mu, \mathsf{Sign}(\mathsf{sk}^{(s)}, \mu, \mathsf{R})) = 1\right] = 1 - \mathrm{negl}(n)$$

- *For any messages* $\mu_0, \mu_1 \in \mathcal{M}$, *any* $N_0, N_1 = poly(n)$, *any ring of well-formed verification keys* $\mathsf{R}_0, \mathsf{R}_1$ *with ring size* $|\mathsf{R}_0| = N_0, |\mathsf{R}_1| = N_1$ *respectively, and any* $\mathsf{vk}^{(s_0)} \in \mathsf{R}_0, \mathsf{vk}^{(s_1)} \in \mathsf{R}_1$ *for any* $s_0 \in [N_0], s_1 \in [N_1]$, *let* $\Sigma_0 \leftarrow \mathsf{Sign}(\mathsf{sk}^{(s_0)}, \mu_0, \mathsf{R}_0)$, $\Sigma_1 \leftarrow \mathsf{Sign}(\mathsf{sk}^{(s_1)}, \mu_1, \mathsf{R}_1)$. *It holds that*

$$\Pr\left[\mathsf{Link}(\mathsf{R}_0, \mu_0, \Sigma_0, \mathsf{R}_1, \mu_1, \Sigma_1) = 1\right] = 1 \quad \text{if } \mathsf{sk}^{(s_0)} = \mathsf{sk}^{(s_1)},$$
$$\Pr\left[\mathsf{Link}(\mathsf{R}_0, \mu_0, \Sigma_0, \mathsf{R}_1, \mu_1, \Sigma_1) = 0\right] \geq 1 - \mathsf{negl}(n) \quad \text{if } \mathsf{sk}^{(s_0)} \neq \mathsf{sk}^{(s_1)}$$

The above probability is taken over the random coins used by Setup, KeyGen, *and* Sign.

Strong Unforgeability. *A LRS scheme is strongly unforgeable w.r.t. insider corruption* (sUnfInsCor), *if for any PPT forger* $\mathcal{A}$, *it holds that* $\mathcal{A}$ *has at most negligible advantage in the following experiment with a challenger* $\mathcal{C}$.

[1] Below we regard the verification key ring as an ordered set, namely, it consists of a set of verification keys, and when it is used in Sign and Ver algorithms, the verification keys are ordered and each one has an index.

- **Setup.** *$\mathcal{C}$ generates* $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$ *and* $(\mathsf{vk}^{(i)}, \mathsf{sk}^{(i)}) \leftarrow \mathsf{KeyGen}(\gamma_{\mathsf{kg}}^{(i)})$ *for all $i \in [N]$, where $N = \text{poly}(n)$ and $(\gamma_{\mathsf{st}}, \gamma_{\mathsf{kg}}^{(i)})$ are the randomnesses used in* Setup *and* KeyGen*, respectively. $\mathcal{C}$ sets* $\mathsf{S} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$ *and initializes two empty sets* L *and* C*. Finally, $\mathcal{C}$ sends* $(\mathsf{PP}, \mathsf{S}, \gamma_{\mathsf{st}})$ *to $\mathcal{A}$.*

  Note that we give to $\mathcal{A}$ the randomness γ_{st} that used for the Setup algorithm, which implies the algorithm is public, does not rely on a trusted setup that may incur concerns on the existing of trapdoors hidden in the output parameters.

- **Probing Phase.** *$\mathcal{A}$ can adaptively query the following oracles:*

  - *Signing oracle* $\mathsf{OSign}(\cdot, \cdot, \cdot)$*:*
    On input a message $\mu \in \mathcal{M}$, a ring of verification keys R *and an index $s \in [N]$ such that* $\mathsf{vk}^{(s)} \in \mathsf{R} \bigcap \mathsf{S}$*, this oracle returns* $\Sigma \leftarrow \mathsf{Sign}(\mathsf{sk}^{(s)}, \mu, \mathsf{R})$ *and adds the tuple* $(\mu, \mathsf{R}, \Sigma)$ *to* L*.*

  - *Corrupting oracle* $\mathsf{OCorrupt}(\cdot)$*:*
    On input an index $s \in [N]$ such that $\mathsf{vk}^{(s)} \in \mathsf{S}$*, this oracle returns* $\gamma_{\mathsf{kg}}^{(s)}$ *and adds* $\mathsf{vk}^{(s)}$ *to* C*.*

- **Forge.** *$\mathcal{A}$ outputs a forgery* $(\mu^*, \mathsf{R}^*, \Sigma^*)$ *and succeeds if (1)* $\mathsf{Ver}(\mu^*, \mathsf{R}^*, \Sigma^*) = 1$*, (2)* $\mathsf{R}^* \subseteq \mathsf{S} \setminus \mathsf{C}$*, and (3)* $(\mu^*, \mathsf{R}^*, \Sigma^*) \notin \mathsf{L}$*.*

Anonymity. *A LRS scheme is signer-anonymous, if for any PPT adversary $\mathcal{A}$, it holds that $\mathcal{A}$ has at most negligible advantage in the following experiment with a challenger $\mathcal{C}$.*

- **Setup**. *$\mathcal{C}$ generates* $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$ *and* $(\mathsf{vk}^{(i)}, \mathsf{sk}^{(i)}) \leftarrow \mathsf{KeyGen}()$ *for all $i \in [N]$, where $N = \text{poly}(n)$ and γ_{st} is the randomness used in* Setup*. $\mathcal{C}$ sets* $\mathsf{S} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$*. Finally, $\mathcal{C}$ sends* $(\mathsf{PP}, \mathsf{S}, \gamma_{\mathsf{st}})$ *to $\mathcal{A}$.*

- **Probing Phase 1.** *$\mathcal{A}$ adaptively queries the signing oracle* $\mathsf{OSign}(\cdot, \cdot, \cdot)$*: On input a ring of verification keys* R*, a message $\mu \in \mathcal{M}$, and an index $s \in [N]$, where requires that* $\mathsf{vk}^{(s)} \in \mathsf{R} \bigcap \mathsf{S}$*, this oracle returns* $\Sigma \leftarrow \mathsf{Sign}(\mathsf{sk}^{(s)}, \mu, \mathsf{R})$*.*

- **Challenge**. *$\mathcal{A}$ provides a challenge* $(\mathsf{R}^*, \mu^*, s_0^*, s_1^*)$ *to the challenger such that $s_0^*, s_1^* \in [N]$, $s_0^* \neq s_1^*$,* $\mathsf{vk}^{(s_0^*)}, \mathsf{vk}^{(s_1^*)} \in \mathsf{S} \bigcap \mathsf{R}^*$ *and none of* $\mathsf{OSign}(\cdot, \cdot, s_0^*)$, $\mathsf{OSign}(\cdot, \cdot, s_1^*)$ *was queried. $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and $\mathcal{A}$ is given the signature* $\Sigma^* \leftarrow \mathsf{Sign}(\mathsf{sk}^{(s_b^*)}, \mu^*, \mathsf{R}^*)$*.*

- **Probing Phase 2**. *Same as the* **Probing Phase 1***, but with the restriction that none of* $\mathsf{OSign}(\cdot, \cdot, \pi_0^*)$ *and* $\mathsf{OSign}(\cdot, \cdot, s_1^*)$ *is queried.*

- **Guess**. *$\mathcal{A}$ outputs a guess b'. If $b' = b$, $\mathcal{C}$ outputs 1, otherwise 0.*

Linkability. *A LRS scheme is signer-linkable, if for any PPT adversary $\mathcal{A}$, it holds that $\mathcal{A}$ has at most negligible advantage in the following experiment with a challenger $\mathcal{C}$.*

- **Setup**. *$\mathcal{C}$ generates* $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$*, where* γ_{st} *is the randomness used in* Setup*. Finally,* $\mathcal{C}$ *sends* $(\mathsf{PP}, \gamma_{\mathsf{st}})$ to $\mathcal{A}$*.*
- **Output Phase**. *$\mathcal{A}$ outputs l $(l \geq 2)$ (ring of well-formed verification keys, messages, signature) tuples* $(\mathsf{R}^*_i, \mu^*_i, \Sigma^*_i)$ *where* $i \in [l]$*.*

$\mathcal{A}$ succeeds if (1) $\mathsf{Ver}(\mathsf{R}^*_i, \mu^*_i, \Sigma^*_i) = 1$ *for* $i \in [l]$*, (2)* $\mathsf{Link}(\mathsf{R}^*_i, \mu^*_i, \Sigma^*_i, \mathsf{R}^*_j, \mu^*_j, \Sigma^*_j) = 0$ *for any* $i, j \in [l]$ *s.t.* $i \neq j$*, and (3)* $| \cup_{i=1} \mathsf{R}^*_i| < l$*.*

Non-Slanderability. *A LRS scheme is signer-non-slanderable, if for any PPT adversary $\mathcal{A}$, it holds that $\mathcal{A}$ has at most negligible advantage in the following experiment with a challenger $\mathcal{C}$.*

- **Setup.** *As same as the setup phase of* **Strong Unforgeability**.
- **Probing Phase.** *As same as the probing phase of* **Strong Unforgeability**.
- **Output Phase**. *$\mathcal{A}$ outputs two (ring of verification keys, message, signature) tuples* $(\mathsf{R}^*, \mu^*, \Sigma^*)$ *and* $(\hat{\mathsf{R}}, \hat{\mu}, \hat{\Sigma})$*.*

Let L *be the list that stores the query-answer tuples for* $\mathsf{OSign}(\cdot, \cdot, \cdot)$*. $\mathcal{A}$ succeeds if (1)* $\mathsf{Ver}(\mathsf{R}^*, \mu^*, \Sigma^*) = 1$*, (2)* $(\hat{\mathsf{R}}, \hat{\mu}, \hat{\Sigma}) \in \mathsf{L}$*, (3)* $(\mathsf{R}^*, \mu^*, \Sigma^*) \notin \mathsf{L}$*, (4)* $\mathsf{R}^* \subseteq \mathsf{S} \setminus \mathsf{C}$*, (5)* $\mathsf{Link}(\mathsf{R}^*, \mu^*, \Sigma^*, \hat{\mathsf{R}}, \hat{\mu}, \hat{\Sigma}) = 1$*.*

3 Preliminaries

In this section, we first review the definition of strongly unforgeable one-time signature in Sect. 3.1, key-homomorphic evaluation algorithm in Sect. 3.2, non-interactive witness-indistinguishable proof systems in Sect. 3.3, and some lattice-based backgrounds.

Notation. We write $[l]$ for a positive integer l to denote the set $\{1, \ldots, l\}$. We denote vectors as lower-case bold letters (e.g. $\mathbf{x}$), and matrices by upper-case bold letters (e.g. $\mathbf{A}$). We say that a function in n is *negligible*, written $\mathrm{negl}(n)$, if it vanishes faster than the inverse of any polynomial in n. We say that a probability $p(n)$ is *overwhelming* if $1 - p(n)$ is negligible. We denote the horizontal concatenation of two matrices $\mathbf{A}$ and $\mathbf{B}$ as $\mathbf{A}|\mathbf{B}$. We denote the vertical concatenation of two matrices $\mathbf{A}$ and $\mathbf{B}$ as $\mathbf{A}; \mathbf{B}$. We denote $\{\mathbf{A}^{(i)}\}_{i \in [l]}$ or $\{\mathbf{B}_j\}_{j \in [l]}$ as the set that consists of l matrices.

3.1 Strongly Unforgeable One-Time Signature

Our construction will use one-time signature with strong unforgeability as a building block. A one-time signature scheme is a signature scheme that is meant to be used to sign only a single message, and is only required to satisfy unforgeability under properly restricted adversaries that receive only one signature/message pair.

Syntax. To capture the practice better, we augment the usual formalization of general one-time signature scheme to cover the cases that users may share some fixed public parameters.

Definition 2 (One-Time Signature Scheme). *A one-time signature (OTS) scheme consists of the following algorithms:*

- $\mathsf{Setup}(1^n) \rightarrow \mathsf{PP}_{\mathsf{OTS}}$. *This is a probabilistic algorithm. On input the security parameter* 1^n*, the algorithm outputs the system public parameter* $\mathsf{PP}_{\mathsf{OTS}}$.

  The public parameters $\mathsf{PP}_{\mathsf{OTS}}$ are common parameters used by all participants in the system, which may be just the security parameter, or include some additional information such as the message space $\mathcal{M}$, the modulo, etc. In the following, $\mathsf{PP}_{\mathsf{OTS}}$ are implicit input parameters to every algorithm.

- $\mathsf{KeyGen}() \rightarrow (\mathsf{vk}_{\mathsf{OTS}}, \mathsf{sk}_{\mathsf{OTS}})$. *This is a probabilistic algorithm. The algorithm outputs a verification key* $\mathsf{vk}_{\mathsf{OTS}}$ *and a signing key* $\mathsf{sk}_{\mathsf{OTS}}$.

- $\mathsf{Sign}(\mathsf{sk}_{\mathsf{OTS}}, \mu) \rightarrow \Sigma_{\mathsf{OTS}}$. *This is a probabilistic algorithm. On input a signing key* $\mathsf{sk}_{\mathsf{OTS}}$ *and a message* $\mu \in \mathcal{M}$*, the algorithm outputs a signature* Σ_{OTS}.

- $\mathsf{Ver}(\mathsf{vk}_{\mathsf{OTS}}, \mu, \Sigma_{\mathsf{OTS}}) \rightarrow 1/0$. *This is a deterministic algorithm. On input a verification key* $\mathsf{vk}_{\mathsf{OTS}}$*, a message* μ*, and a signature* Σ_{OTS}*, the algorithm outputs 1 if the signature is valid, or 0 if the signature is invalid.*

Correctness. *A OTS scheme is correct, if for any* $n \in \mathbb{N}$*, all messages* $\mu \in \mathcal{M}$*, and any* $\mathsf{PP}_{\mathsf{OTS}} \leftarrow \mathsf{Setup}(1^n)$ *as implicit input parameter to every algorithm, it holds that*

$$\mathsf{Pr}[\mathsf{Ver}(\mathsf{vk}_{\mathsf{OTS}}, \mu, \mathsf{Sign}(\mathsf{sk}_{\mathsf{OTS}}, \mu)) = 1] = 1 - \mathsf{negl}(n),$$

the probability is taken over the random coins used by Setup, KeyGen*, and* Sign.

Strong Unforgeability. *A OTS scheme is strongly unforgeable, if for any PPT forger* $\mathcal{A}$*, it holds that* $\mathcal{A}$ *has at most negligible advantage in the following experiment with a challenger* $\mathcal{C}$.

- **Setup.** $\mathcal{C}$ *generates* $\mathsf{PP}_{\mathsf{OTS}} \leftarrow \mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$ *and* $(\mathsf{vk}_{\mathsf{OTS}}, \mathsf{sk}_{\mathsf{OTS}}) \leftarrow \mathsf{KeyGen}()$*, where* γ_{st} *is randomness used in* Setup*. Finally,* $\mathcal{C}$ *sends* $(\mathsf{PP}_{\mathsf{OTS}}, \mathsf{vk}_{\mathsf{OTS}}, \gamma_{\mathsf{st}})$ *to* $\mathcal{A}$.
  Note that we give to $\mathcal{A}$ the randomness γ_{st} that used for the Setup algorithm, which implies the algorithm is public, does not rely on a trusted setup that may incur concerns on the existing of trapdoors hidden in the output parameters.

- **Probing Phase.** $\mathcal{A}$ *issues a query on message* μ. $\mathcal{C}$ *responses the query by running* $\Sigma_{\mathsf{OTS}} \leftarrow \mathsf{Sign}(\mathsf{sk}_{\mathsf{OTS}}, \mu)$*. Finally, the experiment returns the signature* Σ_{OTS} *to* $\mathcal{A}$.

- **Forge.** $\mathcal{A}$ *outputs a forgery* $(\mu^*, \Sigma^*_{\mathsf{OTS}})$. $\mathcal{A}$ *succeeds if* $(\mu^*, \Sigma^*_{\mathsf{OTS}}) \neq (\mu, \Sigma_{\mathsf{OTS}})$ *and* $\mathsf{Ver}(\mathsf{vk}_{\mathsf{OTS}}, \mu^*, \Sigma^*_{\mathsf{OTS}}) = 1$.

3.2 Key-Homomorphic Evaluation Algorithm

In our construction, we borrow the idea from the standard signature work [14], that is employing the key-homomorphic evaluation algorithm $\mathsf{Eval}(\cdot,\cdot)$ from [26,17,13] to evaluate circuits of a PRF. In particular, they used the evaluation algorithm of the work [17]. The inputs of $\mathsf{Eval}(\cdot,\cdot)$ are C and a set of ℓ different matrices $\{\mathbf{A}^{(i)}\}_{i\in[\ell]}$, where $C : \{0,1\}^\ell \to \{0,1\}$ is a fan-in-2 Boolean NAND circuit expression of some functions such as a PRF, and each $\mathbf{A}^{(i)} = \mathbf{A}\mathbf{R}^{(i)} + b^{(i)}\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ corresponds to each input wire of C, and where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{R}^{(i)} \xleftarrow{\$} \{1,-1\}^{m\times m}$, $b^{(i)} \in \{0,1\}$ and $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is the gadget matrix. The algorithm deterministically output a matrix $\mathbf{A}_C = \mathbf{A}\mathbf{R}_C + C(b^{(1)},\ldots,b^{(\ell)})\mathbf{G} \in \mathbb{Z}_q^{n\times m}$. In the analyzation of our unforgeability proof, we will use the following lemma to show $\mathbf{R}_C$ is short enough.

Lemma 1 **([14]).** *Let* $C : \{0,1\}^\ell \to \{0,1\}$ *be a* NAND *Boolean circuit which has depth* $d = c \log \ell$ *for some constant* c. *Let* $\{\mathbf{A}^{(i)} = \mathbf{A}\mathbf{R}^{(i)} + b^{(i)}\mathbf{G} \in \mathbb{Z}_q^{n\times m}\}_{i\in[\ell]}$ *be* ℓ *different matrices correspond to each input wire of* C *where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{R}^{(i)} \xleftarrow{\$} \{1,-1\}^{m\times m}$, $b^{(i)} \in \{0,1\}$ *and* $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ *is the gadget matrix. There is an efficient deterministic evaluation algorithm* $\mathsf{Eval}(C, (\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(\ell)}))$ *runs in time* $\mathrm{poly}(4^d, \ell, n, \log q)$*, the output of the algorithm is a matrix*

$$\mathbf{A}_C = \mathbf{A}\mathbf{R}_C + C(b^{(1)},\ldots,b^{(\ell)})\mathbf{G} = \mathsf{Eval}(C, (\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(\ell)}))$$

where $C(b^{(1)},\ldots,b^{(\ell)})$ *is the output bit of* C *on the arguments* $(b^{(1)},\ldots,b^{(\ell)})$ *and* $\mathbf{R}_C \in \mathbb{Z}^{m\times m}$ *is a low norm matrix has* $\|\mathbf{R}_C\| \le O(\ell^{2c} \cdot m^{3/2})$.

3.3 Non-Interactive Witness-Indistinguishable Proof Systems

We review the NIWI proof system presented by Gordon et al. [27]. Let $\mathbf{B}^{(1)},\ldots,\mathbf{B}^{(l)} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(l)} \in \mathbb{Z}_q^n$ for some $l = l(n)$, and fix some ε. Define the gap language $L_{\sigma,\varepsilon} = (L_{\mathsf{YES}}, L_{\mathsf{NO}})$ as follows:

$$L_{\mathsf{YES}} = \left\{ \begin{pmatrix} \mathbf{B}^{(1)},\ldots,\mathbf{B}^{(l)} \\ \mathbf{z}^{(1)},\ldots,\mathbf{z}^{(l)} \end{pmatrix} \middle| \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ and } i \in [l] : \left\|\mathbf{z}^{(i)} - (\mathbf{B}^{(i)})^T\mathbf{s}\right\| \le \sigma\sqrt{m} \right\}$$

$$L_{\mathsf{NO}} = \left\{ \begin{pmatrix} \mathbf{B}^{(1)},\ldots,\mathbf{B}^{(l)} \\ \mathbf{z}^{(1)},\ldots,\mathbf{z}^{(l)} \end{pmatrix} \middle| \forall \mathbf{s} \in \mathbb{Z}_q^n \text{ and } i \in [l] : \left\|\mathbf{z}^{(i)} - (\mathbf{B}^{(i)})^T\mathbf{s}\right\| > \varepsilon \cdot \sigma\sqrt{m} \right\}$$

There is an (interactive) statistically witness-indistinguishable proof system for $L_{\sigma,\varepsilon}$ when set $\varepsilon \ge O(\sqrt{m/\log m})$ by using the techniques of the work [39]. Then the resulting protocol can be made non-interactive in the standard model by applying the Fiat-Shamir transformation from the work [41].

Lemma 2. *Let $\varepsilon \geq O(\sqrt{m/\log\, m})$. There is an NIWI proof system for $L_{\sigma,\varepsilon}$ in the standard model.*

3.4 Lattice Backgrounds

Matrix Norms. For a vector $\mathbf{x}$, we let $\|\mathbf{x}\|$ denote its l_2-norm. For a matrix $\mathbf{A}$ we denote two matrix norms: $\|\mathbf{A}\|$ denotes the l_2 length of the longest column of $\mathbf{A}$. $\|\widetilde{\mathbf{A}}\|$ denotes the result of applying Gram-Schmidt orthogonalization to the columns of $\mathbf{A}$.

We will need the following lemma to bound the norm of a random matrix in $\{1,-1\}^{m\times m}$.

Lemma 3 ([4]). *Let $\mathbf{R}$ be a $k \times m$ matrix chosen at random from $\{1,-1\}^{k\times m}$. Then there is a universal constant c such that $\Pr\big[\|\mathbf{R}\| > c\sqrt{k+m}\big] < e^{-(k+m)}$.*

Lattices and Gaussian Distributions. Let $m \in \mathbb{Z}$ be a positive integer and $\mathbf{\Lambda} \subset \mathbb{R}^m$ be an m-dimensional full-rank lattice formed by the set of all integral combinations of m linearly independent basis vectors $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \subset \mathbb{Z}^m$, i.e., $\mathbf{\Lambda} = \mathcal{L}(\mathbf{B}) = \big\{\mathbf{Bc} = \sum_{i=1}^{m} c_i\mathbf{b}_i : \mathbf{c} \in \mathbb{Z}^m\big\}$. For positive integers n, m, q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a vector $\mathbf{y} \in \mathbb{Z}_q^m$, the m-dimensional integer lattice $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$ is defined as $\mathbf{\Lambda}_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{Ax} = \mathbf{0} \pmod q\}$. $\mathbf{\Lambda}_q^{\mathbf{y}}(\mathbf{A})$ is defined as $\mathbf{\Lambda}_q^{\mathbf{y}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{Ax} = \mathbf{y} \pmod q\}$. For a vector $\mathbf{c} \in \mathbb{R}^m$ and a positive parameter $\sigma \in \mathbb{R}$, define $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|/\sigma^2)$ and $\rho_{\sigma,\mathbf{c}}(\mathbf{\Lambda}) = \sum_{\mathbf{x}\in\mathbf{\Lambda}} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. For any $\mathbf{y} \in \mathbf{\Lambda}$, define the discrete Gaussian distribution over $\mathbf{\Lambda}$ with center $\mathbf{c}$ and parameter σ as $\mathcal{D}_{\mathbf{\Lambda},\sigma,\mathbf{c}}(\mathbf{y}) = \rho_{\sigma,\mathbf{c}}(\mathbf{y})/\rho_{\sigma,\mathbf{c}}(\mathbf{\Lambda})$. For simplicity, $\rho_{\sigma,\mathbf{0}}$ and $\mathcal{D}_{\mathbf{\Lambda},\sigma,\mathbf{0}}$ are abbreviated as ρ_σ and $\mathcal{D}_{\mathbf{\Lambda},\sigma}$, respectively.

The following Lemma 4 bounds the length of a discrete Gaussian vector with a sufficiently large Gaussian parameter.

Lemma 4 ([38]). *For any lattice $\mathbf{\Lambda}$ of integer dimension m with basis $\mathbf{B}$, $\mathbf{c} \in \mathbb{R}^m$ and Gaussian parameter $\sigma > \|\widetilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log\, m})$, we have $\Pr[\|\mathbf{x}-\mathbf{c}\| > \sigma\sqrt{m} : \mathbf{x} \leftarrow \mathcal{D}_{\mathbf{\Lambda},\sigma,\mathbf{c}}] \leq \mathrm{negl}(n)$.*

The following generalization of leftover hash lemma is needed for our security proof.

Lemma 5 ([4]). *Suppose that $m > (n+1)\log\, q + \omega(\log\, n)$ and that $q > 2$ is prime. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly in $\{1,-1\}^{m\times k}$ mod q where $k = k(n)$ is polynomial in n. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n\times m}$ and $\mathbb{Z}_q^{n\times k}$ respectively. Then, for all vectors $\mathbf{w}$ in $\mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top\mathbf{w})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top\mathbf{w})$.*

The proofs of our LRS construction is based on the following small integer solution (SIS) assumption, learning with errors (LWE) assumption, and the security of PRF.

Definition 3 (SIS Assumption [38,25]). *Let q, m, β be functions of n. Define $\mathsf{SIS}_{q,n,m,\beta}$ problem as: Given a matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ s.t. $\mathbf{Ax} = \mathbf{0} \pmod q$ and $\|\mathbf{x}\| \leq \beta$.*

For $m, \beta = \mathrm{poly}(n)$, $q \geq \beta \cdot \omega(\sqrt{n \log n})$, no (quantum) algorithm can solve $\mathsf{SIS}_{q,n,m,\beta}$ problem in polynomial time.

We use the LWE assumption proposed by Gordon et al. [27] and they proved it was implied by the standard LWE assumption [42]. The main difference is the error distribution χ choosing from different distribution. Gordon et al. consider the discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,\alpha q}$ where $\alpha q = \omega(\sqrt{\log q})$.

Definition 4 (LWE Assumption [42]). *Let q, m be functions of n, $q > 2$, χ be a discretized normal error distribution parameterized by some $\alpha \in (0,1)$, which is obtained by drawing $x \in \mathbb{R}$ from the Gaussian distribution of width α. Define the LWE distribution $A_{\sigma,\chi}$ as: Choose a vector $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and an error $e \leftarrow \chi$, output $(\mathbf{a}, \mathbf{a}^\top \mathbf{s} + e)$. Defines the Search-$\mathsf{LWE}_{q,n,m,\chi}$ as: Fix an $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, given at most m samples from $A_{\sigma,\chi}$, work out $\mathbf{s}$. Defines the Decision-$\mathsf{LWE}_{q,n,m,\chi}$ as: For a uniformly chosen $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, given the oracle to be (1) $A_{\sigma,\chi}$ or (2) the uniform distribution over $\mathbb{Z}_q^{n+1}$, decide which is the case with at most m oracle calls.*

For $q, m, \alpha =$poly(n) such that $\alpha q = \omega(\sqrt{\log q})$, no (quantum) algorithm can solve the (Search/Decision)-$\mathsf{LWE}_{q,n,m,\chi}$ in polynomial time.

Definition 5 (Pseudorandom Functions). *For a security parameter $n > 0$, let $k = k(n)$, $t = t(n)$ and $c = c(n)$. A pseudorandom function $\mathsf{PRF} : \{0,1\}^k \times \{0,1\}^t \to \{0,1\}^c$ is an efficiently computable, deterministic two-input function where the first input, denoted by K, is the key. Let Ω be the set of all functions that map ℓ bits strings to c bits strings. There is a negligible function $\mathsf{negl}(n)$ such that:*

$$|\Pr\left[\mathcal{A}^{\mathsf{PRF}(K,\cdot)}(1^n) = 1\right] - \Pr\left[\mathcal{A}^{F(\cdot)}(1^n) = 1\right]| \leq \mathsf{negl}(n)$$

where the probability is taken over a uniform choice of key $K \xleftarrow{\$} \{0,1\}^k$ and $F \xleftarrow{\$} \Omega$, and the randomness of $\mathcal{A}$.

Algorithms on Lattices. Our work will use the following lattice algorithms.

Lemma 6 ($\mathsf{TrapGen}$ Algorithm [6]). *Let $n \geq 1, q \geq 2, m = O(n \log q)$ be integers. There is a probabilistic algorithm $\mathsf{TrapGen}(1^n, 1^m, q)$ that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor matrix $\mathbf{T_A} \subset \mathbf{\Lambda}_q^\perp(\mathbf{A})$ i.e., $\mathbf{T_A}$ is a basis (full-rank subset) of $\mathbf{\Lambda}_q^\perp(\mathbf{A})$, the distribution of $\mathbf{A}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n\times m}$ has $\|\widetilde{\mathbf{T_A}}\| \leq O(\sqrt{n \log q})$ and $\|\mathbf{T_A}\| \leq O(n \log q)$ with all but negligible probability in n.*

Lemma 7 (SuperTrapGen **Algorithm** [27])**.** *Let $n \geq 1, q \geq 2, m = O(n \log q)$ be integers. There is a probabilistic algorithm* SuperTrapGen$(1^n, 1^m, q, \mathbf{B})$ *that on input 1^n, 1^m, q, and a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ whose columns generate $\mathbb{Z}_q^n$, this algorithm outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor matrix $\mathbf{T_A} \subset \mathbf{\Lambda}_q^{\perp}(\mathbf{A})$ such that $\mathbf{A}\mathbf{B}^{\top} = \mathbf{0} \pmod q$, and the distribution of $\mathbf{A}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n \times m}$. Moreover, it holds that $\|\widetilde{\mathbf{T_A}}\| = \log n \cdot O(\sqrt{mn \log q})$ with all but negligible probability in n.*

Lemma 8 (BasisExt **Algorithm** [19])**.** *For $i = 1, 2, 3$, let $\mathbf{A}_i$ be a matrix in $\mathbb{Z}_q^{n \times m_i}$ whose columns generate $\mathbb{Z}_q^n$ and let $\mathbf{A}' = [\mathbf{A}_1 | \mathbf{A}_2 | \mathbf{A}_3]$. Let $\mathbf{T}_{\mathbf{A}_2}$ be a basis of $\mathbf{\Lambda}^{\perp}(\mathbf{T}_{\mathbf{A}_2})$. There is a deterministic algorithm* BasisExt$(\mathbf{T}_{\mathbf{A}_2}, \mathbf{A}')$ *that outputs a basis $\mathbf{T}_{\mathbf{A}'}$ for $\mathbf{\Lambda}^{\perp}(\mathbf{A}')$ such that $\|\widetilde{\mathbf{T}_{\mathbf{A}'}}\| = \|\widetilde{\mathbf{T}_{\mathbf{A}_2}}\|$.*

The following lattice basis extension algorithm also needed for our security proof, which presented by Agrawal, Boneh and Boyen [4], so we abbreviate that as BasisExtABB algorithm.

Lemma 9 (BasisExtABB **Algorithm** [4])**.** *Let q be a prime, n, m be integers with $m > n$. There is a probabilistic algorithm* BasisExtABB$(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T_B})$ *which takes as input two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}$ whose columns generate $\mathbb{Z}_q^n$, a matrix $\mathbf{R} \in \mathbb{Z}^{m \times m}$, and a basis $\mathbf{T_B} \in \mathbf{\Lambda}_q^{\perp}(\mathbf{B})$, outputs a full-rank matrix $\mathbf{T_F}$ in $\mathbf{\Lambda}_q^{\perp}(\mathbf{F})$ such that $\|\widetilde{\mathbf{T_F}}\| < (\|\mathbf{R}\| + 1) \cdot \|\widetilde{\mathbf{T_B}}\|$ where $\mathbf{F} = [\mathbf{A} | \mathbf{A}\mathbf{R} + \mathbf{B}] \in \mathbb{Z}_q^{n \times 2m}$.*

Lemma 10 (SamplePre **Algorithm** [25])**.** *Let $q > 2$, $m > n$ be integers. There is a probabilistic algorithm* SamplePre$(\mathbf{A}, \mathbf{T_A}, \mathbf{y}, \sigma)$ *which takes as input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ whose columns generate $\mathbb{Z}_q^n$, and a basis $\mathbf{T_A}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, a vector $\mathbf{y} \in \mathbb{Z}_q^n$, and a Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log m})$, outputs a vector $\mathbf{x} \in \mathbf{\Lambda}_q^{\mathbf{y}}(\mathbf{A})$ sampled from a distribution which is statistically close to $\mathcal{D}_{\mathbf{\Lambda}_q^{\mathbf{y}}(\mathbf{A}), \sigma}$.*

Lemma 11 (SampleR **Algorithm** [5])**.** *Let $q > 2$ be a prime, $m > n$ be integers. There is a probabilistic algorithm* SampleR(1^m) *which outputs a $\mathbb{Z}_q$-invertible matrix $\mathbf{R}$ in $\mathbb{Z}^{m \times m}$ from a distribution that is statistically close to $\mathcal{D}_{m \times m}$ with $\|\widetilde{\mathbf{R}}\| \leq O(\sqrt{m}) \cdot \omega(\sqrt{\log m})$.*

Gadget Matrix. The "gadget matrix" $\mathbf{G}$ defined in [37]. We recall the following one fact of $\mathbf{G}$.

Lemma 12 ([37]). *Let q be a prime, and n, m be integers with $m = n \log q$. There is a fixed full-rank matrix such that the lattice $\mathbf{\Lambda}_q^{\perp}(\mathbf{G})$ has a publicly known basis $\mathbf{T_G} \in \mathbb{Z}^{m \times m}$ with $\|\widetilde{\mathbf{T_G}}\| \leq \sqrt{5}$.*

4 Our Construction

In our construction, the main obstacle is how to design a suitable 'key image'.[2] We use the one-time verification key $\mathsf{vk}_{\mathsf{OTS}}$ as the 'key image'. In particular, we employ Lyubashevsky and Micciancio's work [32] to instantiate the OTS scheme in which the $\mathsf{vk}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}$ where $\mathbf{A}_{\mathsf{com}}$ is a random matrix which can be shared by all users and $\mathbf{T}$ is the $\mathsf{sk}_{\mathsf{OTS}}$. In this setting, the first challenge is achieving $\mathsf{signer\text{-}linkable}$ i.e., how to restrict one ring member generating two signatures by generating the other $\mathsf{vk}'_{\mathsf{OTS}} = \mathbf{A}'_{\mathsf{com}}\mathbf{T}'$ and thus breaking the signer-linkable. For instance, an adversary can trivially break this by generating a $\mathbf{A}'_{\mathsf{com}} \neq \mathbf{A}_{\mathsf{com}}$ or $\mathbf{T}' \neq \mathbf{T}$ such that $\mathsf{vk}'_{\mathsf{OTS}} \neq \mathsf{vk}_{\mathsf{OTS}}$. The second challenge is achieving $\mathsf{signer\text{-}non\text{-}slanderable}$ i.e., how to prevent the adversary to forge the $\mathsf{vk}_{\mathsf{OTS}}$ which belongs to an honest signature tuple i.e., forge a $\mathsf{vk}^*_{\mathsf{OTS}}$ such that $\mathsf{vk}^*_{\mathsf{OTS}} = \mathsf{vk}_{\mathsf{OTS}}$. For instance, an adversary can arbitrarily select a $\mathbf{T}^*$ then compute a $\mathbf{A}^*_{\mathsf{com}}$ such that $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}^*_{\mathsf{com}}\mathbf{T}^* = \mathbf{A}_{\mathsf{com}}\mathbf{T} = \mathsf{vk}_{\mathsf{OTS}}$, or corrupt the $\mathbf{T}$ then compute a $\mathbf{A}^*_{\mathsf{com}}$ such that $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}^*_{\mathsf{com}}\mathbf{T} = \mathbf{A}_{\mathsf{com}}\mathbf{T} = \mathsf{vk}_{\mathsf{OTS}}$.

To address the above two challenges, we present two lattice basis extending algorithms $\mathsf{BasisExtBindAcom}$ and $\mathsf{BasisExtBindSK}$ which used to 'bind' the '$\mathbf{A}_{\mathsf{com}}$' and '$\mathbf{T}$' in $\mathsf{vk}_{\mathsf{OTS}}$, respectively. Our $\mathsf{signer\text{-}linkable}$ and $\mathsf{signer\text{-}non\text{-}slanderable}$ proofs show that, when the 'check vectors' $\mathbf{e}_{\mathsf{chk}}$ and $\mathbf{e}'_{\mathsf{chk}}$ are validated, the adversary can not changed or forged the '$\mathbf{A}_{\mathsf{com}}$' or '$\mathbf{T}$' in $\mathsf{vk}_{\mathsf{OTS}}$ as above mentioned, otherwise the underlying hardness assumption SIS is broken.

4.1 Lattice Basis Extending Algorithms

Algorithm: $\mathsf{BasisExtBindAcom}(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathbf{F})$

Inputs: A matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ whose columns generate $\mathbb{Z}_q^n$, a basis $\mathbf{T}_{\mathbf{A}}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, and a matrix $\mathbf{F} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}} | \mathbf{A}_{\mathsf{com}} + \mathbf{A}\right] \in \mathbb{Z}_q^{n \times 2m}$ where $\mathbf{A}_{\mathsf{com}}$ is a uniformly random matrix in $\mathbb{Z}_q^{n \times m}$.

Outputs: A basis $\mathbf{T}_{\mathbf{F}}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{F})$.

The $\mathsf{BasisExtBindAcom}$ algorithm runs as follows:

1. Sample $\mathbf{R}_0, \mathbf{R}_1 \leftarrow \mathsf{SampleR}(1^m)$.
2. Construct $\mathbf{T}_{\mathbf{F}} = \left[\begin{smallmatrix} -\mathbf{R}_0 & -\mathbf{R}_1 \\ \mathbf{T}_{\mathbf{A}}\mathbf{R}_0 & \mathbf{T}_{\mathbf{A}}\mathbf{R}_1 \end{smallmatrix}\right]$ such that $\mathbf{F} \cdot \mathbf{T}_{\mathbf{F}} = \mathbf{0} \pmod q$.

Lemma 13. *The matrix* $\mathbf{T}_{\mathbf{F}}$ *output by* $\mathsf{BasisExtBindAcom}$ *is full-rank and satisfy* $\|\widetilde{\mathbf{T}_{\mathbf{F}}}\| \leq O(m^{3/2}) \cdot \omega(\sqrt{\log\, m})$.

[2] In our setting, 'key image' is a parameter in the output signature tuple. If two signature tuples have the same 'key image', we say these two signatures are linked.

Proof. By Lemma 11, we know the matrices $\mathbf{R}_0, \mathbf{R}_1$ are invertible. By Lemma 6, we know the basis $\mathbf{T}_{\mathbf{A}}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$ is full-rank. Therefore, the matrix $\mathbf{T}_{\mathbf{F}}$ is full-rank. By Lemma 11, we know the Gram-Schmidt norm of $\mathbf{R}_0, \mathbf{R}_1$ bounded by $O(\sqrt{m}) \cdot \omega(\sqrt{\log\, m})$. By Lemma 6, we know $\|\widetilde{\mathbf{T}_{\mathbf{A}}}\| \leq O(\sqrt{n\, \log\, q})$. As analyzed in Sect. 4.3, it requires to set $m = O(n\, \log\, q)$. Therefore, we have $\|\widetilde{\mathbf{T}_{\mathbf{F}}}\| \leq O(m^{3/2}) \cdot \omega(\sqrt{\log\, m})$.

Algorithm: $\mathsf{BasisExtBindSK}(\mathbf{T}_{\mathbf{A}}, \mathbf{F})$

Inputs: A matrix $\mathbf{F} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}} - \mathbf{A}_{\mathsf{com}} | \mathbf{A}_{\mathsf{com}}\right] \in \mathbb{Z}_q^{n \times 2m}$ where $\mathbf{A}_{\mathsf{com}}$ is a uniformly random matrix in $\mathbb{Z}_q^{n \times m}$, and $\mathbf{T}_{\mathbf{A}}$ is a basis of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$ and $\mathbf{A}$ is independent with $\mathbf{A}_{\mathsf{com}}$.

Outputs: A basis $\mathbf{T}_{\mathbf{F}}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{F})$.

The $\mathsf{BasisExtBindSK}$ algorithm runs as follows:

1. Sample $\mathbf{R}_0, \mathbf{R}_1 \leftarrow \mathsf{SampleR}(1^m)$.
2. Construct $\mathbf{T}_{\mathbf{F}} = \left[\begin{smallmatrix} -\mathbf{R}_0 & -\mathbf{R}_1 \\ \mathbf{T}_{\mathbf{A}}\mathbf{R}_0 - \mathbf{R}_0 & \mathbf{T}_{\mathbf{A}}\mathbf{R}_1 - \mathbf{R}_1 \end{smallmatrix}\right]$ such that $\mathbf{F} \cdot \mathbf{T}_{\mathbf{F}} = \mathbf{0} \pmod q$.

Lemma 14. *The matrix* $\mathbf{T}_{\mathbf{F}}$ *output by* $\mathsf{BasisExtBindSK}$ *is full-rank and satisfy* $\|\widetilde{\mathbf{T}_{\mathbf{F}}}\| \leq O(m^{3/2}) \cdot \omega(\sqrt{\log\, m})$.

Proof. The proof is as same as the proof of Lemma 13.

4.2 Construction

$\mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$

1. On input the a security parameter n, sets the parameters q, m, k, σ as specified in Sect. 4.3 below.
2. Select a secure $\mathsf{PRF} : \{0,1\}^k \times \{0,1\}^t \to \{0,1\}$, express it as a NAND Boolean circuit C_{PRF}.
3. Let Π_{OTS} be a one-time signature scheme with strong unforgeability.
4. Compute $\mathsf{PP}_{\mathsf{OTS}} \leftarrow \Pi_{\mathsf{OTS}}.\mathsf{Setup}(1^n)$.
5. Sample $\mathbf{A}_{\mathsf{com}} \leftarrow \mathsf{XOF}(\gamma_{\mathsf{st}})$ where $\mathbf{A}_{\mathsf{com}} \in \mathbb{Z}_q^{n \times m}$.
6. Output the public parameters $\mathsf{PP} = (q, m, k, \sigma, \mathsf{PRF}, (\Pi_{\mathsf{OTS}}, \mathsf{PP}_{\mathsf{OTS}}), \mathbf{A}_{\mathsf{com}}, \gamma_{\mathsf{st}})$.

Note that including the randomness γ_{st} in PP and sample the $\mathbf{A}_{\mathsf{com}}$ by the extendable output function XOF [40] is to guarantee the public has no concerns on the existing of trapdoors for PP.

In the following, PP are implicit input parameters to every algorithm.

$\mathsf{KeyGen}()$

1. Select $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow \mathsf{SuperTrapGen}(1^n, 1^m, q, \mathbf{B})$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$.
2. Let $\mathsf{sk}_{\mathsf{OTS}} := \mathbf{T}_{\mathbf{A}}$ and set $\mathsf{vk}_{\mathsf{OTS}} := \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$.

3. Select a PRF key $\mathbf{k} = (k_1, k_2, \ldots, k_k) \xleftarrow{\$} \{0,1\}^k$.
4. For $j = 1$ to k, select $\mathbf{B}_j \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
5. Select $\mathbf{A}_0, \mathbf{A}_1, \mathbf{C}_0, \mathbf{C}_1 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
6. Output $\mathsf{vk} = (\mathbf{A}, (\mathbf{A}_0, \mathbf{A}_1), \mathbf{B}, \{\mathbf{B}_j\}_{j\in[k]}, (\mathbf{C}_0, \mathbf{C}_1))$ and $\mathsf{sk} = (\mathbf{T}_{\mathbf{A}}, \mathbf{k}, \mathsf{vk}_{\mathsf{OTS}})$.

$\mathsf{Sign}(\mathsf{sk}, \boldsymbol{\mu}, \mathsf{R})$

1. On input a signing key $\mathsf{sk} := \mathsf{sk}^{(s)} = (\mathbf{T}_{\mathbf{A}}^{(s)}, \mathbf{k}^{(s)}, \mathsf{vk}_{\mathsf{OTS}}^{(s)})$ where $s \in [N]$ is the index of the signer in the ring R, a message $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_t) \in \{0,1\}^t$, and a ring of verification keys $\mathsf{R} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$ where each $\mathsf{vk}^{(i)} = (\mathbf{A}^{(i)}, (\mathbf{A}_0^{(i)}, \mathbf{A}_1^{(i)}), \mathbf{B}^{(i)}, \{\mathbf{B}_j^{(i)}\}_{j\in[k]}, (\mathbf{C}_0^{(i)}, \mathbf{C}_1^{(i)}))$.
2. For $i = 1$ to N, set $\mathbf{F}^{(i)} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} | \mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(i)}\right] \in \mathbb{Z}_q^{n\times 2m}$. Let $\mathbf{F}_{\mathsf{chk}} = [\mathbf{F}^{(1)}| \ldots |\mathbf{F}^{(N)}] \in \mathbb{Z}_q^{n\times 2Nm}$.
3. For $i = s$, compute $\mathbf{T}_{\mathbf{F}^{(s)}} \leftarrow \mathsf{BasisExtBindAcom}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}}, \mathbf{F}^{(s)}), \mathbf{T}_{\mathbf{F}_{\mathsf{chk}}} \leftarrow \mathsf{BasisExt}(\mathbf{T}_{\mathbf{F}^{(s)}}, \mathbf{F}_{\mathsf{chk}})$, and $\mathbf{e}_{\mathsf{chk}} \leftarrow \mathsf{SamplePre}(\mathbf{F}_{\mathsf{chk}}, \mathbf{T}_{\mathbf{F}_{\mathsf{chk}}}, \mathbf{0}, \sigma)$.
4. Let $\mathbf{F}'_{\mathsf{chk}} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} - \mathbf{A}_{\mathsf{com}} | \mathbf{A}_{\mathsf{com}}\right] \in \mathbb{Z}_q^{n\times 2m}$. Compute $\mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}} \leftarrow \mathsf{BasisExtBindSK}$ $(\mathbf{T}_{\mathbf{A}^{(s)}}, \mathbf{F}'_{\mathsf{chk}})$ and $\mathbf{e}'_{\mathsf{chk}} \leftarrow \mathsf{SamplePre}(\mathbf{F}'_{\mathsf{chk}}, \mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}}, \mathbf{0}, \sigma)$.
5. Compute $d = \mathsf{PRF}(\mathbf{k}^{(s)}, \boldsymbol{\mu})$.
6. For $i = 1$ to N, compute $\mathbf{A}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}^{(i)} = \mathsf{Eval}(C_{\mathsf{PRF}}, (\{\mathbf{B}_j^{(i)}\}_{j\in[k]}, \mathbf{C}_{\mu_1}^{(i)}, \mathbf{C}_{\mu_2}^{(i)}, \ldots, \mathbf{C}_{\mu_t}^{(i)}))$ $\in \mathbb{Z}_q^{n\times m}$, set $\mathbf{F}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}^{(i)} = \left[\mathbf{A}^{(i)} | \mathbf{A}_{1-d}^{(i)} - \mathbf{A}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}^{(i)}\right] \in \mathbb{Z}_q^{n\times 2m}$.
7. For $i = 1$ to N, select $\mathbf{u}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^n$, compute $\mathbf{e}_1^{(i)} \leftarrow \mathsf{SamplePre}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}}, \mathbf{u}^{(i)}, \sigma)$.
8. For $i = s$, compute $\mathbf{e}_0^{(s)} \leftarrow \mathsf{SamplePre}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}},\ \mathbf{u}'^{(s)}, \sigma)$ where $\mathbf{u}'^{(s)} = (\mathbf{A}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}^{(s)} - \mathbf{A}_{1-d}^{(s)}) \cdot \mathbf{e}_1^{(s)}$.
9. For $i = s+1, \ldots, N, 1, \ldots, s-1$, uniformly choose $\mathbf{e}_0^{(i)} \in \mathbb{Z}^m$ subject to the condition that $\mathbf{F}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}^{(i)} \cdot (\mathbf{e}_0^{(i)}; \mathbf{e}_1^{(i)}) = \mathbf{0} \pmod q$.
10. For $i = 1$ to N, select $\mathbf{s}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^n$, compute $\mathbf{z}^{(i)} = (\mathbf{s}^{(i)})^\top \mathbf{B}^{(i)} + \mathbf{e}_0^{(i)}$.
11. Use the witness $\{\mathbf{s}^{(i)}, i\}_{i\in[N]}$ to construct an NIWI proof π for the gap language $L_{\sigma,\varepsilon}$.
12. Compute one-time signature $\Sigma_{\mathsf{OTS}} \leftarrow \Pi_{\mathsf{OTS}}.\mathsf{Sign}(\mathsf{sk}_{\mathsf{OTS}}, \boldsymbol{\mu})$.
13. Output the signature $\Sigma = (\Sigma_{\mathsf{OTS}}, \mathsf{vk}_{\mathsf{OTS}}, \mathbf{e}_{\mathsf{chk}}, \mathbf{e}'_{\mathsf{chk}}, \{\mathbf{e}_1^{(i)}, \mathbf{z}^{(i)}\}_{i\in[N]}, \pi)$.

$\mathsf{Ver}(\mathsf{R}, \boldsymbol{\mu}, \Sigma)$

1. On input a ring of verification keys $\mathsf{R} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$ where each $\mathsf{vk}^{(i)} = (\mathbf{A}^{(i)}, (\mathbf{A}_0^{(i)}, \mathbf{A}_1^{(i)}), \mathbf{B}^{(i)}, \{\mathbf{B}_j^{(i)}\}_{j\in[k]}, (\mathbf{C}_0^{(i)}, \mathbf{C}_1^{(i)}))$, a message $\boldsymbol{\mu}$, and a signature $\Sigma = (\Sigma_{\mathsf{OTS}}, \mathsf{vk}_{\mathsf{OTS}}, \mathbf{e}_{\mathsf{chk}}, \mathbf{e}'_{\mathsf{chk}}, \{\mathbf{e}_1^{(i)}, \mathbf{z}^{(i)}\}_{i\in[N]}, \pi)$.
2. Compute $(\mathbf{F}_{\mathsf{chk}}, \mathbf{F}'_{\mathsf{chk}})$ as in Sign algorithm. Check if $\|\mathbf{e}_{\mathsf{chk}}\| \le \sigma\sqrt{2Nm}$ and $\|\mathbf{e}'_{\mathsf{chk}}\| \le \sigma\sqrt{2m}$ and $\mathbf{F}_{\mathsf{chk}} \cdot \mathbf{e}_{\mathsf{chk}} = \mathbf{0} \pmod q$ and $\mathbf{F}'_{\mathsf{chk}} \cdot \mathbf{e}'_{\mathsf{chk}} = \mathbf{0} \pmod q$ holds, otherwise return 0.

3. For $i = 1$ to N and $d \in \{0, 1\}$, compute $\mathbf{F}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},d} = \left[\mathbf{A}^{(i)} | \mathbf{A}^{(i)}_d - \mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}\right]$ where $\mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}$ is computed as in Sign algorithm. Check if each $\|\mathbf{e}_1^{(i)}\| \leq \sigma\sqrt{m}$ and $\mathbf{F}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},d} \cdot (\mathbf{z}^{(i)}; \mathbf{e}_1^{(i)}) = \mathbf{0} \pmod q$ holds for $d = 0$ or 1, otherwise return 0.
4. Check if the proof π is correct and $\Pi_{\mathsf{OTS}}.\mathsf{Ver}(\mathsf{vk}_{\mathsf{OTS}}, \boldsymbol{\mu}, \varSigma_{\mathsf{OTS}}) = 1$, return 1, otherwise return 0.

$\mathsf{Link}(\mathsf{R}_0, \boldsymbol{\mu}_0, \varSigma_0, \mathsf{R}_1, \boldsymbol{\mu}_1, \varSigma_1)$

1. On input two valid signature tuples $(\mathsf{R}_0, \boldsymbol{\mu}_0, \varSigma_0)$ and $(\mathsf{R}_1, \boldsymbol{\mu}_1, \varSigma_1)$. Let $\mathsf{vk}_{\mathsf{OTS},0}$ and $\mathsf{vk}_{\mathsf{OTS},1}$ be the one-time verification keys in $\varSigma_0$ and $\varSigma_1$, respectively.
2. Check if $\mathsf{vk}_{\mathsf{OTS},0} = \mathsf{vk}_{\mathsf{OTS},1}$ holds, return 1, otherwise return 0.

4.3 Correctness and Parameters

We now show the correctness of LRS. By Lemma 10, each $\mathbf{e}_1^{(i)}$ in $\varSigma$ follows the distribution $\mathcal{D}_{\boldsymbol{\Lambda}_q^{\mathbf{u}^{(i)}}(\mathbf{A}^{(s)}),\sigma}$, then by the construction of $\mathbf{z}^{(i)}$ and Lemma 7, it holds that $\mathbf{F}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},d} \cdot (\mathbf{z}^{(i)}; \mathbf{e}_1^{(i)}) = \mathbf{0} \pmod q$ for $d = 0$ or 1. By Lemma 10, the $\mathbf{e}_{\mathsf{chk}}$ and $\mathbf{e}'_{\mathsf{chk}}$ in $\varSigma$ respectively follow the distribution $\mathcal{D}_{\boldsymbol{\Lambda}_q^{\perp}(\mathbf{F}_{\mathsf{chk}}),\sigma}$ and $\mathcal{D}_{\boldsymbol{\Lambda}_q^{\perp}(\mathbf{F}'_{\mathsf{chk}}),\sigma}$, therefore, it holds that $\mathbf{F}_{\mathsf{chk}} \cdot \mathbf{e}_{\mathsf{chk}} = \mathbf{0} \pmod q$ and $\mathbf{F}'_{\mathsf{chk}} \cdot \mathbf{e}'_{\mathsf{chk}} = \mathbf{0} \pmod q$. By Lemma 4, $\mathbf{e}_1^{(i)} \leq \sigma\sqrt{m}$, $\mathbf{e}_{\mathsf{chk}} \leq \sigma\sqrt{2Nm}$, and $\mathbf{e}'_{\mathsf{chk}} \leq \sigma\sqrt{2m}$ hold with overwhelming probability. Therefore, the signature is accepted by the Ver algorithm with overwhelming probability. For the correctness of algorithm Link, the signer-linkable proof in Sect. 4.4 shows the correctness holds with overwhelming probability.

We then explain the parameters choosing. We employ the work [31] to instantiate our PRF, which based on standard LWE assumption with polynomial modulus $q = n^{\omega(1)}$. We employ the work [32] to instantiate our OTS, which requires $\|\mathbf{T}_{\mathbf{A}}\|_\infty \leq p$ and $q \geq 2tp\sqrt{m}n^{\varOmega(1)}$ where $p = \lceil \frac{q^{n/m}2^{n/m}-1}{2} \rceil$. To guarantee the hardness of the based $\mathsf{LWE}_{q,n,m,\chi}$ assumption, we need to set $\alpha = \omega(\sqrt{\log q})/q$ as defined in Definition 4. Let n be the security parameter, let the message length be $t = t(n)$ and the secret key length of PRF be $k = k(n)$. Let $\ell = t + k$ be the input length of PRF. To ensure that hard lattices with good short bases can be generated by $\mathsf{TrapGen}$ and $\mathsf{SuperTrapGen}$, we need to set $m = 6n^{1+\delta}$ where $\delta > 0$ is a constant such that $n^\delta > O(\log n)$. To ensure that the distribution on the output of $\mathsf{SamplePre}$ statistically close to the distribution $\mathcal{D}_{\boldsymbol{\Lambda}_q^{\perp}(\mathbf{F}'),\sigma}$, we need to set σ sufficiently large that is $\sigma = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$ (see the signer-non-slanderable proof below). To ensure that vectors sampled using a trapdoor are difficult SIS solutions, we need to set $\beta = O(\ell^{4c} \cdot m^{7/2}) \cdot \omega(\sqrt{\log m})$ such that $\beta \geq O(\ell^{2c} \cdot m^2) \cdot \sigma$ for some constant c (see the signer-non-slanderable proof below). To ensure our construction based on SIS has a worst-case lattice reduction as defined in Definition 3, we need to set $q = O(\ell^{4c} \cdot m^4) \cdot (\omega(\sqrt{\log m}))^2$ such that $q \geq \beta \cdot \omega(\sqrt{n \log n})$.

5 Proofs of Security and Privacy

Theorem 1 (Unforgeability). *Let $m, q, \beta, \alpha, \sigma$ be some polynomials in the security parameter n. For large enough $\sigma = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$ and $\beta = O(\ell^{4c} \cdot m^{7/2}) \cdot \omega(\sqrt{\log m})$, the LRS scheme is* $\mathsf{sUnfInsCor}$ *secure in the standard model.*

Proof. Consider the following security game between an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$. Upon receiving a challenge $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ that is formed by m uniformly random and independent samples from $\mathbb{Z}_q^n$ and the $(\mathsf{PP}_{\mathsf{OTS}}, \mathsf{vk}_{\mathsf{OTS}})$, $\mathcal{S}$ simulates as follows.

Setup. $\mathcal{S}$ takes as input a security parameter n and a randomness γ_{st} to invoke $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n; \mathsf{PP}_{\mathsf{OTS}}, \gamma_{\mathsf{st}})$ algorithm. $\mathcal{S}$ simulates as follows.

- Select a random index $i^{\diamond} \xleftarrow{\$} \{1, \dots, N\}$ and sets $\mathbf{A}^{(i^{\diamond})} = \mathbf{A}$, then sample $(\mathbf{B}^{(i^{\diamond})}, \mathbf{T}_{\mathbf{B}^{(i^{\diamond})}}) \leftarrow \mathsf{SuperTrapGen}(1^n, 1^m, q, \mathbf{A}^{(i^{\diamond})})$.
- For $i = i^{\diamond} + 1, \dots, N, 1, \dots, i^{\diamond} - 1$:
  - Compute $(\mathbf{B}^{(i)}, \mathbf{T}_{\mathbf{B}^{(i)}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
  - Compute $(\mathbf{A}^{(i)}, \mathbf{T}_{\mathbf{A}^{(i)}}) \leftarrow \mathsf{SuperTrapGen}(1^n, 1^m, q, \mathbf{B}^{(i)})$.
  - Let $\mathsf{sk}_{\mathsf{OTS}}^{(i)} = \mathbf{T}_{\mathbf{A}^{(i)}}$ and $\mathsf{vk}_{\mathsf{OTS}}^{(i)} = \mathbf{A}_{\mathsf{com}} \mathbf{T}_{\mathbf{A}^{(i)}}$.
- For $i = i^{\diamond}$, set $\mathsf{vk}_{\mathsf{OTS}}^{(i^{\diamond})} = \mathsf{vk}_{\mathsf{OTS}}$.
- For $i = 1$ to N and $d \in \{0, 1\}$:
  - Choose $\mathbf{R}_{\mathbf{A}_d^{(i)}}, \mathbf{R}_{\mathbf{C}_d^{(i)}} \xleftarrow{\$} \{1, -1\}^{m \times m}$.
  - Construct $\mathbf{A}_d^{(i)} = \mathbf{A}^{(i)} \mathbf{R}_{\mathbf{A}_d^{(i)}} + d\mathbf{G}$ and $\mathbf{C}_d^{(i)} = \mathbf{A}^{(i)} \mathbf{R}_{\mathbf{C}_d^{(i)}} + d\mathbf{G}$ where $\mathbf{G}$ is the gadget matrix.
- For $i = 1$ to N:
  - Select a PRF key $\mathbf{k}^{(i)} = (k_1^{(i)}, k_2^{(i)}, \dots, k_k^{(i)}) \xleftarrow{\$} \{0, 1\}^k$.
- For $i = 1$ to N and $j = 1$ to k:
  - Choose $\mathbf{R}_{\mathbf{B}_j^{(i)}} \xleftarrow{\$} \{1, -1\}^{m \times m}$ and construct $\mathbf{B}_j^{(i)} = \mathbf{A}^{(i)} \mathbf{R}_{\mathbf{B}_j^{(i)}} + k_j^{(i)} \mathbf{G}$.
- Let $\mathsf{S} = (\mathsf{vk}^{(1)}, \dots, \mathsf{vk}^{(N)})$ where each $\mathsf{vk}^{(i)} = (\mathbf{A}^{(i)}, (\mathbf{A}_0^{(i)}, \mathbf{A}_1^{(i)}), \mathbf{B}^{(i)}, \{\mathbf{B}_j^{(i)}\}_{j \in [k]}, (\mathbf{C}_0^{(i)}, \mathbf{C}_1^{(i)}))$, then sends $(\mathsf{PP}, \mathsf{S}, \gamma_{\mathsf{st}})$ to $\mathcal{A}$.

Probing Signing Oracle. $\mathcal{A}$ adaptively issues tuples for querying the signing oracle $\mathsf{OSign}(\cdot, \cdot, \cdot)$. For simplicity, here consider only one tuple $(\boldsymbol{\mu}, \mathsf{R}, s)$ where $s \in [N]$, and requires that $\mathsf{vk}^{(s)} \in \mathsf{S} \bigcap \mathsf{R}$. Let $N' = |\mathsf{R}|$. Assume the ring $\mathsf{R} = (\mathsf{vk}^{(1)}, \dots, \mathsf{vk}^{(N')})$, parse $\mathsf{vk}^{(s)} = (\mathbf{A}^{(s)}, (\mathbf{A}_0^{(s)}, \mathbf{A}_1^{(s)}), \mathbf{B}^{(s)}, \{\mathbf{B}_j^{(s)}\}_{j \in [k]}, (\mathbf{C}_0^{(s)}, \mathbf{C}_1^{(s)}))$. $\mathcal{S}$ does the following to response the signature.

- For $i' = 1$ to N', set $\mathbf{F}^{(i')} = [\mathbf{A}_{\mathsf{com}} \mathbf{T}_{\mathbf{A}^{(i')}} | \mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(i')}] \in \mathbb{Z}_q^{n \times 2m}$. Let $\mathbf{F}_{\mathsf{chk}} = [\mathbf{F}^{(1)} | \dots | \mathbf{F}^{(N)}] \in \mathbb{Z}_q^{n \times 2Nm}$.

- Select $\hat{s} \xleftarrow{\$} \{1, 2, \ldots, N'\} \backslash i^{\diamond}$. Compute $\mathbf{T}_{\mathbf{F}^{(\hat{s})}} \leftarrow \mathsf{BasisExtBindAcom}(\mathbf{A}^{(\hat{s})}, \mathbf{T}_{\mathbf{A}^{(\hat{s})}}, \mathbf{F}^{(\hat{s})})$, $\mathbf{T}_{\mathbf{F}_{\mathsf{chk}}} \leftarrow \mathsf{BasisExt}(\mathbf{T}_{\mathbf{F}^{(\hat{s})}}, \mathbf{F}_{\mathsf{chk}})$, and $\mathbf{e}_{\mathsf{chk}} \leftarrow \mathsf{SamplePre}(\mathbf{F}_{\mathsf{chk}}, \mathbf{T}_{\mathbf{F}_{\mathsf{chk}}}, \mathbf{0}, \sigma)$.
- Let $\mathbf{F}'_{\mathsf{chk}} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(\hat{s})}} - \mathbf{A}_{\mathsf{com}} | \mathbf{A}_{\mathsf{com}}\right] \in \mathbb{Z}_q^{n \times 2m}$. Compute $\mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}} \leftarrow \mathsf{BasisExtBindSK}(\mathbf{T}_{\mathbf{A}^{(\hat{s})}}, \mathbf{F}'_{\mathsf{chk}})$ and $\mathbf{e}'_{\mathsf{chk}} \leftarrow \mathsf{SamplePre}(\mathbf{F}'_{\mathsf{chk}}, \mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}}, \mathbf{0}, \sigma)$.
- Compute $d = \mathsf{PRF}(\mathbf{k}^{(\hat{s})}, \boldsymbol{\mu})$.
- For $i' = 1$ to N':
  - Compute $\mathbf{A}^{(i')}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}} = \mathsf{Eval}(C_{\mathsf{PRF}}, (\{\mathbf{B}_j\}^{(i')}_{j \in [k]}, \mathbf{C}^{(i')}_{\mu_1}, \mathbf{C}^{(i')}_{\mu_2}, \ldots, \mathbf{C}^{(i')}_{\mu_t}))$
  - Set $\mathbf{F}^{(i')}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}, 1-d} = \left[\mathbf{A}^{(i')} | \mathbf{A}^{(i')}_{1-d} - \mathbf{A}^{(i')}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}}\right]$.
  - Select $\mathbf{u}^{(i')} \xleftarrow{\$} \mathbb{Z}_q^n$.
- For $i' = i^{\diamond} + 1, \ldots, N', 1, \ldots, i^{\diamond} - 1$:
  - Compute $\mathbf{e}_1^{(i)} \leftarrow \mathsf{SamplePre}(\mathbf{A}^{(\hat{s})}, \mathbf{T}_{\mathbf{A}^{(\hat{s})}}, \mathbf{u}^{(i')}, \sigma)$.
  - Uniformly choose $\mathbf{e}_0^{(i')} \in \mathbb{Z}^m$ subject to the condition that $\mathbf{F}^{(i')}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}, 1-d} \cdot (\mathbf{e}_0^{(i')}; \mathbf{e}_1^{(i')}) = \mathbf{0} \pmod q$.
- For $i' = i^{\diamond}$, note that $\mathbf{F}^{(i^{\diamond})}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}, 1-d}$ can be transformed to

$$\begin{aligned} \mathbf{F}^{(i^{\diamond})}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}, 1-d} &= \left[\mathbf{A}^{(i^{\diamond})} | \mathbf{A}^{(i^{\diamond})}_{1-d} - \mathbf{A}^{(i^{\diamond})}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}}\right] \\ &= \left[\mathbf{A}^{(i^{\diamond})} | \mathbf{A}^{(i^{\diamond})}(\mathbf{R}_{\mathbf{A}^{(i^{\diamond})}_{1-d}} - \mathbf{R}^{(i^{\diamond})}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}}) + (1-2d)\mathbf{G}\right] \in \mathbb{Z}_q^{n \times 2m} \end{aligned}$$

  then we can extend $\mathbf{T}_{\mathbf{G}}$ to $\mathbf{T}_{\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}, 1-d}}$ by $\mathsf{BasisExtABB}$, then compute $(\mathbf{e}_0^{i\diamond}; \mathbf{e}_1^{i\diamond})$ by $\mathsf{SamplePre}(\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}, 1-d}, \mathbf{T}_{\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}, 1-d}}, \sigma, \mathbf{0})$.
- For $i' = 1$ to N', sample $\mathbf{s}^{(i')} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{z}^{(i')} = (\mathbf{B}^{(i')})^\top \mathbf{s}^{(i')} + \mathbf{e}_0^{(i')}$.
- Construct an NIWI proof π for the gap language $L_{\sigma, \varepsilon}$ by using the witness $\{\mathbf{s}^{(i')}, i'\}_{i' \in [N']}$.
- If $i' \neq i^{\diamond}$, compute one-time signature $\Sigma_{\mathsf{OTS}} \leftarrow \Pi_{\mathsf{OTS}}.\mathsf{Sign}(\mathsf{sk}^{(s)}_{\mathsf{OTS}}, \boldsymbol{\mu})$. If $i' = i^{\diamond}$, send $\boldsymbol{\mu}$ to one-time signature challenger and then receive the response Σ_{OTS}.
- Return the signature $\Sigma = (\Sigma_{\mathsf{OTS}}, \mathsf{vk}^{(s)}_{\mathsf{OTS}}, \mathbf{e}_{\mathsf{chk}}, \mathbf{e}'_{\mathsf{chk}}, \{\mathbf{e}_1^{(i')}, \mathbf{z}^{(i')}\}_{i' \in [N']}, \pi)$ to $\mathcal{A}$ and adds $(\boldsymbol{\mu}, \mathsf{R}, \Sigma)$ to a list L which $\mathcal{S}$ initialized in prior.

Probing Corrupting Oracle. $\mathcal{A}$ adaptively issues index i for querying the corrupting oracle $\mathsf{OCorrupt}(\cdot)$, $\mathcal{S}$ returns $\mathsf{sk}^{(i)}$ to $\mathcal{A}$ and adds $\mathsf{vk}^{(i)}$ to a set C which $\mathcal{S}$ initialized in prior, while if $i = i^{\diamond}$ then $\mathcal{S}$ aborts.

Exploiting the Forgery. $\mathcal{A}$ outputs one forgery $(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*)$. Let $N^* = |\mathsf{R}^*|$. Parse $\boldsymbol{\mu}^* = (\mu_1^*, \ldots, \mu_t^*)$ and $\mathsf{R}^* = (\mathsf{vk}^{*(1)}, \ldots, \mathsf{vk}^{*(N^*)})$ where each $\mathsf{vk}^{(i^*)} = (\mathbf{A}^{(i^*)}, (\mathbf{A}_0^{(i^*)}, \mathbf{A}_1^{(i^*)}), \mathbf{B}^{(i^*)}, \{\mathbf{B}_j^{(i^*)}\}_{j \in [k]}, (\mathbf{C}_0^{(i^*)}, \mathbf{C}_1^{(i^*)}))$. Parse $\Sigma^* = (\Sigma^*_{\mathsf{OTS}}, \mathsf{vk}^*_{\mathsf{OTS}}, \mathbf{e}^*_{\mathsf{chk}}, \mathbf{e}'^*_{\mathsf{chk}}, \{\mathbf{e}_1^{(i^*)}, \mathbf{z}^{(i^*)}\}_{i^* \in [N^*]}, \pi^*)$. $\mathcal{S}$ does the following to exploit the forgery.

- Check if $\mathsf{Ver}(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*) = 1$ and $(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*) \notin \mathsf{L}$ and $\mathsf{R}^* \subseteq \mathsf{S} \setminus \mathsf{C}$, otherwise $\mathcal{S}$ aborts.
- Compute $d = \mathsf{PRF}(\mathbf{k}^{(i^\diamond)}, \boldsymbol{\mu}^*)$.
- For $i^* = i^\diamond$:
  - Compute $\mathbf{A}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*} = \mathbf{A}^{(i^\diamond)}\mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*} + \mathsf{PRF}(\mathbf{k}^{(i^\diamond)}, \boldsymbol{\mu}^*)\mathbf{G}$ by invoking the $\mathsf{Eval}(C_{\mathsf{PRF}}, (\{\mathbf{B}^{(i^\diamond)}_j\}_{j\in[k]}, \mathbf{C}^{(i^\diamond)}_{\mu^*_1}, \mathbf{C}^{(i^\diamond)}_{\mu^*_2}, \ldots, \mathbf{C}^{(i^\diamond)}_{\mu^*_t}))$.
  - Set $\mathbf{F}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*,d} = \left[\mathbf{A}^{(i^\diamond)} | \mathbf{A}^{(i^\diamond)}_d - \mathbf{A}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}\right]$.
- Use $\mathbf{T}_{\mathbf{B}^{(i^\diamond)}}$ to recover $\mathbf{e}_0^{(i^\diamond)}$. Then check if $\|\mathbf{e}_0^{(i^\diamond)}\| \le \sigma\sqrt{m}$ and $\mathbf{F}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*,d} \cdot (\mathbf{z}^{(i^\diamond)}; \mathbf{e}_1^{(i^\diamond)}) = \mathbf{0} \pmod q$ holds, otherwise $\mathcal{S}$ aborts.
- Note that the equation $\mathbf{F}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*,d} \cdot (\mathbf{z}^{(i^\diamond)}; \mathbf{e}_1^{(i^\diamond)}) = \mathbf{0} \pmod q$ can be transformed to the following

$$\left[\mathbf{A}^{(i^\diamond)} | \mathbf{A}^{(i^\diamond)}_d - \mathbf{A}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}\right] \cdot (\mathbf{z}^{(i^\diamond)}; \mathbf{e}_1^{(i^\diamond)}) = \mathbf{0} \pmod q$$

$$\left[\mathbf{A}^{(i^\diamond)} | \mathbf{A}^{(i^\diamond)}(\mathbf{R}_{\mathbf{A}^{(i^\diamond)}_d} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}) + (d - \mathsf{PRF}(\mathbf{k}^{(i^\diamond)}, \boldsymbol{\mu}^*))\right] \cdot (\mathbf{z}^{(i^\diamond)}; \mathbf{e}_1^{(i^\diamond)}) = \mathbf{0} \pmod q$$

$$\left[\mathbf{A}^{(i^\diamond)} | \mathbf{A}^{(i^\diamond)}(\mathbf{R}_{\mathbf{A}^{(i^\diamond)}_d} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*})\right] \cdot ((\mathbf{B}^{(i^\diamond)})^\top \mathbf{s}^{(i^\diamond)} + \mathbf{e}_0^{(i^\diamond)}; \mathbf{e}_1^{(i^\diamond)}) = \mathbf{0} \pmod q$$

$$\mathbf{A}^{(i^\diamond)} \cdot (\mathbf{e}_0^{(i^\diamond)} + (\mathbf{R}_{\mathbf{A}^{(i^\diamond)}_d} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}) \cdot \mathbf{e}_1^{(i^\diamond)}) = \mathbf{0} \pmod q$$

- Return $\mathbf{e}_0^{(i^\diamond)} + (\mathbf{R}_{\mathbf{A}^{(i^\diamond)}_d} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}) \cdot \mathbf{e}_1^{(i^\diamond)}$ as a $\mathsf{SIS}_{q,n,m,\beta}$ solution, and return $(\Sigma^*_{\mathsf{OTS}}, \boldsymbol{\mu}^*)$ as the forged one-time signature.

Claim 1. *The public parameters* PP *and the set of verifications keys* S *that simulated by* $\mathcal{S}$ *is statistically close to those in the real attack.*

Proof. In the real scheme and the simulation, the matrix $\mathbf{A}_{\mathsf{com}}$ is chosen in random or sampled by the extendable output function [40]. Therefore, the distribution of $\mathbf{A}_{\mathsf{com}}$ and PP in the simulation is statistically indistinguishable with real attack. The matrices $\{\mathbf{A}^{(i)}\}_{i\in[N]}$ in the real scheme and the matrices $\{\mathbf{A}^{(i)}\}_{i\in[N]\setminus i^\diamond}$ in the simulation were generated by $\mathsf{SuperTrapGen}$ while the matrix $\mathbf{A}^{(i^\diamond)}$ is formed by m uniformly random and independent samples from $\mathbb{Z}_q^n$ from the SIS challenger. By Lemma 7, we know the $\{\mathbf{A}^{(i)}\}_{i\in[N]}$ in both real and simulated world have distribution that is statistically indistinguishable with real attack. For the matrices $\{\mathbf{B}^{(i)}\}_{i\in[N]}$, it were uniformly random selected in the real scheme, in the simulation, the matrices $\{\mathbf{B}^{(i)}\}_{i\in[N]\setminus i^\diamond)}$ were generated by $\mathsf{TrapGen}$ while the matrix $\mathbf{B}^{(i^\diamond)}$ was generated by $\mathsf{SuperTrapGen}$. By Lemma 6 and 7, we know the $\{\mathbf{B}^{(i)}\}_{i\in[N]}$ in both real and simulated world have distribution that is statistically indistinguishable with real attack. For the matrices $\{\mathbf{B}^{(i)}\}_{i\in[N]}$, both real and simulated world select that in uniformly random, so it is immediate. For the matrices $(\mathbf{A}_0^{(i)}, \mathbf{A}_1^{(i)})$, $\{\mathbf{B}_j^{(i)}\}_{j\in[k]}$, and $(\mathbf{C}_0^{(i)}, \mathbf{C}_1^{(i)})$ for all $i \in [N]$ generated in the simulation have distribution that is statistically indistinguishable with real attack by Lemma 5. Therefore, the set of verifications keys S given to $\mathcal{A}$ is statistically close to those in the real attack.

Claim 2. *The replies of the signing oracle* $\mathsf{OSign}(\cdot,\cdot,\cdot)$ *simulated by* $\mathcal{S}$ *is statistically close to those in the real attack when set* $\sigma = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$.

Proof. By Definition 4, in our parameters setting, the entries $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N')}$ in the signature tuples output from the oracle $\mathsf{OSign}(\cdot,\cdot,\cdot)$ are statistically close to those in the real attack. By the witness indistinguishability of the proof system, the proof π in the signature tuples output from the oracle $\mathsf{OSign}(\cdot,\cdot,\cdot)$ is statistically close to those in the real attack. For the $\mathsf{vk}_{\mathsf{OTS}}$, there is no change in the simulation and real attack. For the Σ_{OTS}, it is immediate since we directly invoke the one-time signature signing algorithm in both simulated and real world. Therefore, we focus on the entries $(\mathbf{e}_{\mathsf{chk}}, \mathbf{e}'_{\mathsf{chk}}, (\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_1^{(N')}))$.

By Lemma 10, for sufficient large Gaussian parameter σ, the distribution of the entries $(\mathbf{e}_{\mathsf{chk}}, \mathbf{e}'_{\mathsf{chk}}, (\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_1^{(N')}))$ generated by $\mathsf{SamplePre}$ are statistically close to the distribution of signatures generated in the real scheme. So we next analyze how to set the parameter σ. In the simulating signing oracle phase, we constructed $\mathbf{F}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d} = \left[\mathbf{A}^{(i\diamond)} | \mathbf{A}^{(i\diamond)}(\mathbf{R}^{(i\diamond)}_{1-d} - \mathbf{R}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}) + (1-2d)\mathbf{G}\right]$. Let $\bar{\mathbf{R}}^{(i\diamond)} = \mathbf{R}^{(i\diamond)}_{1-d} - \mathbf{R}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}$. By Lemma 1, we know $\left\|\widetilde{\bar{\mathbf{R}}^{(i\diamond)}}\right\| \leq O(\ell^{2c} \cdot m^{3/2})$ for some constant c. By Lemma 9, we know $\left\|\widetilde{\mathbf{T}_{\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}}}\right\| < (\left\|\bar{\mathbf{R}}^{(i\diamond)} + 1\right\|) \cdot \left\|\widetilde{\mathbf{T}_{\mathbf{G}}}\right\|$. By Lemma 12, we know $\|\widetilde{\mathbf{T}_{\mathbf{G}}}\| \leq \sqrt{5}$. By Lemma 10, it requires to set $\sigma > \left\|\widetilde{\mathbf{T}_{\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}}}\right\| \cdot \omega(\sqrt{\log m})$. Therefore, to satisfy these requirements, set $\sigma = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$ is sufficient.

Claim 3. $\mathcal{A}$ *can produce a valid* $\mathsf{SIS}_{q,n,m,\beta}$ *solution with overwhelming probability.*

Proof. We argue that $\mathbf{e}_0^{(i^\diamond)} + (\mathbf{R}_{\mathbf{A}_d^{(i\diamond)}} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}) \cdot \mathbf{e}_1^{(i^\diamond)}$ that $\mathcal{S}$ finally output in the simulation is a valid $\mathsf{SIS}_{q,n,m,\beta}$ solution in two steps. We first explain it is sufficiently short, note that $\mathbf{e}_0^{(i^\diamond)}$ and $\mathbf{e}_1^{(i^\diamond)}$ follow the distribution $\mathcal{D}_{\mathbb{Z}^m,\sigma}$. By Lemma 4, $\|\mathbf{e}_0^{(i^\diamond)}\|, \|\mathbf{e}_1^{(i^\diamond)}\| \leq \sigma\sqrt{m}$. By Lemma 1, $\left\|\mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}\right\| \leq O(\ell^{2c} \cdot m^{3/2})$. By Lemma 3, the norm of $\mathbf{R}_{\mathbf{A}_d^{(i\diamond)}}$ is bounded by $\sqrt{m}$. By Lemma 12, $\|\widetilde{\mathbf{T}_{\mathbf{G}}}\| \leq \sqrt{5}$. Therefore, it requires to set $\beta \geq O(\ell^{2c} \cdot m^{2/3}) \cdot \sigma\sqrt{m}$.

Then we prove $\mathbf{e}_0^{(i^\diamond)} + (\mathbf{R}_{\mathbf{A}_d^{(i\diamond)}} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}) \cdot \mathbf{e}_1^{(i^\diamond)}$ is non-zero with overwhelming probability. Suppose that the $\mathbf{e}_1^{(i^\diamond)} = \mathbf{0}$, then for a valid forgery we must have at least one $\mathbf{e}_0^{(i^\diamond)} \neq \mathbf{0}$ and thus $\mathbf{e}_0^{(i^\diamond)} + (\mathbf{R}_{\mathbf{A}_d^{(i\diamond)}} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}) \cdot \mathbf{e}_1^{(i^\diamond)}$ is non-zero. Suppose on the contrary, there exists one $\mathbf{e}_1^{(i^\diamond)} \neq \mathbf{0}$, then we need to argue

$(\mathbf{R}_{\mathbf{A}_d^{(i^\diamond)}} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}) \cdot \mathbf{e}_1^{(i^\diamond)}$ is non-zero with overwhelming probability. Due to we assume $\mathbf{e}_1^{(i^\diamond)} = (e_1, \ldots, e_m) \neq \mathbf{0}$ which means at least one coordinate of $\mathbf{e}_1^{(i^\diamond)}$, denote as e_o where $o \in [m]$, such that $e_o \neq 0$. Let $\bar{\mathbf{R}} = (\mathbf{R}_{\mathbf{A}_d^{(i^\diamond)}} - \mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*})$ and write $\bar{\mathbf{R}} = (\bar{\mathbf{r}}_1, \ldots, \bar{\mathbf{r}}_m)$ and so $\bar{\mathbf{R}} \cdot \mathbf{e}_1^{(i^\diamond)} = \bar{\mathbf{r}}_o e_o + \sum_{\bar{o} \in [m] \setminus o} \bar{\mathbf{r}}_{\bar{o}} e_{\bar{o}}$. Note that for the fixed message μ^* on which $\mathcal{A}$ made the forgery, $\bar{\mathbf{R}}$ (therefore $\bar{\mathbf{r}}_o$) depends on the low-norm matrices $(\mathbf{R}_{\mathbf{A}_0^{(i^\diamond)}}, \mathbf{R}_{\mathbf{A}_1^{(i^\diamond)}})$, $\{\mathbf{R}_{\mathbf{B}_j^{(i^\diamond)}}\}_{j \in [k]}$, $(\mathbf{R}_{\mathbf{C}_0^{(i^\diamond)}}, \mathbf{R}_{\mathbf{C}_1^{(i^\diamond)}})$ and PRF key $\mathbf{k}$. The information about e_o for $\mathcal{A}$ is from the public matrices in the verification set S that given to the $\mathcal{A}$, and note that the PRF keys $\mathbf{k}$ which is not included in S. Therefore, by the pigeonhole principle there is an exponentially large freedom to pick a value to $\bar{\mathbf{r}}_o$ which is compatible with $\mathcal{A}$'s view. This completes the proof.

Theorem 2 (Anonymity). *Set the parameters as Sect. 4.3, the LRS scheme is signer-anonymous in the standard model.*

Proof. The proof proceeds in a sequence of experiments E_0, H_0, H_1, E_1 such that E_0 (resp., E_1) corresponds to the experiment of $\mathsf{Anonymity}$ in Definition 1 with $b = 0$ (resp., $b = 1$), and such that each experiment is indistinguishable from the one before it. This implies that $\mathcal{A}$ has negligible advantage in distinguishing E_0 from E_1, as desired.

E_0 :This experiment first generate $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$, and $\{\mathsf{vk}^{(i)}, \mathsf{sk}^{(i)}\}_{i \in [N]}$ by repeatedly invoking $\mathsf{KeyGen}()$, and $\mathcal{A}$ is given $(\mathsf{PP}, \mathsf{S} = \{\mathsf{vk}^{(i)}\}_{i \in [N]})$ and the randomness γ_{st}. Then $\mathcal{A}$ provides a challenge $(\boldsymbol{\mu}^*, \mathsf{R}^*, s_0^*, s_1^*)$ to the challenger, and requires that $\boldsymbol{\mu}^* \in \mathcal{M}$, $s_0^* \neq s_1^*$ and $\mathsf{vk}^{(s_0^*)}, \mathsf{vk}^{(s_1^*)} \in \mathsf{S} \bigcap \mathsf{R}^*$. For each tuple $(\boldsymbol{\mu}^*, \mathsf{R}^*, s_0^*, s_1^*)$ in the challenge, the experiment uses $\mathsf{sk}^{(s_0^*)}$ to compute the signature tuple Σ^* and responses to $\mathcal{A}$.

H_0 :This experiment is as same as experiment E_0 except that we change how the signature Σ^* is generated: we sample $\mathbf{e}_0^{(s_1^*)}$ by $\mathsf{SamplePre}$ rather than randomly select it from $\mathbb{Z}_q^m$.

Then we show that E_0 and H_0 are indistinguishable for $\mathcal{A}$, which we do by giving a reduction from the hardness assumption $\mathsf{LWE}_{m,q,\alpha q\sqrt{2}}$.

Reduction. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing E_0 and H_0. We use $\mathcal{A}$ to construct an algorithm $\mathcal{S}$ for breaking the hardness assumption $\mathsf{LWE}_{m,q,\alpha q\sqrt{2}}$. $\mathcal{S}$ is given as input $(\mathbf{B}, \mathbf{z}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$, where $\mathbf{B}$ is uniform and $\mathbf{z}$ is either uniform or equal to $\mathbf{B}^\top \mathbf{s} + \mathbf{e}$ for $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \alpha q\sqrt{2}}$.

Setup Phase. $\mathcal{S}$ takes as input a security parameter n and a randomness γ to invoke $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$ algorithm. $\mathcal{S}$ simulates as follows.

- Choose a random index $\bar{i}^* \xleftarrow{\$} \{1, \ldots, N\}$, sets $\mathbf{B}^{(\bar{i}^*)} = \mathbf{B}$.
- For $i = \bar{i}^*+1, \ldots, N, 1, \ldots, \bar{i}^*-1$, compute $(\mathbf{B}^{(i)}, \mathbf{T}_{\mathbf{B}^{(i)}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.

- For $i = 1$ to N, compute $(\mathbf{A}^{(i)}, \mathbf{T}_{\mathbf{A}^{(i)}}) \leftarrow \mathsf{SuperTrapGen}(1^n, 1^m, q, \mathbf{B}^{(i)})$. Set $\mathsf{vk}_{\mathsf{OTS}}^{(i)} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(i)}}$ and $\mathsf{sk}_{\mathsf{OTS}}^{(i)} = \mathbf{T}_{\mathbf{A}^{(i)}}$.
- For $i = 1$ to N and $d \in \{0, 1\}$, select $\mathbf{A}_d^{(i)}, \mathbf{C}_d^{(i)} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.
- For $i = 1$ to N, select a PRF key $\mathbf{k}^{(i)} \xleftarrow{\$} \{0, 1\}^k$.
- For $j = 1$ to k, select $\mathbf{B}_j^{(i)} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.
- Set $\mathsf{S} = \{\mathsf{vk}^{(i)}\}_{i \in [N]}$, $\mathsf{vk}^{(i)} = (\mathbf{A}^{(i)}, (\mathbf{A}_0^{(i)}, \mathbf{A}_1^{(i)}), \mathbf{B}^{(i)}, \{\mathbf{B}_j^{(i)}\}_{j \in [k]}, (\mathbf{C}_0^{(i)}, \mathbf{C}_1^{(i)}))$, then sends $(\mathsf{PP}, \mathsf{S}, \gamma_{\mathsf{st}})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ provides a challenge $(\boldsymbol{\mu}^*, \mathsf{R}^*, s_0^*, s_1^*)$ to the challenger. $\mathcal{S}$ chooses a random bit $b \in \{0, 1\}$ and fixes it throughout the response phase for the challenge. For each tuple $(\boldsymbol{\mu}^*, \mathsf{R}^*, s_0^*, s_1^*)$ in the challenge, $\mathcal{S}$ does as following:

- Let $N^* = |\mathsf{R}^*|$. Check if $s_0^* \neq s_1^*$, $\mathsf{vk}^{(s_0^*)}, \mathsf{vk}^{(s_1^*)} \in \mathsf{S} \bigcap \mathsf{R}^*$ and $\bar{i}^* = s_1^*$, otherwise $\mathcal{S}$ aborts the simulation.
- Compute $d = \mathsf{PRF}(\mathbf{k}^{(s_1^*)}, \boldsymbol{\mu}^*)$.
- Compute $\mathbf{F}_{\mathsf{chk}}$, $\mathbf{F}'_{\mathsf{chk}}$, $\mathbf{e}_{\mathsf{chk}}$, and $\mathbf{e}'_{\mathsf{chk}}$ as in Sign algorithm.
- For $i^* = s_0^*$, select $\mathbf{e}_1^{(s_0^*)} \xleftarrow{\$} \mathbb{Z}_q^m$ and computes $\mathbf{e}_0^{(s_0^*)}$ by $\mathsf{SamplePre}$ such that $\mathbf{F}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d}^{(s_0^*)} \cdot (\mathbf{e}_0^{(s_0^*)}; \mathbf{e}_1^{(s_0^*)}) = \mathbf{0} \pmod q$ holds as in Sign algorithm.
- For $i^* = s_1^*$, let $\mathbf{z}^{(i^*)} = \mathbf{z}$, uniformly choose $\mathbf{e}_1^{(s_1^*)} \in \mathbb{Z}_q^m$ such that $\mathbf{F}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d}^{(s_1^*)} \cdot (\mathbf{z}; \mathbf{e}_1^{(i_1^*)}) = \mathbf{0} \pmod q$ holds.
- For all $i^* \in [N^*]$ and $i^* \neq s_0^*, s_1^*$, select $\mathbf{e}_1^{(i^*)} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sigma}$ and compute $\mathbf{e}_0^{(i^*)} \in \mathbb{Z}_q^m$ uniformly subject to the condition that $\mathbf{F}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d}^{(i^*)} \cdot (\mathbf{e}_0^{(i^*)}; \mathbf{e}_1^{(i^*)}) = \mathbf{0}$ $\pmod q$ holds as in Sign algorithm.
- For $i^* = s_1^* + 1, \ldots, N^*, 1, \ldots, s_1^* - 1$, compute the ciphertext $\mathbf{z}^{(i^*)}$ as in E_0 and H_0. Then construct an NIWI proof π for the gap language $L_{\sigma, \varepsilon}$ as in Sign algorithm.
- Compute one-time signature $\Sigma_{\mathsf{OTS}} \leftarrow \Pi_{\mathsf{OTS}}.\mathsf{Sign}(\mathsf{sk}_{\mathsf{OTS}}, \boldsymbol{\mu}^*)$.
- Return the signature $\Sigma = (\Sigma_{\mathsf{OTS}}, \mathsf{vk}_{\mathsf{OTS}}, \mathbf{e}_{\mathsf{chk}}, \mathbf{e}'_{\mathsf{chk}}, \{\mathbf{e}_1^{(i^*)}, \mathbf{z}^{(i^*)}\}_{i^* \in [N^*]}, \pi)$ to $\mathcal{A}$.

Guess. When $\mathcal{A}$ outputs the guess b', $\mathcal{S}$ outputs the guess b'.

Let $\mathcal{D}_{\$}$ denote the above experiment when $\mathcal{S}$'s input $\mathbf{z}$ is uniformly distributed. Let $\mathcal{D}_{\mathsf{LWE}}$ denote the above experiment when $\mathcal{S}$'s input $\mathbf{z}$ is distributed according to $\mathbf{y} = \mathbf{B}^\top \mathbf{s} + \mathbf{e}$ for $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \alpha q\sqrt{2}}$.

Claim 4. *$\mathcal{A}$'s view in $\mathcal{D}_{\$}$ is statistically close to its view in E_0.*

Proof. In experiment E_0, we have $\mathbf{z}^{(s_1^*)} = (\mathbf{B}^{(s_1^*)})^\top \mathbf{s}^{(s_1^*)} + \mathbf{e}_0^{(s_1^*)}$ where $\mathbf{e}_0^{(s_1^*)}$ is chosen uniformly subject to $\mathbf{F}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d}^{(s_1^*)} \cdot (\mathbf{e}_0^{(s_1^*)}; \mathbf{e}_1^{(s_1^*)}) = \mathbf{0} \pmod q$ and $\mathbf{e}_1^{(s_1^*)} \xleftarrow{\$} \mathbb{Z}_q^m$. In $\mathcal{D}_{\$}$, we let $\mathbf{z}^{(s_1^*)} = \mathbf{z}$ and recall that $\mathbf{z} = \mathbf{B}^\top \mathbf{s} + \mathbf{e}$ for $\mathbf{e} \in \mathbb{Z}_q^m$ is uniformly selected. And $\mathbf{e}_1^{(s_1^*)}$ is chosen uniformly subject to $\mathbf{F}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d}^{(s_1^*)} \cdot (\mathbf{z}; \mathbf{e}_1^{(s_1^*)}) = \mathbf{0}$

(mod q). Recall $\mathbf{F}^{(s_1^*)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*,1-d} = \mathbf{A}^{(s_1^*)}\mathbf{e}_0^{(s_1^*)} + (\mathbf{A}^{(s_1^*)}_{1-d} - \mathbf{A}^{(s_1^*)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}) \cdot \mathbf{e}_1^{(s_1^*)} = \mathbf{0} \pmod q$. We can view $\mathbf{A}^{(s_1^*)}$ and $(\mathbf{A}^{(s_1^*)}_{1-d} - \mathbf{A}^{(s_1^*)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}})$ as regular function $\mathbb{Z}_q^m \to \mathbb{Z}_q^n$. By Lemma 5, the randomly chosen $\mathbf{e}_0^{(s_1^*)}$ is uniform over the images of $\mathbf{A}^{(s_1^*)}$. For a regular function, choosing a uniform element from the images, followed by a uniform element from its pre-images, is equivalent to choosing a uniform element from the domain, as is done in $\mathcal{D}_\$$. Therefore the choice of $\mathbf{e}_0^{(s_1^*)}$ in E_0 is statistically close to uniform over $\mathbb{Z}_q^m$, and hence $\mathbf{z}^{(s_1^*)}$ is statistically indistinguishable between E_0 and $\mathcal{D}_\$$. Similarly, this proof also can show the $\mathbf{e}_1^{(s_1^*)}$ in $\mathcal{D}_\$$ statistically close to uniform over $\mathbb{Z}_q^m$.

Claim 5. *$\mathcal{A}$'s view in $\mathcal{D}_{\mathsf{LWE}}$ is statistically close to its view in H_0.*

Proof. In experiment H_0, $\mathbf{z}^{(s_1^*)} = (\mathbf{B}^{(s_1^*)})^\top \mathbf{s}^{(s_1^*)} + \mathbf{e}_0^{(s_1^*)}$ where $\mathbf{e}_0^{(s_1^*)}$ is sampled by $\mathsf{SamplePre}$ algorithm. In $\mathcal{D}_{\mathsf{LWE}}$, we let $\mathbf{z}^{(s_1^*)} = \mathbf{z}$ and recall that $\mathbf{z} = \mathbf{B}^\top \mathbf{s} + \mathbf{e}$ for $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q\sqrt{2}}$. The proof to show $\mathbf{e}_1^{(s_1^*)}$ in H_0 and $\mathcal{D}_{\mathsf{LWE}}$ indistinguishable is as same as the last claim. Under the setting of the parameters given in Sect. 4.3, and by Lemma 10, $\mathbf{z}^{(s_1^*)}$ is indistinguishable between H_0 and $\mathcal{D}_\$$.

H_1 : This experiment is the same as experiment E_1 except that the proof π is now computed using the witness $\{\mathbf{s}^{(i^*)}, i^*\}^{(s_1^*)}_{i^* \in [N^*]}$ rather than $\{\mathbf{s}^{(i^*)}, i^*\}^{(s_0^*)}_{i^* \in [N^*]}$.

The rest of the proof is straightforward. H_1 is indistinguishable from E_1 by exactly the same argument used to show the indistinguishability of H_0 and E_0. By the witness indistinguishability of the proof system, H_0 and H_1 are indistinguishable. This completes the proof.

Theorem 3 (Linkability). *Set the parameters as Sect. 4.3, the LRS scheme is signer-linkable in the standard model.*

Proof. **Setup Phase**. $\mathcal{S}$ takes as input a security parameter n and a randomness γ_{st} to invoke $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n; \gamma_{\mathsf{st}})$ algorithm, then send $(\mathsf{PP}, \gamma_{\mathsf{st}})$ to $\mathcal{A}$.

Output Phase. $\mathcal{A}$ outputs l ($l \geq 2$) (messages, ring of verification keys, signature) tuples $(\mathsf{R}_i^*, \boldsymbol{\mu}_i^*, \Sigma_i^*)$.

Infer that there must existing a ring member in the union set $\cup_{i=1}\mathsf{R}_i^*$ who generated at least two signature tuples. In other words, this ring member, assuming his index is s, had produced two valid one-time verification keys $(\mathsf{vk}_{\mathsf{OTS}}, \mathsf{vk}^*_{\mathsf{OTS}})$. Let $\mathsf{vk}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}}$ be the honest one-time verification key. Now we analyze $\mathcal{A}$ how to produce the $\mathsf{vk}^*_{\mathsf{OTS}}$. There are two ways:

- The first way is $\mathcal{A}$ produces a $\mathbf{A}^*_{\mathsf{com}} \neq \mathbf{A}_{\mathsf{com}}$. In this case, we have $\mathbf{F}'_{\mathsf{chk}} = \left[\mathbf{A}^*_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} - \mathbf{A}_{\mathsf{com}} | \mathbf{A}_{\mathsf{com}}\right]$. Recall the $\mathsf{BasisExtBindSK}$ algorithm, $\mathcal{A}$ needs to compute a low-norm basis $\mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}} = \left[\begin{smallmatrix} -\mathbf{R}_0 & -\mathbf{R}_1 \\ \mathbf{T}_{\mathbf{A}^{(s)}} - \mathbf{R}_0 & \mathbf{T}_{\mathbf{A}^{(s)}} - \mathbf{R}_1 \end{smallmatrix}\right]$ such that $\mathbf{F}'_{\mathsf{chk}} \cdot \mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}} = \mathbf{0} \pmod q$. It holds that $\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} - \mathbf{A}^*_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R}_0 = \mathbf{0} \pmod q$ and $\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} - \mathbf{A}^*_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R}_1 = \mathbf{0} \pmod q$. Then we have $\mathbf{A}^*_{\mathsf{com}}(\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R}_0 - \mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R}_1) = \mathbf{0} \pmod q$ holds. As the parameters set in Sect. 4.3, $(\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R}_0 - \mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R}_1)$ will be a valid SIS solution.

- The second way is $\mathcal{A}$ produces a $\mathbf{T}_{\mathbf{A}^*} \neq \mathbf{T}_{\mathbf{A}^{(s)}}$. Let $N^* = |\cup_{i=1} \mathsf{R}_i^*|$. In this case, existing an index $s' \in [N^*]$ satisfy that, $\mathbf{F}^{(s')} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^*}|\mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(s')}\right]$, and $\mathbf{F}^{(s')}$ has the basis $\mathbf{T}_{\mathbf{F}^{(s')}} = \begin{bmatrix} -\mathbf{R}_0 & -\mathbf{R}_1 \\ \mathbf{T}_{\mathbf{A}^*}\mathbf{R}_0 & \mathbf{T}_{\mathbf{A}^*}\mathbf{R}_1 \end{bmatrix}$ by the $\mathsf{BasisExtBindAcom}$ algorithm. It holds that $\mathbf{A}^{(s')}\mathbf{T}_{\mathbf{A}^*}\mathbf{R}_0 = \mathbf{0} \pmod q$ and $\mathbf{A}^{(s')}\mathbf{T}_{\mathbf{A}^*}\mathbf{R}_1 = \mathbf{0} \pmod q$. Then we have $\mathbf{A}^{(s')}(\mathbf{T}_{\mathbf{A}^*}\mathbf{R}_0 - \mathbf{T}_{\mathbf{A}^*}\mathbf{R}_1) = \mathbf{0} \pmod q$ holds. As the parameters set in Sect. 4.3, $(\mathbf{T}_{\mathbf{A}^*}\mathbf{R}_0 - \mathbf{T}_{\mathbf{A}^*}\mathbf{R}_1)$ will be a valid SIS solution.

This completes the proof.

Theorem 4 (Non-Slanderability). *Set the parameters as Sect. 4.3, the LRS scheme is signer-non-slanderable in the standard model.*

Proof. **Setup**. As same as the Setup phase of unforgeability proof.

Probing. As same as the Probing phase of unforgeability proof.

Output. $\mathcal{A}$ outputs two signature tuples $(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*)$ and $(\hat{\boldsymbol{\mu}}, \hat{\mathsf{R}}, \hat{\Sigma})$. Let $N^* = |\mathsf{R}^*|$. Check if $\mathsf{Ver}(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*) = 1$ and $(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*) \notin \mathsf{L}$ and $(\hat{\boldsymbol{\mu}}, \hat{\mathsf{R}}, \hat{\Sigma}) \in \mathsf{L}$ and $\mathsf{R}^* \subseteq \mathsf{S} \setminus \mathsf{C}$ and the proof π^* is correct and $\mathsf{Link}(\mathsf{R}^*, \boldsymbol{\mu}^*, \Sigma^*, \hat{\mathsf{R}}, \hat{\boldsymbol{\mu}}, \hat{\Sigma}) = 1$ i.e., $\mathsf{vk}^*_{\mathsf{OTS}} = \hat{\mathsf{vk}}_{\mathsf{OTS}}$, otherwise aborts. Let $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}^*_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}}$ and $\hat{\mathsf{vk}}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}$. We analyze $\mathcal{A}$ how to produce $\mathsf{vk}^*_{\mathsf{OTS}}$ and make $\mathsf{vk}^*_{\mathsf{OTS}} = \hat{\mathsf{vk}}_{\mathsf{OTS}}$ holds. There are two ways:

- The first way is $\mathcal{A}$ selects a basis $\mathbf{T}^*_{\mathbf{A}}$ and then computes the $\mathbf{A}^*_{\mathsf{com}}$ such that $\mathbf{A}^*_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}} = \mathbf{A}_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}$ holds. We have $\mathbf{F}'_{\mathsf{chk}} = \left[\mathbf{A}^*_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}} - \mathbf{A}_{\mathsf{com}}|\mathbf{A}_{\mathsf{com}}\right]$. By the $\mathsf{BasisExtBindSK}$ algorithm, $\mathcal{A}$ needs to generate a low-norm basis $\mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}} = \begin{bmatrix} -\mathbf{R}_0 & -\mathbf{R}_1 \\ \mathbf{T}^*_{\mathbf{A}}-\mathbf{R}_0 & \mathbf{T}^*_{\mathbf{A}}-\mathbf{R}_1 \end{bmatrix}$ such that $\mathbf{F}'_{\mathsf{chk}} \cdot \mathbf{T}_{\mathbf{F}'_{\mathsf{chk}}} = \mathbf{0} \pmod q$. It holds that $\mathbf{A}_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}} - \mathbf{A}^*_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_0 = \mathbf{0} \pmod q$ and $\mathbf{A}_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}} - \mathbf{A}^*_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_1 = \mathbf{0} \pmod q$. Then we have $\mathbf{A}^*_{\mathsf{com}}(\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_0 - \mathbf{T}^*_{\mathbf{A}}\mathbf{R}_1) = \mathbf{0} \pmod q$ holds. As the parameters set in Sect. 4.3, $(\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_0 - \mathbf{T}^*_{\mathbf{A}}\mathbf{R}_1)$ will be a valid SIS solution.
- The second way is $\mathcal{A}$ corrupts the $\hat{\mathbf{T}}_{\mathbf{A}}$ and then computes a $\mathbf{A}^*_{\mathsf{com}}$ such that $\mathbf{A}^*_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}} = \mathbf{A}_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}$ holds. In this case, existing an index $s \in [N^*]$ satisfy that, $\mathbf{F}^{(s)} = \left[\mathbf{A}_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}|\mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(s)}\right]$, and $\mathbf{F}^{(s)}$ has the basis $\mathbf{T}_{\mathbf{F}^{(s)}} = \begin{bmatrix} -\mathbf{R}_0 & -\mathbf{R}_1 \\ \hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_0 & \hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_1 \end{bmatrix}$ by the $\mathsf{BasisExtBindAcom}$ algorithm. It holds that $\mathbf{A}_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_0 - \mathbf{A}^*_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_0 = \mathbf{0} \pmod q$ and $\mathbf{A}_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_1 - \mathbf{A}^*_{\mathsf{com}}\hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_1 = \mathbf{0} \pmod q$. Then we have $\mathbf{A}^*_{\mathsf{com}}(\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_0 - \hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_0) = \mathbf{0} \pmod q$ and $\mathbf{A}^*_{\mathsf{com}}(\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_1 - \hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_1) = \mathbf{0} \pmod q$ holds. As the parameters set in Sect. 4.3, $(\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_0 - \hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_0)$ and $(\mathbf{T}^*_{\mathbf{A}}\mathbf{R}_1 - \hat{\mathbf{T}}_{\mathbf{A}}\mathbf{R}_1)$ will be valid SIS solutions.

This completes the proof.

References

1. Au, M.H., Chow, S.S.M., Susilo, W., Tsang, P.P.: Short linkable ring signatures revisited. EuroPKI 2006, 101115 (2006). https://doi.org/10.1007/11774716_9
2. Au, M.H., Liu, J.K., Susilo, W., Yuen, T.H.: Constant-size id-based linkable and revocable-iff-linked ring signature. INDOCRYPT 2006, 364378 (2006). https://doi.org/10.1007/11941378_26

3. Au, M.H., Liu, J.K., Susilo,W., Yuen, T.H.:Secureid-based linkable and revocable-iff-linked ring signature with constant-size construction. Theor. Comput. Sci. 469, 114 (2013). https://doi.org/10.1016/j.tcs.2012.10.031
4. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28
5. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_6
6. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory of Computing Systems, **48**(3), 535–553, https://doi.org/10.1007/s00224-010-9278-3 (2009)
7. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: Logarithmic-size, no setup–from standard assumptions. In: Ishai Y., Rijmen V. (ed.) EUROCRYPT 2019. LNCS, vol. 11478. pp. 281-311. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-17659-4_10
8. Boneh, D., Dagdelen, O., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, H,D., Wang, X.Y. (ed.) ASIACRYPT 2011. LNCS, vol. 7073. pp. 41-69. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_3
9. Boyen, X., Haines, T.: Forward-secure linkable ring signatures from bilinear maps. Cryptography 2(4), 35 (2018). https://doi.org/10.3390/cryptography2040035
10. Bender, A., Jonathan Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. In: Halevi S., Rabin T. (ed.) TCC 2006. LNCS, vol. 11478. pp. 60-79. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_4
11. Baum, C., Lin, H., Oechsner, S.: Towards practical lattice-based one-time linkable ring signatures. ICICS 2018, 303322 (2018). https://doi.org/10.1007/978-3-030-01950-1_18
12. Branco, P., Mateus, P.: A code-based linkable ring signature scheme.ProvSec2018, 203219 (2018). https://doi.org/10.1007/978-3-030-01446-9_12
13. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key- homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In: Nguyen P.Q., Oswald E. (ed.) EUROCRYPT 2014. LNCS, vol. 11478. pp. 281-311. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30
14. Boyen, X., Li, Q.Y.: Towards tightly secure lattice short signature and id-based encryption. In: Cheon J., Takagi T. (ed.) ASIACRYPT 2016. LNCS, vol. 11478. pp. 404-434. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_14
15. Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices. In: Moriai, S., Wang, H.X. (ed.) ASIACRYPT 2020. LNCS, vol. 12492. pp. 464–492. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-64834-3_16
16. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 6273. ACM Press, New York (1993) https://doi.org/10.1145/168588.168596
17. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: Noar, M. (ed.) ITCS 2014. LNCS, vol. 11478. pp. 1-12. Springer, Heidelberg (2014). https://doi.org/10.1145/2554797.2554799
18. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. Journal. ACM, pp. 557–594. ACM Press. (2004). https://doi.org/10.1145/1008731.1008734

19. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110. pp. 523-552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
20. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 449466. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_27
21. Eaton, E., Song, F.: A Note on the Instantiability of the Quantum Random Oracle. In: Ding, J.T., Tillich, J (ed.) PQC 2020. LNCS, vol. 12100. pp. 503-523. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-44223-1_27
22. Fujisaki, E.: Sub-linear size traceable ring signatures without random oracles. CT-RSA 2011, 393415 (2011). https://doi.org/10.1007/978-3-642-19074-225
23. Fujisaki, E., Suzuki, K.: Traceable ring signature. PKC 2007, 181200 (2007). https://doi.org/10.1007/978-3-540-71677-8_13
24. Grilo, A., Hövelmanns, K., Hülsing, A., Majenz, C.: Tight adaptive reprogramming in the QROM. IACR Cryptol. ePrint Archive 2020, 1361 (2020). https://eprint.iacr.org/2020/1361
25. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ishai Y., Rijmen V. (ed.) STOC 2019. LNCS, vol. 11478. pp. 197-206. ACM (2008). https://doi.org/10.1145/1374376.1374407
26. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A (ed.) CRYPTO 2013. LNCS, vol. 8042. pp. 75-92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5
27. Gordon, S.D., C., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477. pp. 395-412. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_23
28. Katz, J.: Digital Signatures. Springer, Heidelberg (2010). https://doi.org/10.1007/978-0-387-27712-7
29. Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Linkable ring signature with unconditional anonymity. IEEE Trans. Knowl. Data Eng. 26(1), 157165 (2014). https://doi.org/10.1109/TKDE.2013.17
30. Lu, X., Au, M.H., Zhang, Z.: Raptor: a practical lattice-based (linkable) ring signature. In: Deng, R.H (ed.) ACNS 2019. LNCS, vol. 11464. pp. 110-130. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-21568-2_6
31. Lai, Q.Q., Liu, F.H., Wang, Z.D.: Almost tight security in lattices with polynomial moduliPRF, IBE, all-but-many LTF, and more. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V (ed.) PKC 2020. LNCS, vol. 12110. pp. 652-681. Springer, Edinburgh (2020). https://doi.org/10.1007/978-3-030-45374-9_22
32. Lyubashevsky, V., Micciancio, D: Asymptotically Efficient Lattice-Based Digital Signatures. J. Cryptol. 31(3), 774-797 (2018). https://doi.org/10.1007/s00145-017-9270-z
33. Leurent, G., Nguyen, P.Q.: How risky is the random-oracle model?. In: Halevi, S (ed.) CRYPTO 2009. LNCS, vol. 5677. pp. 445–464. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-030-29959-0_35
34. Liu, Z., Nguyen, K., Yang, G.M., Wang, H.X., Wong, D.S.: A Lattice-Based Linkable Ring Signature Supporting Stealth Addresses. In: Sako, K., Schneider, S.A., Ryan, P (ed.) ESORICS 2019. LNCS, vol. 11735. pp. 726–746. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-29959-0_35
35. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Ishai Y., Rijmen V. (ed.) ACISP 2004, 325–335 (2004). https://doi.org/10.1007/978-3-540-27800-928
36. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a cryptographic perspective. vol 671. Kluwer Academic Publishers, 2002. https://dblp.org/rec/books/daglib/0018102.bib

37. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D. (ed.) EUROCRYPT 2012. LNCS, vol. 7237. pp. 700-718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
38. Micciancio, D., Regev., O.: Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing, **37** (1), 267–302. https://doi.org/10.1137/s0097539705447360
39. Micciancio, D., Vadhan. P, S.: Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729. pp. 282–298. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_17
40. NIST. SHA-3 standard: Permutation-based hash and extendable-output functions.Technical report, 2015. Available at https://doi.org/10.6028/NIST.FIPS.202.
41. Peikert, C., Shiehian, S.: Non-interactive zero knowledge for NP from (plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 89114. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_4
42. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: 37th ACM STOC, pp. 8493 (2005). https://doi.org/10.1145/1060590.1060603
43. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C (ed.) ASIACRYPT 2001. LNCS, vol. 2248. pp. 552565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
44. Sun, S., Au, M.H., Liu, J.K., Yuen, T.H.: Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In: ESORICS 2017 Part II. pp. 456474 (2017). https://doi.org/10.1007/978-3-319-66399-9_25
45. Sokolov.A.A.: Lin2-Xor Lemma and Log-size Linkable Ring Signature. IACR Cryptol. ePrint Archive 2020, 688 (2020). https://eprint.iacr.org/2020/688
46. Torres, W.A., Steinfeld, R., Sakzad, A., Liu, J.K., Kuchta, V., Bhattacharjee, N., Au, M.H., Cheng, J.: Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1. 0). In: Susilo, W., Yang, G (ed.) Information Security and Privacy 2018. LNCS, vol. 10946. pp. 558576. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-319-93638-3_32
47. Torres, W.A., Steinfeld, R., Sakzad, A., Liu, J.K., Kuchta, V.: Post-quantum linkable ring signature enabling distributed authorised ring confidential transactions in blockchain. IACR Cryptol. ePrint Archive 2020, 1121 (2020). https://eprint.iacr.org/2020/1121
48. Torres,W.A., Kuchta, V., Steinfeld, R., Sakzad, A., Liu, J.K., Cheng, J:. Lattice ringct v2. 0 with multiple input and multiple output wallets. In: Jaccard, J.J (ed.) in Information Security and Privacy 2019. LNCS, vol. 11547. pp. 156175. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-21548-4_9
49. Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. ISPEC 2005, 4860 (2005). https://doi.org/10.1007/978-3-540-31979-55
50. Tsang, P.P., Wei, V.K., Chan, T.K., Au, M.H., Liu, J.K., Wong, D.S.: Separable linkable threshold ring signatures. INDOCRYPT 2004, 384398 (2004). https://doi.org/10.1007/978-3-540-30556-9_30
51. Yuen, T.H., Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Efficient linkable and/or threshold ring signature without random oracles. Comput. J. 56(4), 407421 (2013). https://doi.org/10.1093/comjnl/bxs115
52. Zhang, H., Zhang, F., Tian, H., Au, M.H.: Anonymous post-quantum cryptocash. In: Meiklejohn, S., Sako, K (ed.) Financial Cryptography and Data Security 2018. LNCS, vol. 10957. pp. 461–479. Springer, Heidelberg (2018). https://doi.org/10.1007/3-540-36178-2_33