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 property 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 in the standard model. Toward our goal, we present a lattice basis extending algorithm which is the key ingredient in our construction, that may be of independent interest.

Keywords: Lattice-Based cryptography · Linkable ring signature · Standard model.

1 Introduction

Ring signatures, introduced by Rivest et al. [39], 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-anonymity 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 can 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) [28] 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 [46,3,9,42,41] especially in recent years, driven by the rapid development of cryptocurrencies.

Another important line of research is constructing LRS schemes from *lattices* [4,6,30,29,45,44,43], since lattice-based cryptography has attracted more attention due to its distinctive features, especially the quantum-resistant. However, *these works have so far required the random oracle (ROM) model [5] (or similar heuristics) for their security analysis.* Katz (Sect. 6.2.1 of [25]) mentioned that existing some negative results about the cryptographic systems that rely on ROM. Canetti et al. [15] and Dodis et al. [19] showed that proof in ROM can only serve as a heuristic argument, it may lead to insecure schemes when the ROM is implemented in practical scenarios. Leurent and Nguyen [27] presented the attacks extracting the secret keys on several hash-then-sign type signature schemes (including the lattice-based signature [21]) and identity-based encryption schemes if the underlying hash functions are modeled as a random oracle. Quantum Random Oracle Model (QROM) is a generalized notion of ROM [7]. Though the proof of security in QROM is stronger than one in the ROM, it does not mean the security in the QROM implies standard-model security [20]. And Grilo et al. [24] showed that the proofs in QROM lack conceptual simplicity and tightness. Moreover, in some cryptosystems with advanced functionality, the adverse effect caused by employing (Q)ROM or related heuristics will be enlarged and unexpected. For instance, Chatterjee et al. [18] formalized the security models for ring signatures in quantum setting which tries to capture adversaries with quantum access to the signer, but as pointed by [17] it performs in contrast to ordinary signatures, since which is unclear if their models are as strong as the standard security notion when restricted to the classical world. And recently, Branco [13] present a novel ring signature in standard model, they explained why their work cannot rely on ROM, and introduced the ramifications of relying on ROM in their construction. Consequently, we can conclude that constructing cryptosystems in standard model is more reassuring, especially for the cryptographic primitives with advanced functionality such as linkable ring signatures.

1.1 Our Contribution

In this work, our main contribution is to present a lattice-based linkable ring signature (LRS) scheme in the standard model (i.e. without resorting to random oracles or common reference strings). Toward this goal, we present a new lattice basis extending algorithm which is the key ingredient in our construction and is instrumental in the security proofs. By arming with appropriate lattice techniques, we achieve our work without undermining the *compactness*. In particular, our security models (*Unforgeability*, *Anonymity*, *Linkability*, and *Non-Slanderability*) provide strong guarantees to capture the security requirements that practical scenarios imposed on LRS. At the same time, our work is asymptotically efficient on signature size since it grows only linearly in the ring size. In other words, our construction provides strong confidence on security in threefold: provably secure without relying on random oracle model or any random oracle heuristics, and instantiated from the well-studied *standard lattice assumptions* (SIS and LWE) make our work being quantum-resistant, and satisfies strong security notions without compensate the compactness. Supporting by this confidence on security, our work is more reassuring to confront the underlying challenges in practice. However, as for the majority of SIS/LWE-based cryptographic constructions in the standard model, the public key and signature sizes of our construction are still large for practical implementation. We do not want to oversell our results, but take this as a stepping stone towards the goal of practical LRS in standard model, as this is the first lattice-based LRS scheme in the standard model. Therefore, it is certainly an important direction for future research to improve the efficiency of our work to enable deployment and implementation in practice.

1.2 Our Methods

In this section, we give the methods with respect to the construction and proofs. It is instructive since our construction and proofs involved varied techniques and primitives.

Compact Construction Without Relying Random Oracle. We first explain how to construct a compact LRS without relying on random oracle. In LRS, the random oracle functions always are employed to hash the messages and to build the *key image*[1]. A general method to remove the random oracle in hashing the message is matching the message bit by public parameters, as prior works [36,11,16]. But this method severely undermines the compactness since it makes the scheme cumbersome and the size of public keys and signatures enlarged quickly (that is *quadratical* in works [36,11]) with the number of the ring member. In our construction, we employ the key-homomorphic evaluation algorithm from [22,12,8] to process the message, which is inspired by the standard signature scheme [10]. More specifically, the key-homomorphic evaluation algorithm $\mathsf{Eval}(\cdot,\cdot)$ takes as input a set of matrices and a fan-in-2 Boolean NAND circuit C which is expressed as a PRF function in our setting. When a user generates the key pair, there are a PRF key $\mathbf{k} = (k_1, \ldots, k_k)$ is randomly selected from $\{0,1\}^k$, k "PRF key matching" matrices $\{\mathbf{B}_j\}_{j\in[k]}$ and two "message matching" matrices $(\mathbf{C_0}, \mathbf{C_1})$ are randomly selected from $\mathbb{Z}_q^{n\times m}$. When the user issuing signatures for a message $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m) \in \{0,1\}^m$, it first computes an evaluated matrix $\mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}} = \mathsf{Eval}(C_{\mathsf{PRF}}, (\{\mathbf{B}^{(i)}_j\}_{j\in[k]}, \mathbf{C}^{(i)}_{\mu_1}, \ldots, \mathbf{C}^{(i)}_{\mu_m})) \in \mathbb{Z}_q^{n\times m}$, then samples a preimage $\mathbf{e}^{(i)}$ with respect to this evaluated matrix. Since the signing algorithm only generates one evaluated matrix for each ring member, our ring signature size does not enlarge redundantly with the number of users, since it grows only *linearly* with the ring size.

Then we explain how to build the key image without relying on random oracle. Our method is straightforward, that is we employ the verification key of a one-time signature scheme as the key image. This one-time signature scheme is proposed by Lyubashevsky and Micciancio [31], which is compact and provably secure in the standard model. Later we will explain how to overcome the obstacles in the security proofs.

[1] In linkable ring signatures, 'key image' is usually a parameter in the output signature tuple. If two signature tuples have the same key image, we say these two signatures are linked.

A New Lattice Basis Extending Algorithm: Supporting Employing Simulation Tool in Linkability and Non-Slanderability Proofs. We note that the *key image* is one of the most important parameters for LRS, since it undertakes the functionality and security. The undertook functionality is the Link algorithm of LRS, on input two signature tuples, each of which contains a key image (say I_1 and I_2, respectively), we say these two signature tuples are linked when $I_1 = I_2$. Note that the key image is contained in the signature tuple, so the key image must have the property of *unforgeability*, otherwise the security is trivially broken. And note that the security notion of *non-slanderability* is broken when the key image is forgeable since the adversary can arbitrarily produce a signature and then link it to any signature that he saw. Moreover, the key image can not be *second-generated* with respect to the same secret key, because the security notion of *linkability* is broken when a user can generate one more key image from the same key pair. Therefore, the key image undertakes two security notions: linkability and non-slanderability. In order to be competent that, i.e., make the key image satisfy the properties of *unforgeability* and *second-generated*, as aforementioned, we employ the inherent properties of one-time signature to resolve that, namely, we employ the one-time verification key of a one-time signature scheme [31] as the key image. *But the barrier is how to simulate that in proofs of linkability and non-slanderability when without resorting to random oracle.* Before explaining that, we first recall the definition of linkability and non-slanderability, the key point for the adversary is to forge or second-generate the key image rather than the whole signature tuple. Especially in the model of linkability, only the public parameters pp is generated by the challenger, and the remained parameters such as verification keys and signatures are generated by the adversary. At the same time, pp are generated by a deterministic algorithm since the randomness is public. Therefore, in this setting, we can observe that it is hard to show a reduction since there are no extra parameters for embedding the hard instance into the simulation, and the adversary's ability of *forging/second-generated the key image* cannot be exploited to solve the underlying hardness.

We resolve that by presenting a new lattice basis extending algorithm BasisExtBindOVK. As prior lattice basis extending algorithms [16,33,2], the BasisExtBindOVK is used in the scenario: For a lattice $\mathbf{L}$ with basis $\mathbf{B}$, to delegate a short basis as the key to a child, the parents employ this algorithm with input $(\mathbf{L}, \mathbf{B})$ to create a new lattice $\mathbf{L}'$ with a short basis $\mathbf{B}'$. After the preimage sampling with respect to each evaluated matrix $\mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}$, our signing algorithm additionally sample a 'check' preimage $\mathbf{e}_{\mathsf{chk}}$ by an extended basis that from the BasisExtBindOVK. In this setting, once the preimage $\mathbf{e}_{\mathsf{chk}}$ passes the validation check, then the key image cannot be forged or seconded-generate unless the underlying hardness assumption is broken. More specifically, in our proofs of linkability and non-slanderability, we build the connection between the challenge instance $\mathbf{A}_{\mathsf{in}}$ and the public matrix $\mathbf{A}_{\mathsf{com}}$ in pp by the simulation tool of SampleRwithBasis [2], then the underlying lattice problem SIS for the given instance $\mathbf{A}$ is resolved by exploiting the adversary's ability in linkability and non-slanderability. For a better illustration, below we explain some details. In our setting, the key image is $\mathsf{vk}_{\mathsf{OTS}} := \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$, the matrix $\mathbf{F}$ inputed in BasisExtBindOVK algorithm is constructed as $\mathbf{F} = [\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}} \mid \mathbf{A}_{\mathsf{com}} + \mathbf{A}]$, and the basis of $\mathbf{F}$ is $\mathbf{S}_{\mathbf{F}} = \left[\begin{smallmatrix} -\mathbf{R} & \mathbf{I}_m \\ \mathbf{T}_{\mathbf{A}}\mathbf{R} & -\mathbf{T}_{\mathbf{A}} \end{smallmatrix}\right]$. Under this setting, there are two ways for the adversary to attack the key image $\mathsf{vk}_{\mathsf{OTS}} := \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$: the adversary produces a $\mathbf{A}^*_{\mathsf{com}} \neq \mathbf{A}_{\mathsf{com}}$ such that $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}^*_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$, or produces a $\mathbf{T}_{\mathbf{A}^*} \neq \mathbf{T}_{\mathbf{A}}$ such that $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^*}$. Recall that both cases need to compute the basis $\mathbf{S}_{\mathbf{F}}$ such that $\mathbf{F} \cdot \mathbf{S}_{\mathbf{F}} = \mathbf{0}$, then by our construction on matrices $\mathbf{F}$ and $\mathbf{S}_{\mathbf{F}}$, this equation can be transformed to $\mathbf{A}_{\mathsf{com}} \cdot \mathbf{R}' = \mathbf{0}$ where $\mathbf{R}'$ is transformed from $\mathbf{S}_{\mathbf{F}}$. In our parameters setting, the matrix $\mathbf{R}'$ has a low Gram-Schmidt norm (cf. Section 4.3). Then the simulator can build a connection between $\mathbf{A}_{\mathsf{in}}$ and $\mathbf{A}_{\mathsf{com}}$ by the simulation tool of SampleRwithBasis, the result is $\mathbf{A}_{\mathsf{in}}\bar{\mathbf{R}} = \mathbf{A}_{\mathsf{com}}$ where $\bar{\mathbf{R}}$ is a low-norm matrix. By $\mathbf{A}_{\mathsf{com}} \cdot \mathbf{R}' = \mathbf{0}$ and $\mathbf{A}_{\mathsf{in}}\bar{\mathbf{R}} = \mathbf{A}_{\mathsf{com}}$, we have $\mathbf{A}_{\mathsf{in}} \cdot (\bar{\mathbf{R}}\mathbf{R}') = \mathbf{0}$, and thus the underlying hardness assumption SIS is broken. This completes the reduction.

Proofs Outline. Below we give the outlines of Unforgeability and Signer-Anonymity proofs, as the Linkability and Non-Slanderable were given above. We note that our security models provide strong guarantees to capture the security requirements that practical scenarios imposed on LRS. Particularly, all security models allow the adversary to obtain the randomnesses used in *Setup phase*, which implies the algorithm is public, and does not rely on a trusted setup that may incur concerns on the existing of trapdoors hidden in the output parameters. On the security notion of unforgeability, our work achieves the *strongly unforgeable w.r.t. insider corruption*, namely, it allows the adversary to *corrupt the signing keys of honest ring members* and allows to query the signing oracle with *adversarially-chosen-ring*. Since the *Probing phase* of security models of Anonymity and Non-Slanderability is as same as the model of unforgeability, so both models also support the attacks w.r.t. insider corruption and adversarially-chosen-ring. Furthermore, the property of '*strongly*' unforgeable of unforgeability means the adversary is allowed to output a forgery signature with respect to a queried ring and message.

Unforgeability Proof Outline. The barrier in the security proof of unforgeability is how to simulate the signing oracle when the adversary queries on the index $i^\diamond$ but the simulator does not have the corresponding signing key $\mathbf{T}_{\mathbf{A}^{(i^\diamond)}}$. More specifically, the simulator first randomly picks one index $i^\diamond$ from $[N]$, then embed the challenge instance $\mathbf{A}$ in the verification key of the $i^\diamond$-th ring member. But in this setting, the simulator cannot response the correct signature with respect to index $i^\diamond$ since he does not know the corresponding signing key. Our method for resolving that is inspired by unforgeability proof of the standard signature work [10], in which they employ the public basis of a gadget matrix $\mathbf{G}$ to respond the signing query on index $i^\diamond$, and this public basis cannot be exploited by the adversary to provide a valid forgery in challenge phase. Below we describe the simulation details.

In the Setup phase of the simulation, the simulator first embeds a randomly picked PRF secret key $\mathbf{k}^{(i)} = (k_1^{(i)}, k_2^{(i)}, \ldots, k_k^{(i)})$ in each ring member's verification key. Particularly, it first picks a random index $i^\diamond$ from $\{1, \ldots, N\}$ and let $\mathbf{A}^{(i^\diamond)} = \mathbf{A}$ i.e., embed the SIS problem instance into verification key. Then it takes the $\mathbf{A}^{(i^\diamond)}$ as input in $\mathsf{SuperTrapGen}$ algorithm, outputs $(\mathbf{B}^{(i^\diamond)}, \mathbf{T}_{\mathbf{B}^{(i^\diamond)}})$. In this way, it prepared the signing key of the ring member with index $i^\diamond$ which is used to response the corrupting query in the corrupting oracle probing phase. Secondly, for all the index $i \in [N] \setminus i^\diamond$, it uses $\mathsf{TrapGen}$ algorithm to generate $(\mathbf{B}^{(i)}, \mathbf{T}_{\mathbf{B}^{(i)}})$ and then produce $(\mathbf{A}^{(i)}, \mathbf{T}_{\mathbf{A}^{(i)}})$ by taking $\mathbf{B}^{(i)}$ as input to $\mathsf{SuperTrapGen}$ algorithm. Finally, for all the index $i \in [N]$, $j \in [k]$, and $d \in \{0, 1\}$, it constructs the matrices $\mathbf{A}_d^{(i)} = \mathbf{A}^{(i)}\mathbf{R}_{\mathbf{A}}^{(i)} + d\mathbf{G}$, $\mathbf{B}_j^{(i)} = \mathbf{A}^{(i)}\mathbf{R}_j^{(i)} + k_j^{(i)}\mathbf{G}$, and $\mathbf{C}_d^{(i)} = \mathbf{A}^{(i)}\mathbf{R}_{\mathbf{C}}^{(i)} + d\mathbf{G}$ where all the $\mathbf{R}$ shape matrices are randomly chosen from $\{1, -1\}^{m \times m}$ and $\mathbf{G}$ is the gadget matrix. In this way, the reduction algorithm can response the signing query of all the ring members. For the ring members with index $i \in [N] \setminus i^\diamond$, it responses the signature by the basis $\mathbf{T}_{\mathbf{A}^{(i)}}$. For the ring member with index $i^\diamond$, it responds the signature by the gadget trapdoor $\mathbf{T}_{\mathbf{G}}$. For a valid forgery with respect to message $\boldsymbol{\mu}^*$, since $d = \mathsf{PRF}(\mathbf{k}^{(i^\diamond)}, \boldsymbol{\mu}^*)$ is unpredictable to the adversary, therefore, the reduction algorithm outputs a valid SIS solution with essential probability $1/2$.

Anonymity Proof Outline. We use two techniques to hide the identity of the real signer, the lattice basis randomization algorithm $\mathsf{BasisRand}$ as above mentioned and the methodology of a group signature from Gordon et al. [23]. This methodology mainly includes two lattice techniques, $\mathsf{SuperTrapGen}$ algorithm and a NIWI proof system. Their NIWI proof system has to rely on ROM since it is obtained by Fiat-Shamir transformation, but fortunately, which is resolved by the recent works [37,14]. In our setting, every ring member holds a pair of $(\mathbf{A}^{(i)}, \mathbf{T_A}^{(i)})$ which is generated by $\mathsf{SuperTrapGen}(\mathbf{B}^{(i)})$ such that $\mathbf{A}\mathbf{B}^\top = \mathbf{0}$ where $\mathbf{B}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$. In the signing phase, let s be the index of the signer in the ring, the signer chooses a vector $\mathbf{e}^{(i)} \xleftarrow{\$} \mathbb{Z}^m$ for every ring member except himself, then samples the $\mathbf{e}^{(s)}$ from a specified Gaussian distribution by $\mathbf{T_A}^{(s)}$. Then the signer chooses $\mathbf{x}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^n$ for each ring member, and takes the $\mathbf{e}^{(i)}$ as the LWE *error* in computing $\mathbf{z}^{(i)} = (\mathbf{x}^{(i)})^\top \mathbf{B}^{(i)} + \mathbf{e}^{(i)}$. In this way, the preimage $\mathbf{e}^{(s)}$ that was sampled by the signer is hidden in the LWE ciphertext $\mathbf{z}^{(i)}$ and thus the signer-anonymity is preserved. Finally, the signer produces a NIWI proof of well-formedness, namely, existing a ciphertext $\mathbf{z}^{(s)}$ in the set $\{\mathbf{z}^{(i)}\}_{i \in [N]}$ encrypts a short vector $\mathbf{e}^{(s)}$. Below we describe the simulation details.

The anonymity proof proceeds in a sequence of experiments E_0, H_0, H_1, E_1 such that each experiment is indistinguishable from the one before it. The experiment E_0 (resp., E_1) corresponds to the anonymity experiment (cf. Definition 1) with $b = 0$ (resp., $b = 1$). Let (s_0^*, s_1^*) be the indexes that adversary provides in $\mathsf{Challenge}$ phase. The experiment H_0 is as same as E_0 except that we sample $\mathbf{e}_0^{(s_1^*)}$ by a specified function rather than randomly select it from $\mathbb{Z}_q^m$. Suppose an adversary can distinguish E_0 and H_0 i.e., can distinguish the $\mathbf{z}^{(s_1^*)} = (\mathbf{s}^{(s_1^*)})^\top \mathbf{B}^{(s_1^*)} + \mathbf{e}_0^{(s_1^*)}$ where $\mathbf{e}_0^{(s_1^*)} \leftarrow \mathbb{Z}_q^m$ from the $\mathbf{z}'^{(s_1^*)} = (\mathbf{s}^{(s_1^*)})^\top \mathbf{B}^{(s_1^*)} + \mathbf{e}_0'^{(s_1^*)}$ where $\mathbf{e}_0'^{(s_1^*)} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$, then we can construct a reduction algorithm to solve the LWE assumption. The experiment H_1 is as same as H_0 except that we change the witness from s_0^*'s to s_1^*'s. By the witness indistinguishability of the proof system, H_0 and H_1 are indistinguishable. Finally, H_1 is indistinguishable from E_1 by exactly the same argument used to show the indistinguishability of H_0 and E_0.

2 Definitions

In this section, we introduce 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:*

– $\mathsf{Setup}(1^n) \to \mathsf{PP}$. *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. To guarantee the public has no concerns on the existing of trapdoors for PP, the randomness used in Setup can be included in PP.
  In the following, PP is implicit input parameter to every algorithm.

– $\mathsf{KeyGen}() \to (\mathsf{vk},\mathsf{sk})$. *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.

– $\mathsf{Sign}(\mathsf{sk}, \mu, \mathsf{R}) \to \Sigma$. *On input a signing key* sk*, a message $\mu \in \mathcal{M}$, and a ring of verification keys* $\mathsf{R} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$[2]. *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.

– $\mathsf{Ver}(\mathsf{R}, \mu, \Sigma) \to 1/0$. *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.*

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

[2] 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.

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*

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

- *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*

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

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

- **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 $N = \mathrm{poly}(n)$ 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} of 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 $\mathsf{vk}^{(s)} \in \mathsf{R} \cap \mathsf{S}$, 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-anonymity, 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* $(\mathsf{vk}^{(i)}, \mathsf{sk}^{(i)}) \leftarrow \mathsf{KeyGen}(\gamma_{\mathsf{kg}}^{(i)})$ *for all* $i \in [N]$*, where* $N = \mathrm{poly}(n)$ *and* $(\gamma_{\mathsf{st}}, \gamma_{\mathsf{kg}}^{(i)})$ *are the randomness used in* Setup *and* KeyGen*, respectively.* $\mathcal{C}$ *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**. *As same as the probing phase of **Unforgeability**.*

– **Challenge**. $\mathcal{A}$ *outputs a message* μ^**, a ring of verification keys* R^**, and two distinct indices* $s_0^*, s_1^* \in [N]$*, such that*
(1) $\mathsf{vk}^{(s_0^*)}, \mathsf{vk}^{(s_1^*)} \in \mathsf{S} \cap \mathsf{R}^*$ *and*
(2) *none of* $\mathsf{OSign}(\cdot, \cdot, s_0^*)$*,* $\mathsf{OSign}(\cdot, \cdot, s_1^*)$*,* $\mathsf{OCorrupt}(s_0^*)$*,* $\mathsf{OCorrupt}(s_1^*)$ *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**. *As same as the* **Probing Phase 1***, but with the restriction that none of* $\mathsf{OSign}(\cdot, \cdot, s_0^*)$*,* $\mathsf{OSign}(\cdot, \cdot, s_1^*)$*,* $\mathsf{OCorrupt}(s_0^*)$*,* $\mathsf{OCorrupt}(s_1^*)$ *was 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* **Unforgeability**.

– **Probing Phase.** *As same as the probing phase of* **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)* $(\hat{\mathsf{R}}, \hat{\mu}, \hat{\Sigma}) \in \mathsf{L}$ *where* $\hat{\Sigma}$ *is replied from* $\mathsf{OSign}(\hat{\mu}, \hat{\mathsf{R}}, \hat{i})$ *for some* $\hat{i} \in [N]$*, (3)* $(\mathsf{R}^*, \mu^*, \Sigma^*) \notin \mathsf{L}$*, (4)* $\mathsf{vk}^{(\hat{i})} \notin \mathsf{R}^*$*, (5)* $\mathsf{Link}(\mathsf{R}^*, \mu^*, \Sigma^*, \hat{\mathsf{R}}, \hat{\mu}, \hat{\Sigma}) = 1$*.*

3 Preliminaries

In this section, we first review the strongly unforgeable one-time signature in Sect. 2, 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 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 $\mathbf{A} \mid \mathbf{B}$. 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. For a matrix $\mathbf{A}$ we denote some matrix norms: $\|\mathbf{A}\|_1$ denotes the ℓ_1-norm of $\mathbf{A}$, $\|\mathbf{A}\|$ denotes the ℓ_2-norm of the longest column of $\mathbf{A}$, $\|\mathbf{A}\|_\infty$ denotes the ℓ_∞-norm of $\mathbf{A}$, $\|\widetilde{\mathbf{A}}\|$ denotes the result of applying Gram-Schmidt orthogonalization to the columns of $\mathbf{A}$.

3.1 Strongly Unforgeable One-Time Signature

Our construction will use the 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 a 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:*

- $\mathsf{Setup}(1^n) \to \mathsf{PP}_{\mathsf{OTS}}$. *On input the security parameter 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.

- $\mathsf{KeyGen}() \to (\mathsf{vk}_{\mathsf{OTS}}, \mathsf{sk}_{\mathsf{OTS}})$. *The algorithm outputs a verification key* $\mathsf{vk}_{\mathsf{OTS}}$ *and a signing key* $\mathsf{sk}_{\mathsf{OTS}}$.
- $\mathsf{Sign}(\mathsf{sk}_{\mathsf{OTS}}, \mu) \to \Sigma_{\mathsf{OTS}}$. *On input a signing key* $\mathsf{sk}_{\mathsf{OTS}}$ *and a message $\mu \in \mathcal{M}$, the algorithm outputs a signature* Σ_{OTS}.

– $\mathsf{Ver}(\mathsf{vk}_{\mathsf{OTS}}, \mu, \Sigma_{\mathsf{OTS}}) \to 1/0$. *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.*

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

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

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

Unforgeability. *An 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* $\mathcal{A}$*.*

– **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,* $\mathcal{C}$ *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* $\mathsf{Ver}(\mathsf{vk}_{\mathsf{OTS}}, \mu^*, \Sigma^*_{\mathsf{OTS}}) = 1$*.*

We employ the OTS scheme, asymptotically efficient and without random oracle, that was presented by Lyubashevsky and Micciancio's work [31] as our OTS scheme. The scheme is parametrized by integers n, m, k, q, w, p and $\mathcal{H} = \mathbb{Z}_q^{n\times m}$, $\mathcal{K} = \{\mathbf{K} \in \mathbb{Z}_q^{m\times k} : \|\mathbf{K}\|_\infty \leq p\}$, $\mathcal{M} \subseteq \{\boldsymbol{\mu} \in \{0,1\}^k : \|\boldsymbol{\mu}\|_1 = w\}$, and $\mathcal{S} = \{\mathbf{s} \in \mathbb{Z}_q^m : \|\mathbf{s}\|_\infty \leq wp\}$.

Specifically, this OTS scheme is defined by the following procedures:

– Setup: A random and common matrix $\mathbf{H} \in \mathcal{H} \subseteq \mathbb{Z}_q^{n\times m}$ is chosen and can be shared by all users. To guarantee the public has no concerns on the existing of planted trapdoors in $\mathbf{H}$, it could be demanded that $\mathbf{H} = \mathrm{XOF}(s)$ where XOF is some extendable output function [40] and s is a public seed. The matrix $\mathbf{H}$ will be used as a hash function mapping (a subset of) $\mathbb{Z}_q^m$ to $\mathbb{Z}_q^n$ and extended to matrices in $\mathbb{Z}_q^{m\times k}$.

– $\mathsf{Key\ Generation}$: A signing key $\mathbf{K} \in \mathcal{K}$ is chosen uniformly at random, the corresponding verification key $\hat{\mathbf{K}} = \mathbf{H}\mathbf{K} \in \mathbb{Z}_q^{n\times k}$.

– $\mathsf{Signing}$: On input the signing key $\mathbf{K}$ and message $\boldsymbol{\mu} \in \mathcal{M}$, the signing algorithm outputs $\mathbf{s} = \mathbf{K}\boldsymbol{\mu} \in \mathcal{S}$.

– $\mathsf{Verification}$: On input verification key $\hat{\mathbf{K}}$, message $\boldsymbol{\mu}$, and signature $\mathbf{s}$, checks if $\mathbf{s} \in \mathcal{S}$ and $\mathbf{H}\mathbf{s} = \hat{\mathbf{K}}\boldsymbol{\mu}$ holds, return 1, otherwise return 0.

The correctness and security of the OTS scheme is based on the following three properties:

- Closure. $\mathbf{K}\boldsymbol{\mu} \in \mathcal{S}$ for all $\mathbf{K} \in \mathcal{K}$ and $\boldsymbol{\mu} \in \mathcal{M}$.
- Collision Resistance. The function family $\{\mathbf{H} : \mathbb{Z}_q^m \to \mathbb{Z}_q^n \mid \mathbf{H} \in \mathcal{H}\}$ is collision resistant, it means that for any efficient adversary and any randomly chosen $\mathbf{H}$, outputs a collision $(\mathbf{s} \neq \mathbf{s}', \mathbf{Hs} = \mathbf{Hs}')$ with at most negligible probability.
- $(\frac{1}{2})$-Hiding. For any $\mathbf{H} \in \mathcal{H}$, $\mathbf{K} \in \mathcal{K}$, and $\boldsymbol{\mu} \in \mathcal{M}$, let

$$\mathcal{D}_{\mathbf{H}}(\mathbf{K}, \mathbf{m}) = \{\mathbf{K}' \in \mathcal{K} : \mathbf{HK} = \mathbf{HK}' \wedge \mathbf{K}\boldsymbol{\mu} = \mathbf{K}'\boldsymbol{\mu}\}$$

be the set secret keys that are consistent with the verification key $\mathbf{HK}$ and $\boldsymbol{\mu}$-signature $\mathbf{K}\boldsymbol{\mu}$ associated with $\mathbf{K}$. The scheme is $(\frac{1}{2})$-Hiding if for any $\mathbf{H} \in \mathcal{H}$,

$$\Pr_{\mathbf{K}\in\mathcal{K}}\left[\forall \boldsymbol{\mu} \neq \boldsymbol{\mu}', |\mathcal{D}_{\mathbf{H}}(\mathbf{K}, \mathbf{m}) \cap \mathcal{D}_{\mathbf{A}_{\mathsf{com}}}(\mathbf{K}, \mathbf{m}')| \leq \frac{1}{2} \cdot |\mathcal{D}_{\mathbf{A}_{\mathsf{com}}}(\mathbf{K}, \mathbf{m})|\right] \geq \delta$$

where δ close to 1.

Lemma 1 ([31]). *If the* Closure *property holds, then the OTS scheme is correct.*

Lemma 2 ([31]). *Let $q \geq 2wp\sqrt{m}n^{\Omega(1)}$. Then the function family $\{\mathbf{H} : \mathcal{S} \to \mathbb{Z}_q^n \mid \mathbf{H} \in \mathcal{H}\}$ satisfies the* Collision Resistance *property based on the hardness of the $SIS_{n,m,q,2wp}$ problem.*

Lemma 3 ([31]). *Let $p = \lceil \frac{q^{n/m}2^{n/m}-1}{2} \rceil$. Then the OTS scheme satisfies the* $(\frac{1}{2})$-Hiding *property.*

Lemma 4 ([31]). *If the* Closure, Collision Resistance, *and* $(\frac{1}{2})$-Hiding *properties hold, then the OTS scheme is strongly unforgeable.*

3.2 Key-Homomorphic Evaluation Algorithm

In our construction, we borrow the idea from the standard signature work [10], that is employing the key-homomorphic evaluation algorithm $\mathsf{Eval}(\cdot,\cdot)$ from [22,12,8] to evaluate circuits of a PRF. In particular, they used the evaluation algorithm of the work [12]. 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 [33]. The algorithm deterministically output a matrix $\mathbf{A}_C = \mathbf{A}\mathbf{R}_C + C(b^{(1)}, \ldots, b^{(\ell)})\mathbf{G} \in \mathbb{Z}_q^{n\times m}$. The following lemma stats that $\mathbf{R}_C$ is short enough, which will be used in the analyzation of our unforgeability proof.

Lemma 5 ([10]). *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 first review the NIWI proof system presented by Gordon et al. [23]. 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:

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

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

By the methodology of Gordon et al. [23], there is an interactive witness indistinguishable proof system for $L_{\sigma,\varepsilon}$ when set $\varepsilon \ge O(\sqrt{m/\log m})$ by using the techniques of the work [35], then the resulting protocol can be made non-interactive in the standard model by applying the Fiat-Shamir transformation from the work [37]. We can summarize these observations as the following lemma.

Lemma 6. *Let $\varepsilon \ge O(\sqrt{m/\log m})$. There is an NIWI proof system for $L_{\sigma,\varepsilon}$ in the standard model.*

3.4 Lattice Backgrounds

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

Lemma 7 ([1]). *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 $\Pr\left[\|\mathbf{R}\| > c\sqrt{k+m}\right] < e^{-(k+m)}$.*

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., $\mathbf{\Lambda} = \mathcal{L}(\mathbf{B}) = \left\{\mathbf{B}\mathbf{c} = \sum_{i=1}^m c_i \mathbf{b}_i : \mathbf{c} \in \mathbb{Z}^m\right\}$. 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 $\mathbf{\Lambda}_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \pmod q\}$. $\mathbf{\Lambda}_q^{\mathbf{y}}(\mathbf{A})$ is defined as $\mathbf{\Lambda}_q^{\mathbf{y}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{y} \pmod q\}$. 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 bounds the length of a discrete Gaussian vector with a sufficiently large Gaussian parameter.

Lemma 8 ([34]). *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 $\Pr[\|\mathbf{x}-\mathbf{c}\| > \sigma\sqrt{m} : \mathbf{x} \leftarrow \mathcal{D}_{\mathbf{\Lambda},\sigma,\mathbf{c}}] \leq \mathrm{negl}(n)$.*

Micciancio and Goldwassser [32] showed that a full-rank set $\mathbf{T}$ in a lattice $\mathbf{\Lambda}$ can be converted into a basis $\mathbf{B}$ with an equally low Gram-Schmidt norm.

Lemma 9 ([32]). *Let $\mathbf{\Lambda}$ be an m-dimensional lattice. There is a deterministic algorithm that, given an arbitrary basis of $\mathbf{\Lambda}$ and a full-rank set $\mathbf{T}$ in $\mathbf{\Lambda}$, returns a basis $\mathbf{B}$ of $\mathbf{\Lambda}$ such that $\|\widetilde{\mathbf{B}}\| \leq \|\widetilde{\mathbf{T}}\|$.*

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

Lemma 10 ([1]). *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 $\{1,-1\}^{m\times k} \bmod q$ 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 $\mathbf{v}$ in $\mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{A}\mathbf{R}, \mathbf{R}^\top\mathbf{v})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top\mathbf{v})$.*

Definition 3 (SIS Assumption [21,34]). *Let q and β be functions of n. An instance of the $\mathsf{SIS}_{q,\beta}$ problem is a uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ for any desired $m = \mathrm{poly}(n)$. The goal is to find a nonzero integer vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$ and $\|\mathbf{x}\| \leq \beta$. For $\beta = \mathrm{poly}(n)$, $q \geq \beta \cdot \omega(\sqrt{n \log n})$, no (quantum) algorithm can solve $\mathsf{SIS}_{q,\beta}$ problem in polynomial time.*

We use the LWE assumption proposed by Gordon et al. [23] and they proved it is implied by the standard LWE assumption [38]. 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 [38]). *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 $(\mathbf{a}, \mathbf{a}^\top \mathbf{x} + e)$. Defines the Search-$\mathsf{LWE}_{q,n,m,\chi}$ as: Fix an $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_q^n$, 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 $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_q^n$, 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, α =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 $k = k(n)$, $m = m(n)$ and $c = c(n)$. A pseudorandom function $\mathsf{PRF} : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^c$ 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:*

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

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 11 ($\mathsf{SuperTrapGen}$ **Algorithm [23]**)**.** *Let $n \geq 1, q \geq 2, m = O(n \log q)$ be integers. There is a probabilistic algorithm $\mathsf{SuperTrapGen}(1^n, 1^m, q, \mathbf{B})$ 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 $\mathbf{T_A} \subset \mathbf{\Lambda}_q^\perp(\mathbf{A})$ i.e., $\mathbf{T_A}$ is a basis (full-rank subset) of $\mathbf{\Lambda}_q^\perp(\mathbf{A})$ 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 $\|\widetilde{\mathbf{T}}_\mathbf{A}\| = O(\log n \cdot \sqrt{mn \log q})$ and $\|\mathbf{T_A}\| = O(n \log n \sqrt{m} \cdot \log q)$ with all but negligible probability in n.*

Lemma 12 ($\mathsf{SampleRwithBasis}$ **Algorithm** [2])**.** *Let $q > 2$ be a prime, $m > n$ be integers. There is a probabilistic algorithm $\mathsf{SampleRwithBasis}(\mathbf{A})$ which takes as input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ whose columns generate $\mathbb{Z}_q^n$, then generates a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{S_B}$ of $\mathbf{\Lambda}_q^\perp(\mathbf{B})$ by invoking trapdoor generation algorithm. The algorithm outputs a $\mathbb{Z}_q$-invertible matrix $\mathbf{R}$ in $\mathbb{Z}^{m \times m}$ from a distribution that is statistically close to $\mathcal{D}_{m \times m}$ such that $\mathbf{B} = \mathbf{A}\mathbf{R}^{-1} \pmod q$.*

Lemma 13 ($\mathsf{BasisExt}$ **Algorithm** [16])**.** *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 $\mathbf{A}' = [\mathbf{A}_1 \mid \mathbf{A}_2 \mid \mathbf{A}_3]$. Let $\mathbf{T}_{\mathbf{A}_2}$ be a basis of $\mathbf{\Lambda}^\perp(\mathbf{T}_{\mathbf{A}_2})$. There is a deterministic algorithm $\mathsf{BasisExt}(\mathbf{T}_{\mathbf{A}_2}, \mathbf{A}')$ that outputs a basis $\mathbf{T}_{\mathbf{A}'}$ for $\mathbf{\Lambda}^\perp(\mathbf{A}')$ such that $\|\widetilde{\mathbf{T}}_{\mathbf{A}'}\| = \|\widetilde{\mathbf{T}}_{\mathbf{A}_2}\|$.*

Lemma 14 (BasisRand **Algorithm** [16]). *Let $\mathbf{S}_{\mathbf{A}'} \in \mathbb{Z}^{m' \times m'}$ be an extended basis of $\mathbf{\Lambda}^{\perp}(\mathbf{A}')$ output by the* BasisExt *algorithm. There is a probabilistic algorithm* $\mathsf{BasisRand}(\mathbf{S}_{\mathbf{A}'}, \sigma)$ *which takes as input a basis $\mathbf{S}_{\mathbf{A}'}$ and a parameter $\sigma \geq \|\widetilde{\mathbf{S}}_{\mathbf{A}'}\| \cdot \omega(\sqrt{\log m})$, outputs a basis $\mathbf{S}_{\mathbf{A}''} \in \mathbb{Z}^{m' \times m'}$ of $\mathbf{\Lambda}^{\perp}(\mathbf{A}')$ which is statistically independent with the original basis $\mathbf{S}_{\mathbf{A}'}$, and has $\|\widetilde{\mathbf{S}}_{\mathbf{A}''}\| \leq \sigma \cdot \sqrt{m'}$ holds with overwhelming probability.*

The following lattice basis extension algorithm also needed for our security proof, which presented by Agrawal, Boneh and Boyen [1], so we abbreviate that as BasisExtABB algorithm.

Lemma 15 (BasisExtABB **Algorithm** [1]). *Let q be a prime, n, m be integers with $m > n$. There is a probabilistic algorithm* $\mathsf{BasisExtABB}(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{T}_{\mathbf{B}})$ *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 $\mathbf{T}_{\mathbf{B}} \in \mathbf{\Lambda}_q^{\perp}(\mathbf{B})$, outputs a full-rank matrix $\mathbf{T}_{\mathbf{F}}$ in $\mathbf{\Lambda}_q^{\perp}(\mathbf{F})$ such that $\|\widetilde{\mathbf{T}}_{\mathbf{F}}\| < (\|\mathbf{R}\| + 1) \cdot \|\widetilde{\mathbf{T}}_{\mathbf{B}}\|$ where $\mathbf{F} = [\mathbf{A} \mid \mathbf{AR} + \mathbf{B}] \in \mathbb{Z}_q^{n \times 2m}$.*

Lemma 16 (SamplePre **Algorithm** [21]). *Let $q > 2$, $m > n$ be integers. There is a probabilistic algorithm* $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}_{\mathbf{A}}, \mathbf{u}, \sigma)$ *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 $\mathbf{T}_{\mathbf{A}}$ of $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$, and a Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T}}_{\mathbf{A}}\| \cdot \omega(\sqrt{\log m})$, outputs a vector $\mathbf{e} \in \mathbf{\Lambda}_q^{\mathbf{u}}(\mathbf{A})$ sampled from a distribution which is statistically close to $\mathcal{D}_{\mathbf{\Lambda}_q^{\mathbf{u}}(\mathbf{A}), \sigma}$.*

Lemma 17 (SampleR **Algorithm** [2]). *Let $q > 2$ be a prime, $m > n$ be integers. There is a probabilistic algorithm* $\mathsf{SampleR}(1^m)$ *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 $\|\widetilde{\mathbf{R}}\| \leq O(\sqrt{mn \log q}) \cdot \omega(\sqrt{\log m})$.*

Gadget Matrix. The "gadget matrix" $\mathbf{G}$ defined in [33]. We recall the following one fact of $\mathbf{G}$.

Lemma 18 ([33]). *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 $\mathbf{T}_{\mathbf{G}} \in \mathbb{Z}^{m \times m}$ with $\|\widetilde{\mathbf{T}}_{\mathbf{G}}\| \leq \sqrt{5}$.*

4 Our Construction

In this section, we give the lattice basis extension algorithm BasisExtBindOVK in Sect. 4.1, based on that we present the construction of LRS in Sect. 4.2, and then we give the concrete parameters in Sect. 4.3.

4.1 Lattice Basis Extending Algorithm

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

Inputs: A matrix $\mathbf{A}$ whose columns generate $\mathbb{Z}_q^n$, a basis $\mathbf{T}_\mathbf{A}$ of $\mathbf{\Lambda}_q^\perp(\mathbf{A})$, and a matrix $\mathbf{F} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_\mathbf{A} \mid \mathbf{A}_{\mathsf{com}} + \mathbf{A}\right] \in \mathbb{Z}_q^{n\times 2m}$ 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 $\mathsf{BasisExtBindOVK}$ algorithm runs as follows:

1. Sample $\mathbf{R} \leftarrow \mathsf{SampleR}(1^m)$.
2. Let $\mathbf{I}_m$ be a $m \times m$ identity matrix. Construct $\mathbf{S}_\mathbf{F} = \left[\begin{smallmatrix} -\mathbf{R} & \mathbf{I}_m \\ \mathbf{T}_\mathbf{A}\mathbf{R} & -\mathbf{T}_\mathbf{A} \end{smallmatrix}\right]$. Note that $\mathbf{F} \cdot \mathbf{S}_\mathbf{F} = \mathbf{0} \pmod q$.
3. Use Lemma 9 to convert $\mathbf{S}_\mathbf{F}$ into a basis $\mathbf{T}_\mathbf{F}$ of $\mathbf{\Lambda}_q^\perp(\mathbf{F})$ with the same Gram-Schmidt norm as $\mathbf{S}_\mathbf{F}$.

Lemma 19. *The matrix* $\mathbf{S}_\mathbf{F}$ *output by* $\mathsf{BasisExtBindOVK}$ *is full-rank and satisfy* $\|\widetilde{\mathbf{S}}_\mathbf{F}\| \le O(m^2) \cdot \omega(\sqrt{\log m})$.

Proof. By Lemma 17, we know the matrix $\mathbf{R}$ is invertible. By Lemma 11, we know the basis $\mathbf{T}_\mathbf{A}$ of $\mathbf{\Lambda}_q^\perp(\mathbf{A})$ is full-rank. Therefore, the matrix $\mathbf{S}_\mathbf{F}$ is full-rank, so we can convert $\mathbf{S}_\mathbf{F}$ into a basis $\mathbf{T}_\mathbf{F}$ of $\mathbf{\Lambda}_q^\perp(\mathbf{F})$ by Lemma 9. By Lemma 17, we know the Gram-Schmidt norm of $\mathbf{R}$ is bounded by $O(\sqrt{mn\log q}) \cdot \omega(\sqrt{\log m})$. By Lemma 11, we know $\|\widetilde{\mathbf{T}}_\mathbf{A}\| \le O(\log n \cdot \sqrt{mn\log q})$. As analyzed in Sect. 4.3, it requires to set $m = O(n \log q)$. Therefore, we have $\|\widetilde{\mathbf{S}}_\mathbf{F}\| \le O(m^2)\cdot\omega(\sqrt{\log m})$.

4.2 Construction

$\mathsf{Setup}(1^n)$

1. On input a security parameter n, sets the parameters $q, m, k, \sigma_1, \sigma_2, \sigma_3$ as specified in Sect. 4.3 below.
2. Select a secure $\mathsf{PRF} : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}$, express it as a NAND Boolean circuit C_{PRF}.
3. Sample $\mathbf{A}_{\mathsf{com}} = \mathsf{XOF}(s)$ where $\mathbf{A}_{\mathsf{com}} \in \mathbb{Z}_q^{n\times m}$.
4. Output the public parameters $\mathsf{PP} = (q, m, k, \sigma_1, \sigma_2, \sigma_3, \mathsf{PRF}, \mathbf{A}_{\mathsf{com}}, s)$.

Note that including the seed s 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 planted trapdoors in $\mathbf{A}_{\mathsf{com}}$.
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 generate $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{SuperTrapGen}(1^n, 1^m, q, \mathbf{B})$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$.
2. Let $\mathsf{sk_{OTS}} := \mathbf{T_A}$ and $\mathsf{vk_{OTS}} := \mathbf{A}_{\mathsf{com}}\mathbf{T_A}$.
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 $\mathsf{sk} = (\mathbf{T_A}, \mathbf{k}, \mathsf{vk_{OTS}})$.

In the rest of the construction, for a ring $\mathsf{R} = (\mathsf{vk}^{(1)}, \ldots, \mathsf{vk}^{(N)})$, we implicitly parse each verification key $\mathsf{vk}^{(i)} = (\mathbf{A}^{(i)}, (\mathbf{A}_0^{(i)}, \mathbf{A}_1^{(i)}), \mathbf{B}^{(i)}, \{\mathbf{B}_j^{(i)}\}_{j\in[k]}, (\mathbf{C}_0^{(i)}, \mathbf{C}_1^{(i)}))$, and the corresponding signing key $\mathsf{sk}^{(i)} = (\mathbf{S}_{\mathbf{A}^{(i)}}, \mathbf{k}^{(i)}, \mathsf{vk}_{\mathsf{OTS}}^{(i)})$.

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

1. On input a signing key $\mathsf{sk}^{(s)}$ where $s \in [N]$ is the index of the signer in the ring R, a message $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m) \in \{0,1\}^m$, and a ring of verification keys R.
2. Compute $d = \mathsf{PRF}(\mathbf{k}^{(s)}, \boldsymbol{\mu})$.
3. For $i = 1$ to N, compute $\mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}} = \mathsf{Eval}(C_{\mathsf{PRF}}, (\{\mathbf{B}_j^{(i)}\}_{j\in[k]}, \mathbf{C}^{(i)}_{\mu_1}, \ldots, \mathbf{C}^{(i)}_{\mu_m}))$ $\in \mathbb{Z}_q^{n\times m}$, set $\mathbf{F}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d} = \left[\mathbf{A}^{(i)} \mid \mathbf{A}^{(i)}_{1-d} - \mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}\right] \in \mathbb{Z}_q^{n\times 2m}$.
4. For $i = 1$ to N, select $\mathbf{u}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^n$, compute $\mathbf{e}_1^{(i)} \leftarrow \mathsf{SamplePre}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}}, \mathbf{u}^{(i)}, \sigma_1)$.
5. For $i = s$, compute $\mathbf{e}_0^{(s)} \leftarrow \mathsf{SamplePre}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}},\ \mathbf{u}'^{(s)}, \sigma_1)$ where $\mathbf{u}'^{(s)} = (\mathbf{A}^{(s)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}} - \mathbf{A}^{(s)}_{1-d}) \cdot \mathbf{e}_1^{(s)}$.
6. 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 $\mathbf{F}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d} \cdot (\mathbf{e}_0^{(i)}; \mathbf{e}_1^{(i)}) = \mathbf{0} \pmod q$.
7. For $i = 1$ to N, select $\mathbf{x}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^n$, compute $\mathbf{z}^{(i)} = (\mathbf{x}^{(i)})^\top \mathbf{B}^{(i)} + \mathbf{e}_0^{(i)}$.
8. Use the witness $\{\mathbf{x}^{(i)}, i\}_{i\in[N]}$ to construct an NIWI proof π for the gap language $L_{\sigma,\varepsilon}$ as Sect. 3.3.
9. Compute the one-time signature $\mathbf{s} = \mathbf{T}_{\mathbf{A}^{(s)}}\boldsymbol{\mu}$
10. For $i = 1$ to N, set $\mathbf{F}^{(i)} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} \mid \mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(i)}\right] \in \mathbb{Z}_q^{n\times 2m}$.
11. Compute $\mathbf{T}_{\mathbf{F}^{(s)}} \leftarrow \mathsf{BasisExtBindOVK}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}}, \mathbf{F}^{(s)})$.
12. Compute $\mathbf{T}_{\mathbf{F}_{\mathsf{chk}}} \leftarrow \mathsf{BasisRand}(\mathsf{BasisExt}(\mathbf{T}_{\mathbf{F}^{(s)}}, \mathbf{F}_{\mathsf{chk}}), \sigma_2)$ where $\mathbf{F}_{\mathsf{chk}} = [\mathbf{F}^{(1)} \mid \cdots \mid \mathbf{F}^{(N)}] \in \mathbb{Z}_q^{n\times 2Nm}$.

13. Sample $\mathbf{e}_{\mathsf{chk}} \leftarrow \mathsf{SamplePre}(\mathbf{F}_{\mathsf{chk}}, \mathbf{T}_{\mathbf{F}_{\mathsf{chk}}}, \mathbf{0}, \sigma_3)$.
14. Output the signature $\Sigma = (\mathbf{s}, \mathsf{vk}_{\mathsf{OTS}}, \mathbf{e}_{\mathsf{chk}}, \{\mathbf{e}_1^{(i)}, \mathbf{z}^{(i)}\}_{i\in[N]}, \pi)$.

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

1. On input a ring of verification keys R, a message $\boldsymbol{\mu}$, and a signature Σ.
2. Compute $\mathbf{F}_{\mathsf{chk}}$ as in Sign algorithm. Check if $\|\mathbf{e}_{\mathsf{chk}}\| \le \sigma_3\sqrt{2Nm}$ 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 $\mathbf{F}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},d} = \left[\mathbf{A}^{(i)} \mid \mathbf{A}_d^{(i)} - \mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}\right]$ as Sign algorithm. Check if $\|\mathbf{e}_1^{(i)}\| \le \sigma_1\sqrt{m}$ 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 $\mathbf{s}$ is well-formed and $\mathbf{A}_{\mathsf{com}} \cdot \mathbf{s} = \mathsf{vk}_{\mathsf{OTS}} \cdot \boldsymbol{\mu}$, otherwise return 0.
5. Check if the proof π is correct, return 1, otherwise return 0.

$\mathsf{Link}(\mathsf{R}_0, \boldsymbol{\mu}_0, \Sigma_0, \mathsf{R}_1, \boldsymbol{\mu}_1, \Sigma_1)$

1. On input two valid signature tuples $(\mathsf{R}_0, \boldsymbol{\mu}_0, \Sigma_0)$ and $(\mathsf{R}_1, \boldsymbol{\mu}_1, \Sigma_1)$.
2. Let $\mathsf{vk}_{\mathsf{OTS},0}$ and $\mathsf{vk}_{\mathsf{OTS},1}$ be the one-time verification keys in Σ_0 and Σ_1, respectively.
3. 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. We first prove the OTS scheme satisfies the $\mathsf{Closure}$ property. In our parameter setting below, it is required to set $q = O(\ell^{4c} \cdot m^4) \cdot (\omega(\sqrt{\log m}))^2$ for some constant c where $\ell = k + m$ is the input length of PRF, $m = 6n^{1+\tau}$ where $\tau > 0$ is a constant such that $n^\tau > O(\log n)$. To ensure the OTS scheme is $\frac{1}{2}$-Hiding, it is required to set $p = \lceil \frac{q^{n/m}2^{n/m}-1}{2} \rceil$ (see Lemma 3), therefore, $p < q$. In this setting, it holds that $\|\mathbf{T}_{\mathbf{A}^{(s)}}\|_\infty \le p$ since $\|\mathbf{T}_{\mathbf{A}^{(s)}}\| = O(n \log n\sqrt{m} \cdot \log q)$ by Lemma 11. Therefore, $\mathbf{T}_{\mathbf{A}^{(s)}} \in \mathcal{K}$ and $\mathbf{T}_{\mathbf{A}^{(s)}}\boldsymbol{\mu} \in \mathcal{S}$ i.e., $\mathbf{s} \in \mathcal{S}$ for all $\mathbf{T}_{\mathbf{A}^{(s)}} \in \mathcal{K}$, and so the $\mathsf{Closure}$ property holds and $\mathbf{s}$ is well-formed.

By Lemma 16, each $\mathbf{e}_1^{(i)}$ in Σ follows the distribution $\mathcal{D}_{\Lambda_q^{\mathbf{u}^{(i)}}(\mathbf{A}^{(s)}),\sigma_1}$, then by the construction of $\mathbf{z}^{(i)}$ and Lemma 11, 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 16, the $\mathbf{e}_{\mathsf{chk}}$ in Σ follows the distribution $\mathcal{D}_{\Lambda_q^\perp(\mathbf{F}_{\mathsf{chk}}),\sigma_3}$, therefore, it holds that $\mathbf{F}_{\mathsf{chk}} \cdot \mathbf{e}_{\mathsf{chk}} = \mathbf{0} \pmod q$. By Lemma 8, $\mathbf{e}_1^{(i)} \le \sigma\sqrt{m}$ and $\mathbf{e}_{\mathsf{chk}} \le \sigma_3\sqrt{2Nm}$ holds with overwhelming probability. Therefore, the signature is accepted by the Ver algorithm with overwhelming probability.

For the correctness of Link, let $\Sigma_0 = (\Sigma_{\mathsf{OTS},0}, \mathsf{vk}_{\mathsf{OTS},0} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(0)}}, \dots)$ and $\Sigma_1 = (\Sigma_{\mathsf{OTS},1}, \mathsf{vk}_{\mathsf{OTS},1} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(1)}}, \dots)$ be generated by $\mathsf{sk}_0 = \mathbf{T}_{\mathbf{A}^{(0)}}$ and

$\mathsf{sk}_1 = \mathbf{T}_{\mathbf{A}^{(1)}}$, respectively. In the case $\mathsf{sk}_0 = \mathsf{sk}_1$ i.e., $\mathbf{T}_{\mathbf{A}^{(0)}} = \mathbf{T}_{\mathbf{A}^{(1)}}$, the signer-linkable proof in Sect. 5 shows that it is infeasible to change $\mathbf{A}_{\mathsf{com}}$ to a $\mathbf{A}'_{\mathsf{com}}$ such that $\mathsf{vk}_{\mathsf{OTS},0} \neq \mathsf{vk}_{\mathsf{OTS},1}$ unless the underlying hardness assumption is broken, therefore, $\mathsf{vk}_{\mathsf{OTS},0} = \mathsf{vk}_{\mathsf{OTS},1}$ holds with overwhelming probability. In the case $\mathsf{sk}_0 \neq \mathsf{sk}_1$ i.e., $\mathbf{T}_{\mathbf{A}^{(0)}} \neq \mathbf{T}_{\mathbf{A}^{(1)}}$, the signer-non-slanderable proof in Sect. 5 shows that it is infeasible to compute a $\mathbf{A}'_{\mathsf{com}}$ such that $\mathsf{vk}_{\mathsf{OTS},0} = \mathsf{vk}_{\mathsf{OTS},1}$ unless the underlying hardness assumption is broken, therefore, $\mathsf{vk}_{\mathsf{OTS},0} \neq \mathsf{vk}_{\mathsf{OTS},1}$ holds with overwhelming probability.

We now explain the parameters choosing.

- Let n be the security parameter and $k = k(n)$ be the secret key length of PRF. To ensure that hard lattices with good short bases can be generated by $\mathsf{SuperTrapGen}$, we need to set $m = 6n^{1+\tau}$ where $\tau > 0$ is a constant such that $n^\tau > O(\log n)$.
- To ensure that the distribution on the output of $\mathsf{SamplePre}$ statistically close to the distribution $\mathcal{D}_{\mathbb{Z}^m,\sigma_1}$ and $\mathcal{D}_{\mathbb{Z}^{2Nm},\sigma_3}$, we need to set the Gaussian parameter σ_1 and σ_3 sufficiently large that is $\sigma_1 = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$ and $\sigma_3 = O(N^{1/2} \cdot m^{5/2}) \cdot \omega(\sqrt{\log m})$, respectively (see the unforgeability proof below). To ensure the distribution on the output of $\mathsf{BasisRand}$ statistically independent with the original basis, we need to set the $\sigma_2 = O(m^2) \cdot \omega(\sqrt{\log m})$ (see the unforgeability 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 $\beta \geq O(\ell^{2c} \cdot m^{3/2}) \cdot \sigma_1 \sqrt{m}$ for some constant c (see the unforgeability proof below).
- For the parameter q, we employ the work [26] to instantiate our PRF, which based on standard LWE assumption with polynomial modulus $q = O(n^{8+\epsilon})$ for any $\epsilon \in (0, 1)$. On the other hand, we employ the work [31] to instantiate our OTS, it is required to set $q \geq 2wp\sqrt{m}n^{\Omega(1)}$ such that the property $\mathsf{collision\ resistant}$ holds (see Lemma 2). To ensure our construction based on SIS has a worst-case lattice reduction as defined in Definition 3, we need to set the modulus $q \geq \beta \cdot \omega(\sqrt{n \log n})$. To guarantee the hardness of the based $\mathsf{LWE}_{q,n,m,\chi}$ assumption, we need to set $\alpha = \omega(\sqrt{\log q})/q$ such that $\alpha q = \omega(\sqrt{\log q})$ as defined in Definition 4. To satisfy these requirements, we set $q = O(\ell^{4c} \cdot m^4) \cdot (\omega(\sqrt{\log m}))^2$.

As the parameters set above, we note that the OTS scheme achieves the correctness and strongly unforgeable (see Lemma 1 and 4) and the PRF is secure.

5 Proofs of Security and Privacy

Theorem 1 (Unforgeability). *Let $m, q, \beta, \alpha, \sigma_1, \sigma_2, \sigma_3$ be some polynomials in the security parameter n. For large enough $\sigma_1 = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$, $\sigma_2 = O(m^2) \cdot \omega(\sqrt{\log m})$, $\sigma_3 = O(N^{1/2} \cdot m^{5/2}) \cdot \omega(\sqrt{\log m})$, 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. We proof the theorem by giving a reduction.

Reduction. Suppose the PPT adversary $\mathcal{A}$ has non-negligible advantage in forging the signature by mounting the attack as defined in the security model of Definition 1 on LRS, then there exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ attacking the $\mathrm{SIS}_{q,n,m,\beta}$ problem. Consider the following security game between $\mathcal{A}$ and $\mathcal{S}$. Upon receiving a challenge $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that is formed by m uniformly random and independent samples from $\mathbb{Z}_q^n$, $\mathcal{S}$ simulates as follows.

Setup. $\mathcal{S}$ takes as input a security parameter n and a randomness γ_{st} to invoke $\mathsf{PP} \leftarrow \mathsf{Setup}(1^n;\gamma_{\mathsf{st}})$ 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 $i = i^\diamond + 1, \dots, N, 1, \dots, i^\diamond - 1$:
  - Select $\mathbf{B}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
  - 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 $i = i^\diamond$, select $\hat{\mathbf{K}}^{(i^\diamond)} \xleftarrow{\$} \mathcal{K}$ and set $\mathsf{vk}_{\mathsf{OTS}}^{(i^\diamond)} := \hat{\mathbf{K}}^{(i^\diamond)}$.
- 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 $\mathsf{vk}^{(s)} \in \mathsf{S}\cap\mathsf{R}$. 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.

- Compute $d = \mathsf{PRF}(\mathbf{k}^{(s)}, \boldsymbol{\mu})$.

- For $i = 1$ to N:
  - Compute $\mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}} = \mathsf{Eval}(C_{\mathsf{PRF}}, (\{\mathbf{B}_j\}^{(i)}_{j\in[k]}, \mathbf{C}^{(i)}_{\mu_1}, \mathbf{C}^{(i)}_{\mu_2}, \ldots, \mathbf{C}^{(i)}_{\mu_m}))$
  - Set $\mathbf{F}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d} = \left[\mathbf{A}^{(i)} \mid \mathbf{A}^{(i)}_{1-d} - \mathbf{A}^{(i)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}\right]$.
  - Select $\mathbf{u}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^n$.
- For $i = i^\diamond + 1, \ldots, N, 1, \ldots, i^\diamond - 1$:
  - Compute $\mathbf{e}_1^{(i)} \leftarrow \mathsf{SamplePre}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}}, \mathbf{u}^{(i)}, \sigma_1)$.
  - 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

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

  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 $\mathsf{SamplePre}(\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}, \mathbf{T}_{\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}}, \sigma_1, \mathbf{0})$.
- For $i = 1$ to N, sample $\mathbf{x}^{(i)} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{z}^{(i)} = (\mathbf{B}^{(i)})^\top \mathbf{x}^{(i)} + \mathbf{e}_0^{(i)}$.
- Construct an NIWI proof π for the gap language $L_{\sigma,\varepsilon}$ by using the witness $\{\mathbf{x}^{(i)}, i\}_{i\in[N]}$.
- For $i = 1$ to N, set $\mathbf{F}^{(i)} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(i)}} \mid \mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(i)}\right] \in \mathbb{Z}_q^{n\times 2m}$. Let $\mathbf{F}_{\mathsf{chk}} = [\mathbf{F}^{(1)} \mid \cdots \mid \mathbf{F}^{(N)}] \in \mathbb{Z}_q^{n\times 2Nm}$.
- Compute $\mathbf{T}_{\mathbf{F}^{(s)}} \leftarrow \mathsf{BasisExtBindOVK}(\mathbf{A}^{(s)}, \mathbf{T}_{\mathbf{A}^{(s)}}, \mathbf{F}^{(s)}), \mathbf{T}_{\mathbf{F}_{\mathsf{chk}}} \leftarrow \mathsf{BasisRand}$ $(\mathsf{BasisExt}(\mathbf{T}_{\mathbf{F}^{(s)}}, \mathbf{F}_{\mathsf{chk}}), \sigma_2)$, and $\mathbf{e}_{\mathsf{chk}} \leftarrow \mathsf{SamplePre}(\mathbf{F}_{\mathsf{chk}}, \mathbf{T}_{\mathbf{F}_{\mathsf{chk}}}, \mathbf{0}, \sigma_3)$.
- If $s = i^\diamond$, select $\bar{s} \xleftarrow{\$} \{1, 2, \ldots, N\} \setminus i^\diamond$, then use $\mathbf{T}_{\mathbf{A}^{(\bar{s})}}$ to compute the $\mathbf{T}_{\mathbf{F}^{(\bar{s})}}$, $\mathbf{T}_{\mathbf{F}_{\mathsf{chk}}}$, and $\mathbf{e}_{\mathsf{chk}}$ as the last step.
- If $i \neq i^\diamond$, compute the one-time signature $\mathbf{s} = \mathbf{T}_{\mathbf{A}^{(i)}}\boldsymbol{\mu}$. If $i' = i^\diamond$, uniformly choose $\mathbf{s} \xleftarrow{\$} \mathcal{S}$ subject to the condition that $\mathbf{A}_{\mathsf{com}}\mathbf{s} = \hat{\mathbf{K}}\boldsymbol{\mu}$.
- Return the signature $\Sigma = (\mathbf{s}, \mathsf{vk}^{(s)}_{\mathsf{OTS}}, \mathbf{e}_{\mathsf{chk}}, \{\mathbf{e}_1^{(i)}, \mathbf{z}^{(i)}\}_{i\in[N]}, \pi)$ 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 $\Sigma^* = (\mathbf{s}^*, \mathsf{vk}^*_{\mathsf{OTS}}, \mathbf{e}^*_{\mathsf{chk}}, \{\mathbf{e}_1^{(i^*)}, \mathbf{z}^{(i^*)}\}_{i^*\in[N^*]}, \pi^*)$. $\mathcal{S}$ 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 $\mathsf{Eval}(C_{\mathsf{PRF}}, (\{\mathbf{B}_j^{(i^\diamond)}\}_{j\in[k]}, \mathbf{C}^{(i^\diamond)}_{\mu_1^*}, \mathbf{C}^{(i^\diamond)}_{\mu_2^*}, \ldots, \mathbf{C}^{(i^\diamond)}_{\mu_m^*}))$.
  - Set $\mathbf{F}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*,d} = \left[\mathbf{A}^{(i^\diamond)} \mid \mathbf{A}_d^{(i^\diamond)} - \mathbf{A}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}^*}\right]$.

– Use $\mathbf{T}_{\mathbf{B}^{(i^\diamond)}}$ to recover $\mathbf{e}_0^{(i^\diamond)}$. Then check if $\|\mathbf{e}_0^{(i^\diamond)}\| \le \sigma_1\sqrt{m}$ 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

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

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

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

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

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

Claim. 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. 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 11, 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 chosen uniformly at random while the matrix $\mathbf{B}^{(i^\diamond)}$ was generated by $\mathsf{SuperTrapGen}$. By Lemma 11, 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 10. Therefore, the set of verifications keys S given to $\mathcal{A}$ is statistically close to those in the real attack.

Claim. 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 $\sigma_1 = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$, $\sigma_2 = O(m^2) \cdot \omega(\sqrt{\log m})$, and $\sigma_3 = O(N^{1/2} \cdot m^{5/2}) \cdot \omega(\sqrt{\log m})$.

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. For the π, there is no change in the simulation and real attack. For the $\mathsf{vk_{OTS}}$ and $\mathbf{s}$, there is no change for the queried index $i \in [N] \setminus i^\diamond$ in both simulated and real world. For the queried index $i = i^\diamond$, both $\mathsf{vk_{OTS}}$ and $\mathbf{s}$ are selected from the desired distribution, so they are statistically close to those in the real attack. Therefore, we focus on the entries $(\mathbf{e}_{\mathsf{chk}}, (\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_1^{(N)}))$.

By Lemma 16, for sufficient large Gaussian parameter σ_1 and σ_3, the distribution of the entries $(\mathbf{e}_{\mathsf{chk}}, (\mathbf{e}_1^{(1)}, \ldots, \mathbf{e}_1^{(N)}))$ generated by $\mathsf{SamplePre}$ are statistically close to the distribution of signatures generated in the real scheme. By Lemma 14, for sufficient large Gaussian parameter σ_2, the distribution of the output of the $\mathsf{BasisRand}$ algorithm is statistically independent with the original basis. We compute the σ_1, σ_2, and σ_3 as follows.

- In the simulating signing oracle phase, we constructed $\mathbf{F}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d} = \big[\mathbf{A}^{(i\diamond)} \mid \mathbf{A}^{(i\diamond)}(\mathbf{R}^{(i\diamond)}_{1-d} - \mathbf{R}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}) + (1-2d)\mathbf{G}\big]$. By Lemma 7, $\mathbf{R}^{(i\diamond)}_{1-d} \leq c\sqrt{m}$ for some constant c. By Lemma 5, $\big\|\widetilde{\mathbf{R}}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}\big\| \leq O(\ell^{2c} \cdot m^{3/2})$ for some constant c. Let $\bar{\mathbf{R}}^{(i\diamond)} = \mathbf{R}^{(i\diamond)}_{1-d} - \mathbf{R}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}$. By Lemma 15, $\big\|\widetilde{\mathbf{T}}_{\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}}\big\| < (\big\|\bar{\mathbf{R}}^{(i\diamond)} + 1\big\|) \cdot \big\|\widetilde{\mathbf{T_G}}\big\|$. By Lemma 18, we know $\|\widetilde{\mathbf{T}}_{\mathbf{G}}\| \leq \sqrt{5}$. By Lemma 16, it requires to set $\sigma_1 > \big\|\widetilde{\mathbf{T}}_{\mathbf{F}^{(i\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu},1-d}}\big\| \cdot \omega(\sqrt{\log m})$. To satisfy these requirements, set $\sigma_1 = O(\ell^{2c} \cdot m^{3/2}) \cdot \omega(\sqrt{\log m})$ is sufficient.
- By Lemma 11, $\big\|\widetilde{\mathbf{T}}_{\mathbf{A}^{(s)}}\big\| = O(\log n \cdot \sqrt{mn \log q})$. By Lemma 19, $\big\|\widetilde{\mathbf{T}}_{\mathbf{F}^{(s)}}\big\| \leq O(m^2) \cdot \omega(\sqrt{\log m})$. Let $\mathbf{T}'_{\mathbf{F}_{\mathsf{chk}}}$ be the extend basis that output from $\mathsf{BasisExt}$ $(\mathbf{T}_{\mathbf{F}^{(s)}}, \mathbf{F}_{\mathsf{chk}})$. By Lemma 13, $\big\|\widetilde{\mathbf{T}'}_{\mathbf{F}_{\mathsf{chk}}}\big\| = \big\|\widetilde{\mathbf{T}}_{\mathbf{F}^{(s)}}\big\|$. By Lemma 14, it is required to set $\sigma_2 \geq \big\|\widetilde{\mathbf{T}'}_{\mathbf{F}_{\mathsf{chk}}}\big\| \cdot \omega(\sqrt{\log m})$. To satisfy these requirements, set $\sigma_2 = O(m^2) \cdot \omega(\sqrt{\log m})$ is sufficient.
- By Lemma 14, $\big\|\widetilde{\mathbf{T}}_{\mathbf{F}_{\mathsf{chk}}}\big\| \leq \sigma_2\sqrt{2Nm}$.

  By Lemma 16, it is required to set $\sigma_3 \geq \big\|\widetilde{\mathbf{T}}_{\mathbf{F}_{\mathsf{chk}}}\big\| \cdot \omega(\sqrt{\log m})$. To satisfy these requirements, set $\sigma_3 = O(N^{1/2} \cdot m^{5/2}) \cdot \omega(\sqrt{\log m})$ is sufficient.

Claim. $\mathcal{A}$ 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 8, $\|\mathbf{e}_0^{(i^\diamond)}\|, \|\mathbf{e}_1^{(i^\diamond)}\| \leq \sigma_1\sqrt{m}$. By Lemma 5, $\left\|\mathbf{R}^{(i^\diamond)}_{C_{\mathsf{PRF}},\boldsymbol{\mu}}\right\| \leq O(\ell^{2c} \cdot m^{3/2})$. By Lemma 7, the norm of $\mathbf{R}_{\mathbf{A}_d^{(i^\diamond)}}$ is bounded by $\sqrt{m}$. By Lemma 18, $\|\widetilde{\mathbf{T}}_{\mathbf{G}}\| \leq \sqrt{5}$. Therefore, it requires to set $\beta \geq O(\ell^{2c} \cdot m^{3/2}) \cdot \sigma_1\sqrt{m}$.

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 $\bar{\mathbf{R}} \cdot \mathbf{e}_1^{(i^\diamond)} = \bar{\mathbf{r}}_o e_o + \sum_{\bar{o} \in [m] \backslash o} \bar{\mathbf{r}}_{\bar{o}} e_{\bar{o}}$. Note that for the fixed message $\boldsymbol{\mu}^*$ on which $\mathcal{A}$ made the forgery, $\bar{\mathbf{R}}$ (therefore $\bar{\mathbf{r}}_o$) depends on the low-norm matrices $(\mathbf{R}_{\mathbf{A}_0^{(i^\diamond)}}, \mathbf{R}_{\mathbf{A}_1^{(i^\diamond)}}), \left\{\mathbf{R}_{\mathbf{B}_j^{(i^\diamond)}}\right\}_{j \in [k]}, (\mathbf{R}_{\mathbf{C}_0^{(i^\diamond)}}, \mathbf{R}_{\mathbf{C}_1^{(i^\diamond)}})$ and PRF key $\mathbf{k}^{(i^\diamond)}$. The information about $\bar{\mathbf{r}}_o$ 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.

Finally, we analyze $\mathcal{A}$'s advantage. Let ϵ_{LRS} denotes the advantage of $\mathcal{A}$ successfully forge the signature Σ^* with respect to the message it wants to forge. Let ϵ_{PRF} denotes the advantage of $\mathcal{A}$ successfully predict the the bit value b with respect to the message it wants to forge. Let ϵ_{SIS} denotes the advantage of $\mathcal{S}$ successfully output a $\mathsf{SIS}_{q,n,m,\beta}$ solution. Assume the based PRF is secure, $\mathcal{A}$ can not distinguish PRF from random functions, it will randomly pick either $\left\{\mathbf{A}_0^{(i^*)}\right\}_{i^* \in [N^*]}$ or $\left\{\mathbf{A}_1^{(i^*)}\right\}_{i^* \in [N^*]}$ to make a forgery. Therefore, with $\frac{1}{2}$ chance $\mathcal{A}$ will forge the one that $\mathcal{S}$ will be able to use to break the $\mathsf{SIS}_{q,n,m,\beta}$. Moreover, the probability $\mathsf{Pr}[i^\diamond \in \mathsf{R}^*] \geq \frac{1}{N}$. Therefore, we have $\epsilon_{\mathsf{SIS}} \geq \epsilon_{\mathsf{LRS}}/(2N) - \epsilon_{\mathsf{PRF}} - \epsilon_{\mathsf{OTS}} - \mathrm{negl}(n)$ where $\mathrm{negl}(n)$ denote the negligible statistical error in the simulation. For the running time to answer one signing query, $\mathcal{S}$'s running time is bounded by $O(T_{\mathsf{BasisExtABB}} + T_{\mathsf{SamplePre}} + T_{\mathsf{otsSign}} + T_{\mathsf{Eval}})$. So the total running time of $\mathcal{S}$ in the simulation is bounded by $O(Q_{\mathsf{Sign}} \cdot (T_{\mathsf{BasisExtABB}} + T_{\mathsf{SamplePre}} + T_{\mathsf{otsSign}} + T_{\mathsf{Eval}}))$. 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 $\left\{\mathsf{vk}^{(i)}, \mathsf{sk}^{(i)}\right\}_{i\in[N]}$ by repeatedly invoking $\mathsf{KeyGen}(\gamma^{(i)}_{\mathsf{kg}})$, 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 $(\mathsf{R}^*, \boldsymbol{\mu}^*, s^*_0, s^*_1)$ to the challenger after the probing phase, and requires that $s^*_0 \neq s^*_1$, $\mathsf{vk}^{(s^*_0)}, \mathsf{vk}^{(s^*_1)} \in \mathsf{S} \cap \mathsf{R}^*$, and none of $\mathsf{OSign}(\cdot, \cdot, s^*_0)$, $\mathsf{OSign}(\cdot, \cdot, s^*_1)$, $\mathsf{OCorrupt}(s^*_0)$, $\mathsf{OCorrupt}(s^*_1)$ was queried. For the challenge $(\mathsf{R}^*, \boldsymbol{\mu}^*, s^*_0, s^*_1)$, the experiment uses $\mathsf{sk}^{(s^*_0)}$ to compute the signature tuple Σ^* and responses to $\mathcal{A}$. After the probing phase and with the restriction that none of $\mathsf{OSign}(\cdot, \cdot, s^*_0)$, $\mathsf{OSign}(\cdot, \cdot, s^*_1)$, $\mathsf{OCorrupt}(s^*_0)$, $\mathsf{OCorrupt}(s^*_1)$ was queried, $\mathcal{A}$ outputs the guess.

H_0: This experiment is as same as experiment E_0 except that we change how the signature Σ^* is generated: we sample $\mathbf{e}^{(s^*_1)}_0$ by $\mathsf{SamplePre}$ rather than randomly select it from $\mathbb{Z}^m$.

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 $\mathsf{LWE}_{q,n,m,\chi}$.

Reduction. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing E_0 and H_0, then there exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ breaking the hardness assumption $\mathsf{LWE}_{q,n,m,\chi}$. $\mathcal{S}$ is given as input $(\mathbf{B}, \mathbf{z}) \in \mathbb{Z}^{n\times m}_q \times \mathbb{Z}^m_q$, 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 $i^\diamond \xleftarrow{\$} \{1, \ldots, N\}$, sets $\mathbf{B}^{(i^\diamond)} = \mathbf{B}$.
- For $i = i^\diamond + 1, \ldots, N, 1, \ldots, i^\diamond - 1$, select $\mathbf{B}^{(i)} \xleftarrow{\$} \mathbb{Z}^{n\times m}_q$.
- For $i = 1$ to N, compute $(\mathbf{A}^{(i)}, \mathbf{T}_{\mathbf{A}^{(i)}}) \leftarrow \mathsf{SuperTrapGen}(1^n, 1^m, q, \mathbf{B}^{(i)}, \gamma^{(i)}_{\mathsf{kg}})$. Set $\mathsf{vk}^{(i)}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(i)}}$ and $\mathsf{sk}^{(i)}_{\mathsf{OTS}} = \mathbf{T}_{\mathbf{A}^{(i)}}$.
- For $i = 1$ to N and $d \in \{0, 1\}$, select $\mathbf{A}^{(i)}_d, \mathbf{C}^{(i)}_d \xleftarrow{\$} \mathbb{Z}^{n\times m}_q$.
- For $i = 1$ to N, select a PRF key $\mathbf{k}^{(i)} \xleftarrow{\$} \{0, 1\}^k$.
- For $j = 1$ to k, select $\mathbf{B}^{(i)}_j \xleftarrow{\$} \mathbb{Z}^{n\times m}_q$.
- Set $\mathsf{S} = \{\mathsf{vk}^{(i)}\}_{i\in[N]}$, $\mathsf{vk}^{(i)} = (\mathbf{A}^{(i)}, (\mathbf{A}^{(i)}_0, \mathbf{A}^{(i)}_1), \mathbf{B}^{(i)}, \{\mathbf{B}^{(i)}_j\}_{j\in[k]}, (\mathbf{C}^{(i)}_0, \mathbf{C}^{(i)}_1))$, then sends $(\mathsf{PP}, \mathsf{S}, \gamma_{\mathsf{st}})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ provides a challenge $(\mathsf{R}^*, \boldsymbol{\mu}^*, s^*_0, s^*_1)$ 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 $(\mathsf{R}^*, \boldsymbol{\mu}^*, s^*_0, s^*_1)$ in the challenge, $\mathcal{S}$ does as following:

- Let $N^* = |\mathsf{R}^*|$. Check if $s_0^* \neq s_1^*$, $\mathsf{vk}^{(s_0^*)}, \mathsf{vk}^{(s_1^*)} \in \mathsf{S} \cap \mathsf{R}^*$ and $i^\diamond = s_1^*$, otherwise $\mathcal{S}$ outputs a random bit and aborts the simulation.
- Compute $d = \mathsf{PRF}(\mathbf{k}^{(s_0^*)}, \boldsymbol{\mu}^*)$.
- Compute $\mathbf{F}_{\mathsf{chk}}$ 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 $\mathbf{F}^{(s_0^*)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d} \cdot (\mathbf{e}_0^{(s_0^*)}; \mathbf{e}_1^{(s_0^*)}) = \mathbf{0} \pmod q$ 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 $\mathbf{F}^{(s_1^*)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d} \cdot (\mathbf{z}; \mathbf{e}_1^{(i_1^*)}) = \mathbf{0} \pmod q$ 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 $\mathbf{e}_0^{(i^*)} \in \mathbb{Z}^m$ uniformly subject to the condition that $\mathbf{F}^{(i^*)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d} \cdot (\mathbf{e}_0^{(i^*)}; \mathbf{e}_1^{(i^*)}) = \mathbf{0} \pmod q$ 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 $\mathbf{s} = \mathbf{T}_{\mathbf{A}^{(s_0^*)}} \boldsymbol{\mu}^*$.
- Return the signature $\Sigma^* = (\mathbf{s}, \mathsf{vk}_{\mathsf{OTS}}, \mathbf{e}_{\mathsf{chk}}, \{\mathbf{e}_1^{(i^*)}, \mathbf{z}^{(i^*)}\}_{i^* \in [N^*]}, \pi)$ and the randomness set $\{\gamma_{\mathsf{kg}}^{(i)}\}_{i \in [N] \setminus \{s_0^*, s_1^*\}}$ 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. $\mathcal{A}$'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 $\mathbf{F}^{(s_1^*)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d} \cdot (\mathbf{e}_0^{(s_1^*)}; \mathbf{e}_1^{(s_1^*)}) = \mathbf{0} \pmod q$ 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 $\mathbf{F}^{(s_1^*)}_{C_{\mathsf{PRF}}, \boldsymbol{\mu}^*, 1-d} \cdot (\mathbf{z}; \mathbf{e}_1^{(s_1^*)}) = \mathbf{0} \pmod q$. 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 10, 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. $\mathcal{A}$'s view in $\mathcal{D}_{\mathsf{LWE}}$ is statistically close to its view in H_0.

Proof. In experiment H_0, $\mathbf{z}^{(s_1^*)} = (\mathbf{B}^{(s_1^*)})^\top \mathbf{x}^{(s_1^*)} + \mathbf{e}_0^{(s_1^*)}$ 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 $\mathbf{z} = \mathbf{B}^\top \mathbf{x} + \mathbf{e}$ 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 16, $\mathbf{z}^{(s_1^*)}$ 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 $\{\mathbf{x}^{(i^*)}, i^*\}_{i^* \in [N^*]}^{(s_1^*)}$ rather than $\{\mathbf{x}^{(i^*)}, i^*\}_{i^* \in [N^*]}^{(s_0^*)}$.

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. In LRS, the $\mathsf{vk}_{\mathsf{OTS}}$ consists of two parts, $\mathbf{A}_{\mathsf{com}}$ and $\mathbf{T}_{\mathbf{A}}$ i.e., $\mathsf{vk}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$. The goal of the adversary is to produce a new $\mathsf{vk}_{\mathsf{OTS}}^* \neq \mathsf{vk}_{\mathsf{OTS}}$. However, in order to achieve that, the adversary must generate a $\mathbf{T}_{\mathbf{F}^{(s)}}$ for some index s belong to the ring that the adversary provided, which contradicts the hardness of the SIS problem.

Reduction. Suppose the PPT adversary $\mathcal{A}$ has non-negligible advantage in breaking the signer-linkability by mounting the attack as defined in the linkability model of Definition 1 on LRS, then there exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ attacking the $\mathrm{SIS}_{q,n,m,\beta}$ problem. Consider the following security game between $\mathcal{A}$ and $\mathcal{S}$. Upon receiving a challenge $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that is formed by m uniformly random and independent samples from $\mathbb{Z}_q^n$, $\mathcal{S}$ simulates as follows.

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 $\mathcal{S}$ does as follows.

- Compute $(\mathbf{B}, \mathbf{T}_{\mathbf{B}}, \bar{\mathbf{R}}) \leftarrow \mathsf{SampleRwithBasis}(\mathbf{A}_{\mathsf{com}})$ such that $\mathbf{B} = \mathbf{A}_{\mathsf{com}}\bar{\mathbf{R}}^{-1}$.
- Let $\mathbf{A} = [\mathbf{a}_1 \mid \cdots \mid \mathbf{a}_m]$.
- For $i = 1$ to m:
  - Sample $\mathbf{r}_i' \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{T}_{\mathbf{B}}, \mathbf{a}_i, \sigma)$ such that $\mathbf{B}\mathbf{r}_i' = \mathbf{a}_i$.
  - Repeat the last step until $\mathbf{r}_i'$ is $\mathbb{Z}_q$ linearly independent of $\mathbf{r}_1, \ldots, \mathbf{r}_{i-1}$.

- Let $\mathbf{R}' \in \mathbb{Z}^{m\times m}$ be the matrix whose columns are $\mathbf{r}'_1, \ldots, \mathbf{r}'_m$.
- 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 $(\mathsf{R}_i^*, \boldsymbol{\mu}_i^*, \Sigma_i^*)$. It holds that $\mathsf{Ver}(\mathsf{R}_i^*, \boldsymbol{\mu}_i^*, \Sigma_i^*) = 1$ for $i \in [l]$, $\mathsf{Link}(\mathsf{R}_i^*, \boldsymbol{\mu}_i^*, \Sigma_i^*, \mathsf{R}_j^*, \boldsymbol{\mu}_j^*, \Sigma_j^*) = 0$ for any $i, j \in [l]$ s.t. $i \neq j$, and $|\cup_{i=1} \mathsf{R}_i^*| < l$, otherwise $\mathcal{S}$ aborts the simulation.

Infer that there must exist 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 $\mathsf{vk}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}}$ and $N^* = |\cup_{i=1} \mathsf{R}_i^*|$. There are two ways for $\mathcal{A}$ to produce the $\mathsf{vk}_{\mathsf{OTS}}^*$:

- $\mathcal{A}$ produces a $\mathbf{T}_{\mathbf{A}^*} \neq \mathbf{T}_{\mathbf{A}^{(s)}}$ such that $\mathsf{vk}_{\mathsf{OTS}}^* = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^*}$. In this case, existing an index $s \in [N^*]$ satisfy that, $\mathbf{F}^{(s)} = \left[\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^*} \mid \mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(s)}\right]$. Recall the $\mathsf{BasisExtBindOVK}$ algorithm, $\mathbf{F}^{(s)}$ has the basis $\mathbf{T}_{\mathbf{F}^{(s)}} = \begin{bmatrix} -\mathbf{R} & \mathbf{I}_m \\ \mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R} & -\mathbf{T}_{\mathbf{A}^{(s)}} \end{bmatrix}$ such that $\mathbf{F}^{(s)} \cdot \mathbf{T}_{\mathbf{F}^{(s)}} = \mathbf{0} \pmod q$. It holds that $\mathbf{A}_{\mathsf{com}}(\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R} - \mathbf{T}_{\mathbf{A}}^*\mathbf{R}) = \mathbf{0} \pmod q$. By the prior setting in Setup phase, we have $\mathbf{A}_{\mathsf{com}} = \bar{\mathbf{A}}\mathbf{R}'^{-1}\bar{\mathbf{R}}$, therefore, $\mathbf{R}'^{-1}\bar{\mathbf{R}}(\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R} - \mathbf{T}_{\mathbf{A}}^*\mathbf{R})$ will be a valid SIS solution.
- $\mathcal{A}$ produces a $\mathbf{A}_{\mathsf{com}}^* \neq \mathbf{A}_{\mathsf{com}}$ such that $\mathsf{vk}_{\mathsf{OTS}}^* = \mathbf{A}_{\mathsf{com}}^*\mathbf{T}_{\mathbf{A}}$. However, in the phase of verifying the one-time signature, it is required that the equation $\mathbf{A}_{\mathsf{com}}^*\Sigma_{\mathsf{OTS}} = \mathsf{vk}_{\mathsf{OTS}}^*\boldsymbol{\mu}$ holds. Therefore, $\mathcal{A}$ still needs to compute a $\mathbf{T}_{\mathbf{A}^*}$ such that $\mathbf{A}_{\mathsf{com}}^*\mathbf{T}_{\mathbf{A}^*}\boldsymbol{\mu} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}\boldsymbol{\mu}$ holds. In this case, $\mathbf{F}^{(s)} = \left[\mathbf{A}_{\mathsf{com}}^*\mathbf{T}_{\mathbf{A}^*} \mid \mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(s)}\right]$ has the basis $\mathbf{T}_{\mathbf{F}^{(s)}} = \begin{bmatrix} -\mathbf{R} & \mathbf{I}_m \\ \mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R} & -\mathbf{T}_{\mathbf{A}^{(s)}} \end{bmatrix}$. It holds that $\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} = \mathbf{0} \pmod q$. By the prior setting in Setup phase, $\mathbf{R}'^{-1}\bar{\mathbf{R}}\mathbf{T}_{\mathbf{A}^{(s)}}$ will be a valid SIS solution.

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

Proof. We argue the $\mathbf{R}'^{-1}\bar{\mathbf{R}}(\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R} - \mathbf{T}_{\mathbf{A}}^*\mathbf{R})$ and $\mathbf{R}'^{-1}\bar{\mathbf{R}}\mathbf{T}_{\mathbf{A}^{(s)}}$ are valid $\mathsf{SIS}_{q,n,m,\beta}$ solution in two steps. We first explain they are sufficiently short. By Lemma 11, the Gram-Schmidt norm of $\mathbf{T}_{\mathbf{A}^{(s)}}$ and $\mathbf{T}_{\mathbf{A}}^*$ is bounded as $O(\log n \cdot \sqrt{nm \log q})$. By Lemma 17 and 12, the Gram-Schmidt norm of $\mathbf{R}$, $\mathbf{R}'$, and $\bar{\mathbf{R}}$ is bounded as $O(\sqrt{nm \log q}) \cdot \omega(\sqrt{\log m})$. Therefore, even the larger one i.e., $\mathbf{R}'^{-1}\bar{\mathbf{R}}(\mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R} - \mathbf{T}_{\mathbf{A}}^*\mathbf{R})$ whose Gram-Schmidt norm is bounded as $O(\log n \cdot (nm \log q)^2) \cdot (\omega(\sqrt{\log m}))^3$ which is less than the $\beta = O(\ell^{4c} \cdot m^{7/2}) \cdot \omega(\sqrt{\log m})$ as given in Sect. 4.3. Then we prove they are non-zero. The proof is similar with the non-zero proof in unforgeability proof (cf. Proof of Theorem 1). Let $\hat{\mathbf{R}} = \mathbf{R}'^{-1}\bar{\mathbf{R}}$. Observe that for the fix message $\boldsymbol{\mu}^*$ on which $\mathcal{A}$ made the additional signature, $\hat{\mathbf{R}}$ depends on the low-norm matrices $\mathbf{R}'$ and $\bar{\mathbf{R}}$. The only information about $\hat{\mathbf{R}}$ is from the public matrix $\mathbf{A}_{\mathsf{com}}$ in PP. So by the pigeonhole principle there is a (exponentially) large freedom to pick entries of $\hat{\mathbf{R}}$ which is compatible with $\mathcal{A}'s$ view. 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. For a $\hat{\mathsf{vk}}_{\mathsf{OTS}}$ in a honest signature tuple, the goal of the adversary is to produce a $\mathsf{vk}^*_{\mathsf{OTS}} = \hat{\mathsf{vk}}_{\mathsf{OTS}}$ in a forged signature tuple. Specifically, for a $\hat{\mathsf{vk}}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$, there are three ways to produce a new $\mathsf{vk}^*_{\mathsf{OTS}}$ s.t. $\mathsf{vk}^*_{\mathsf{OTS}} = \hat{\mathsf{vk}}_{\mathsf{OTS}}$, generate a new $\mathbf{A}^*_{\mathsf{com}}$ i.e., $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}^*_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$, $\mathbf{T}^*_{\mathbf{A}}$ i.e., $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}}$, or $\mathbf{A}^*_{\mathsf{com}}$ and $\mathbf{T}^*_{\mathbf{A}}$ i.e., $\mathsf{vk}^*_{\mathsf{OTS}} = \mathbf{A}^*_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}}$. However, in order to achieve that, the adversary must generate a $\mathbf{T}_{\mathbf{F}^{(s)}}$ for some index s belong to the ring that the adversary provided, which contradicts the hardness of the SIS problem.

Reduction. Suppose the PPT adversary $\mathcal{A}$ has non-negligible advantage in breaking the signer-linkability by mounting the attack as defined in the linkability model of Definition 1 on LRS, then there exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ attacking the $\mathrm{SIS}_{q,n,m,\beta}$ problem. Consider the following security game between $\mathcal{A}$ and $\mathcal{S}$. Upon receiving a challenge $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that is formed by m uniformly random and independent samples from $\mathbb{Z}_q^n$, $\mathcal{S}$ simulates as follows.

Setup Phase. As same as the **Setup Phase** (cf. proof of Theorem 1) of unforgeability proof except $\mathcal{S}$ additionally compute the $(\bar{\mathbf{R}}, \mathbf{R}')$ as in the **Setup Phase** of linkability proof (cf. proof of Theorem 3) before send $(\mathsf{PP}, \gamma_{\mathsf{st}})$ to $\mathcal{A}$.

Probing Phase. As same as the **Probing Phase** of the unforgeability proof (cf. proof of Theorem 1).

Output. $\mathcal{A}$ outputs two signature tuples $(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*)$ and $(\hat{\boldsymbol{\mu}}, \hat{\mathsf{R}}, \hat{\Sigma})$ where $\hat{\Sigma}$ is replied from $\mathsf{OSign}(\hat{\mu}, \hat{\mathsf{R}}, \hat{i})$ for $\hat{i} \in [N]$. Let $N^* = |\mathsf{R}^*|$. Check if $\mathsf{Ver}(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*) = 1$, $(\boldsymbol{\mu}^*, \mathsf{R}^*, \Sigma^*) \notin \mathsf{L}$, $(\hat{\boldsymbol{\mu}}, \hat{\mathsf{R}}, \hat{\Sigma}) \in \mathsf{L}$, $\mathsf{vk}^{(\hat{i})} \notin \mathsf{R}^*$, 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 $\hat{\mathsf{vk}}_{\mathsf{OTS}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$. There are three ways for $\mathcal{A}$ to produce a new $\mathsf{vk}^*_{\mathsf{OTS}}$ s.t. $\mathsf{vk}^*_{\mathsf{OTS}} = \hat{\mathsf{vk}}_{\mathsf{OTS}}$.

- $\mathcal{A}$ corrupts the $\mathbf{T}_{\mathbf{A}}$ and then computes a $\mathbf{A}^*_{\mathsf{com}}$ such that $\mathbf{A}^*_{\mathsf{com}}\mathbf{T}_{\mathbf{A}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$. In this case, existing an index $s \in [N^*]$ satisfy that, $\mathbf{F}^{(s)} = \left[\mathbf{A}^*_{\mathsf{com}}\mathbf{T}_{\mathbf{A}} \mid \mathbf{A}_{\mathsf{com}} + \mathbf{A}^{(s)}\right]$. By the $\mathsf{BasisExtBindOVK}$ algorithm, $\mathbf{F}^{(s)}$ has the basis $\mathbf{T}_{\mathbf{F}^{(s)}} = \begin{bmatrix} -\mathbf{R} & \mathbf{I}_m \\ \mathbf{T}_{\mathbf{A}^{(s)}}\mathbf{R} & -\mathbf{T}_{\mathbf{A}^{(s)}} \end{bmatrix}$. It holds that $\mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}^{(s)}} = \mathbf{0} \pmod q$. By the prior setting in Setup phase, we have $\mathbf{A}_{\mathsf{com}} = \mathbf{A}\mathbf{R}'^{-1}\bar{\mathbf{R}}$, therefore, $\mathbf{R}'^{-1}\bar{\mathbf{R}}\mathbf{T}_{\mathbf{A}^{(s)}}$ will be a valid SIS solution.
- $\mathcal{A}$ computes a basis $\mathbf{T}^*_{\mathbf{A}}$ such that $\mathbf{A}_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$. This case is as same as the first case of the linkability proof (cf. proof of Theorem 3).
- $\mathcal{A}$ selects a basis $\mathbf{T}^*_{\mathbf{A}}$ and then computes the $\mathbf{A}^*_{\mathsf{com}}$ such that $\mathbf{A}^*_{\mathsf{com}}\mathbf{T}^*_{\mathbf{A}} = \mathbf{A}_{\mathsf{com}}\mathbf{T}_{\mathbf{A}}$. This case is as same as the second case of the linkability proof (cf. proof of Theorem 3).

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

Proof. The proof is as same as the proof of claim in signer-linkable proof (cf. proof of Theorem 3).

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 553–572. Springer, Heidelberg (2010), https://doi.org/10.1007/978-3-642-13190-5_28
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6223, pp. 98–115. Springer, Heidelberg (2010), https://doi.org/10.1007/978-3-642-14623-7_6
3. Au, M.H., Chow, S.S.M., Susilo, W., Tsang, P.P.: Short linkable ring signatures revisited. In: Atzeni, A.S., Lioy, A. (eds.) Public Key Infrastructure, Third European PKI Workshop: Theory and Practice, EuroPKI 2006, Turin, Italy, June 19-20, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4043, pp. 101–115. Springer, Heidelberg (2006), https://doi.org/10.1007/11774716_9
4. Backes, M., Döttling, N., Hanzlik, L., Kluczniak, K., Schneider, J.: Ring signatures: Logarithmic-size, no setup - from standard assumptions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11478, pp. 281–311. Springer, Heidelberg (2019), https://doi.org/10.1007/978-3-030-17659-4_10
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993. pp. 62–73. ACM, New York (1993), https://doi.org/10.1145/168588.168596
6. Beullens, W., Katsumata, S., Pintore, F.: Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12492, pp. 464–492. Springer, Heidelberg (2020), https://doi.org/10.1007/978-3-030-64834-3_16
7. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer, Heidelberg (2011), https://doi.org/10.1007/978-3-642-25385-0_3
8. 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. (eds.)

Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 533–556. Springer, Heidelberg (2014), https://doi.org/10.1007/978-3-642-55220-5_30

9. Boyen, X., Haines, T.: Forward-secure linkable ring signatures from bilinear maps. Cryptogr. **2**(4), 35 (2018), https://doi.org/10.3390/cryptography2040035
10. Boyen, X., Li, Q.: Towards tightly secure lattice short signature and id-based encryption. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10032, pp. 404–434. Springer, Heidelberg (2016), https://doi.org/10.1007/978-3-662-53890-6_14
11. Brakerski, Z., Kalai, Y.T.: A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive: Report 2010/086, 2010 (2010)
12. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: Naor, M. (ed.) Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014. pp. 1–12. ACM, New York (2014), https://doi.org/10.1145/2554797.2554799
13. Branco, P., Döttling, N., Wohnig, S.: Universal ring signatures in the standard model. IACR Cryptol. ePrint Arch. p. 1265 (2022), https://eprint.iacr.org/2022/1265
14. Canetti, R., Chen, Y., Holmgren, J., Lombardi, A., Rothblum, G.N., Rothblum, R.D., Wichs, D.: Fiat-shamir: from practice to theory. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. pp. 1082–1090 (2019)
15. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM **51**(4), 557–594 (2004), https://doi.org/10.1145/1008731.1008734
16. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 523–552. Springer, Heidelberg (2010), https://doi.org/10.1007/978-3-642-13190-5_27
17. Chatterjee, R., Chung, K., Liang, X., Malavolta, G.: A note on the post-quantum security of (ring) signatures. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13178, pp. 407–436. Springer (2022), https://doi.org/10.1007/978-3-030-97131-1_14
18. Chatterjee, R., Garg, S., Hajiabadi, M., Khurana, D., Liang, X., Malavolta, G., Pandey, O., Shiehian, S.: Compact ring signatures from learning with errors. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 282–312. Springer, Heidelberg (2021), https://doi.org/10.1007/978-3-030-84242-0_11
19. Dodis, Y., Oliveira, R., Pietrzak, K.: On the generic insecurity of the full domain hash. In: Shoup, V. (ed.) Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-

18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3621, pp. 449–466. Springer, Heidelberg (2005), https://doi.org/10.1007/11535218_27
20. Eaton, E., Song, F.: A note on the instantiability of the quantum random oracle. In: Ding, J., Tillich, J. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12100, pp. 503–523. Springer (2020), https://doi.org/10.1007/978-3-030-44223-1_27
21. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008. pp. 197–206. ACM, New York (2008), https://doi.org/10.1145/1374376.1374407
22. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 75–92. Springer, Heidelberg (2013), https://doi.org/10.1007/978-3-642-40041-4_5
23. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6477, pp. 395–412. Springer, Heidelberg (2010), https://doi.org/10.1007/978-3-642-17373-8_23
24. Grilo, A.B., Hövelmanns, K., Hülsing, A., Majenz, C.: Tight adaptive reprogramming in the QROM. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 637–667. Springer, Heidelberg (2021), https://doi.org/10.1007/978-3-030-92062-3_22
25. Katz, J.: Digital Signatures. Springer (2010). https://doi.org/10.1007/978-0-387-27712-7, https://doi.org/10.1007/978-0-387-27712-7
26. Lai, Q., Liu, F., Wang, Z.: Almost tight security in lattices with polynomial moduli - prf, ibe, all-but-many ltf, and more. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12110, pp. 652–681. Springer, Heidelberg (2020), https://doi.org/10.1007/978-3-030-45374-9_22
27. Leurent, G., Nguyen, P.Q.: How risky is the random-oracle model? In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5677, pp. 445–464. Springer (2009), https://doi.org/10.1007/978-3-642-03356-8_26
28. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings. Lecture Notes in Computer Science, vol. 3108, pp. 325–335. Springer, Heidelberg (2004), https://doi.org/10.1007/978-3-540-27800-9_28

29. Liu, Z., Nguyen, K., Yang, G., Wang, H., Wong, D.S.: A lattice-based linkable ring signature supporting stealth addresses. In: Sako, K., Schneider, S.A., Ryan, P.Y.A. (eds.) Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11735, pp. 726–746. Springer (2019), https://doi.org/10.1007/978-3-030-29959-0_35
30. Lu, X., Au, M.H., Zhang, Z.: Raptor: A practical lattice-based (linkable) ring signature. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11464, pp. 110–130. Springer, Heidelberg (2019), https://doi.org/10.1007/978-3-030-21568-2_6
31. 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
32. Micciancio, D., Goldwasser, S.: Complexity of lattice problems - a cryptograhic perspective, The Kluwer international series in engineering and computer science, vol. 671. Springer, Heidelberg (2002)
33. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, pp. 700–718. Springer, Heidelberg (2012), https://doi.org/10.1007/978-3-642-29011-4_41
34. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007), https://doi.org/10.1137/S0097539705447360
35. Micciancio, D., Vadhan, S.P.: Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 282–298. Springer, Heidelberg (2003), https://doi.org/10.1007/978-3-540-45146-4_17
36. Park, S., Sealfon, A.: It wasn't me! - repudiability and claimability of ring signatures. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11694, pp. 159–190. Springer, Heidelberg (2019), https://doi.org/10.1007/978-3-030-26954-8_6
37. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11692, pp. 89–114. Springer, Heidelberg (2019), https://doi.org/10.1007/978-3-030-26948-7_4
38. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005. pp. 84–93. ACM, New York (2005), https://doi.org/10.1145/1060590.1060603

39. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2248, pp. 552–565. Springer, Heidelberg (2001), https://doi.org/10.1007/3-540-45682-1_32
40. SHA, N.: Standard: Permutation-based hash and extendable-output functions (draft fips pub 202) (2014)
41. Sokolov, A.A.: Lin2-xor lemma and log-size linkable ring signature. IACR Cryptol. ePrint Arch. p. 688 (2020), https://eprint.iacr.org/2020/688
42. 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: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10493, pp. 456–474. Springer (2017), https://doi.org/10.1007/978-3-319-66399-9_25
43. Torres, W.A.A., Kuchta, V., Steinfeld, R., Sakzad, A., Liu, J.K., Cheng, J.: Lattice ringct V2.0 with multiple input and multiple output wallets. In: Jang-Jaccard, J., Guo, F. (eds.) Information Security and Privacy - 24th Australasian Conference, ACISP 2019, Christchurch, New Zealand, July 3-5, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11547, pp. 156–175. Springer, Heidelberg (2019), https://doi.org/10.1007/978-3-030-21548-4_9
44. Torres, W.A.A., Steinfeld, R., Sakzad, A., Kuchta, V.: Post-quantum linkable ring signature enabling distributed authorised ring confidential transactions in blockchain. IACR Cryptol. ePrint Arch. p. 1121 (2020), https://eprint.iacr.org/2020/1121
45. Torres, W.A.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. (eds.) Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10946, pp. 558–576. Springer, Heidelberg (2018), https://doi.org/10.1007/978-3-319-93638-3_32
46. Tsang, P.P., Wei, V.K., Chan, T.K., Au, M.H., Liu, J.K., Wong, D.S.: Separable linkable threshold ring signatures. In: Canteaut, A., Viswanathan, K. (eds.) Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3348, pp. 384–398. Springer (2004), https://doi.org/10.1007/978-3-540-30556-9_30