Towards Practical Lattice-Based One-Time Linkable Ring Signatures

Carsten Baum[1]*, Huang Lin[2], and Sabine Oechsner[3]**

[1] Department of Computer Science, Bar-Ilan University
carsten.baum@biu.ac.il
[2] ASTRI Security Lab
linhuang@astri.org
[3] Department of Computer Science, Aarhus University
oechsner@cs.au.dk

Abstract. Ring signatures, as introduced by Rivest, Shamir, and Tauman (Asiacrypt '01), allow to generate a signature for a message on behalf of an ad-hoc set of parties. To sign a message, only the public keys must be known and these can be generated independently. It is furthermore not possible to identify the actual signer based on the signature. Ring signatures have recently gained attention due to their applicability in the construction of practical anonymous cryptocurrencies, where they are used to secure transactions while hiding the identity of the actual spender. To be applicable in that setting, ring signatures must allow to determine when a party signed multiple transactions, which is done using a property called linkability.
This work presents a linkable ring signature scheme constructed from a lattice-based collision-resistant hash function. We follow the idea of existing schemes which are secure based on the hardness of the discrete logarithm problem, but adapt and optimize ours to the lattice setting. In comparison to other designs for (lattice-based) linkable ring signatures, our approach avoids the standard solution for achieving linkability, which involves proofs about correct evaluation of a pseudorandom function using heavy zero-knowledge machinery.

Keywords: lattice-based cryptography, ring signature scheme, anonymous cryptocurrency

1 Introduction

Digital signatures are one of the most important concepts in the area of cryptography. They permit a party to generate a key pair (SK, PK), give PK to the

* Supported by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Ministers Office.

** This work has been supported by the European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 669255 (MPCPRO). Part of work done while visiting NTT Secure Platform Laboratories.

public and add certain information Ω - called the signature - to a message m. Ω is derived using the private key SK and later allows a verifier, equipped with the public verification key PK, to attest that the signer indeed generated Ω for this specific message m. Verification is done in a way such that only a party who possesses certain secret information that only the signer has, namely the secret signing key SK, can generate a valid signature for PK.

Ring signatures, which were first suggested by Rivest, Shamir, and Tauman [42], relax the condition of having exactly one pair (SK, PK) for signing and verification to a certain extent. They allow a party among a set of N participants to sign a message on behalf of all of them. Here it is crucial that the verifier cannot identify the party that signed the message, while nobody outside of the N participants should be able to sign a message as if he was a participant himself. In comparison to group signatures, the set of parties does not need to be known ahead of time, but only when the signature is generated. Therefore, no key-generation algorithm which generates correlated randomness for all N parties needs to be involved and the rings can be set up ad-hoc[4].

For such a ring signature, each signer could issue an arbitrary number of signatures. Fujisaki and Suzuki introduced the notion of traceable ring signatures [19], where the signer signs a message with respect to a list of ring members and a public issue such as an election. There is a public procedure to determine whether two signatures come from one signer, i.e., the signer is linked if a signer signs the same message with respect to the same list of ring member and same issue twice [18]. A related idea is so-called *linkable ring signatures*, in which case the true signer will be linked when he signs two messages (different or identical) with respect to the same ring. In a more restricted version of linkable ring signatures, *one-time* linkable ring signatures, a signer is linked as soon as he reveals two signatures. This property has proven to be vital in the construction of cryptocurrencies, such as to prevent double spending attacks and to preserve the anonymity of a spender since the address or the respective secret key in the design of the anonymous cryptocurrency is supposed to be one-time [39].

1.1 Related work

Lattice-based signature schemes. The line of work on lattice-based signature schemes was, to the best of our knowledge, initiated by Goldreich et al. [21], while the first practical construction was based on NTRU [25, 24]. A scheme that fits into this line of work is the provably secure construction due to Gentry et al., also called hash-and-sign [20]. This approach, where the signing key is a secret trapdoor which is used to sample a short lattice vector, was further developed in [8, 17]. A different direction, called Fiat Shamir with Aborts, was first explored by Lyubashevsky [32, 31]. Recently, very efficient signature schemes such as Bliss [15], Tesla [23] and Dilithium [16] have been designed within this framework.

[4] We relax this a bit and assume that there exists a CRS which is known to all parties and which allows them to derive their respective key pairs (SK, PK).

(Linkable) ring signature schemes. There exists a wealth of literature on ring signature and linkable ring signature schemes such as [42, 18, 30, 5, 19] and we only list some of the relevant works here. However, the above mentioned signature schemes have a signature size that is linearly dependent on the number of users N in the ring. Chandran, Groth, and Sahai [10] construct a ring signature scheme with signature size $O(\sqrt{N})$ based on composite order groups with a bilinear map. The Groth-Kohlweiss framework [22] is based on homomorphic commitments and provides a ring signature scheme with a logarithmic signature size. Franklin and Zhang [18] propose a general framework for linkable ring signatures. They extend the"PRF made public" paradigm by Bellare and Goldwasser [4] in order to provide linkability by combining a PRF evaluation of the secret key with a NIZK proof of correct evaluation. The smallest ring signatures to date have constant signature size and are based on accumulators. The construction by Dodis et al. [14] uses accumulators based on the strong RSA assumption, while Nguyen's [38] relies on pairing-based cryptography. There exists also a linkable version of [14] by Tsang and Wei [43] that retains the constant-sized signatures.

Lattice-based ring signature schemes. Lattice-based ring signatures were first introduced explicitly through the work of Brakerski and Tauman-Kalai [9] who proposed a general framework for ring signatures in the standard model and showed how to instantiate it based on the SIS assumption. The resulting signatures have size $O(mN)$ for message length m and ring size N. Subsequently, Wang and Sun [45] proposed two ring signatures schemes from the SIS assumption in the random oracle and standard model, respectively, both of linear signature size. The first ring signature scheme based on the LWE assumption was proposed by Melchor et al. [36] and is an extension of [31] to the ring signature setting. Like the previous schemes, it yields signatures of linear size. Furthermore, the anonymous identification scheme of Kawachi et al. [26] based on the SIS-problem can be turned into a ring signature scheme with linear-size (in the ring size) signatures using standard techniques. Recently, Libert et al. [28] proposed the first lattice-based ring signature scheme with only logarithmic signature size using a Merkle-tree based construction.

1.2 Our contribution

We present a lattice-based ring signature scheme based on the Module-SIS and Module-LWE problem. Our scheme has a signature size which is linear in N, which is asymptotically less efficient than the construction of Libert et al. [28]. However, [28] achieves sublinear size at the cost of huge hidden constants and is thus far from practical, even without linkability. Moreover, for applications such as cryptocurrencies [44], linear-sized signatures might in practice be less of a concern: When forming an ad-hoc ring, one must still publish the public keys of all the users in the ring since the verification algorithms of these constructions [22, 28] both contain critical steps involving this public key list, and hence still results in linear space and computational cost for the real-world implementation.

Thus, what matters in practice is the sum of the size of the signature and the public keys.

There exist other candidates for post-quantum ring signature schemes such as hash-based [13] or multi-variate-quadratic-equation based schemes [37] although neither of them provides linkability. Compared to those candidates, lattice-based signatures on average have the advantage of smaller combined signature and public key size [1]. Hash-based signatures, in addition, have the practical intricacy of state management [6] since one has to remember which secret key has been used to avoid reusing it in a stateful hash-based signature scheme. Whether or not this feature is compatible with the anonymity requirement of a cryptoccurency [44] is unknown. The stateless hash-based signature on the other hand usually has a much larger signature size [6] compared with the stateful version.

1.3 Technical Overview

As mentioned before, the standard approach for transforming a ring signature scheme into a linkable ring signature scheme, following Franklin and Zhang [18], is to add a PRF evaluation of the signer's secret key to the signature, as well as a zero-knowledge proof of correct evaluation of the PRF under one of the secret keys corresponding to the public keys. This generic approach applies to any ring signature scheme and was explored for lattice-based PRFs in [29, 28, 46]. However, such proofs come with quite a substantial overhead. Our construction instead follows the approach of Liu et al. [30] that avoids this technique. The main observation is that the signer in their scheme has two "public" keys: One that is published before signature generation as part of the ring of signers, and the other one that is appended to each signature. Hence, another "public key" under different public parameters that corresponds to the signer's secret signing key can be used as linkability tag. Since both kinds of public keys share the same algebraic structure, the two "public keys" of the signer, i.e. the actual public key and the linkability tag, can be tied together without appending another non-interactive zero-knowledge proof to the signature.

Since our construction will be based on the Module-SIS problem, the public keys of the parties are of the form $PK = \boldsymbol{A}\boldsymbol{r}$ for secret key $\boldsymbol{r}$ and public matrix $\boldsymbol{A}$. Linkability will be ensured by providing linkability tags $I = \boldsymbol{B}\boldsymbol{r}$ for another public matrix $\boldsymbol{B}$. Interestingly, the reason why our construction achieves only one-time linkability is inherent in this approach: any evaluation $\boldsymbol{B}\boldsymbol{r}$ leaks information about $\boldsymbol{r}$. If a fresh matrix $\boldsymbol{B}$ is generated for each ring, then a malicious party can receive more leakage on $\boldsymbol{r}$ than intended and hence may be able to recover the signer's secret key.

In order to obtain more efficient lattice-based (linkable) ring signatures, it may be tempting to try to instantiate current sublinear-size ring signatures in the lattice setting. Note, however, that this is far from trivial, as these solutions are specifically tailored to a certain assumption like Dodis et al.'s accumulator-based ring signatures [14], or suffer from the well-known problem that hard lattice assumptions do not provide enough algebraic structure to support existing

sublinear approaches based on homomorphic operations like that of Groth and Kohlweiss [22].

It is an interesting open problem to find practical lattice-based linkable ring signatures that allows for more nuanced linkability and/or that have sublinear size.

2 Preliminaries

We will use $[N]$ as shorthand for the set $\{1, \ldots, N\}$. Let R be the cyclotomic ring $R = \mathbb{Z}[X]/\langle X^\nu + 1\rangle$, where $\nu = 2^p$ and $p \in \mathbb{N}^+$. Let q be an odd prime and define $R_q = \mathbb{Z}_q[X]/\langle X^\nu + 1\rangle$. Here $\mathbb{Z}_q$ denotes the integers modulo q, which will be represented as elements from the interval $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$. For $f = \sum_i f_i X^i \in R$, the norms of f are defined as

$$l_1 : \|f\|_1 = \sum\nolimits_i |f_i|,\ l_2 : \|f\|_2 = \left(\sum\nolimits_i |f_i|^2\right)^{1/2},\ l_\infty : \|f\|_\infty = \max_i |f_i|\,.$$

If $f \in R_q$, then we will represent each coset from $\mathbb{Z}_q$ with its unique representative from the aforementioned interval and consider the norm of the obtained $\mathbb{Z}$-vector. Let S_β denote the set of elements $x \in R$ with l_∞-norm at most β. Let $\mathbf{0}_v \in \mathbb{Z}^{v\times v}$ and $\mathbf{I}_v \in \mathbb{Z}^{v\times v}$ denote the zero and identity matrix over $\mathbb{Z}$ of dimension $v \times v$.

Remark 1. We use the following standard relations among different l-norms of a vector:

1. Let R be defined as above, $f, g \in R$ such that $\|f\|_\infty \leq \beta$, $\|g\|_1 \leq \gamma$, then $\|fg\|_\infty \leq \beta\gamma$.
2. If $f \in R$, $\boldsymbol{g} \in R^v$ satisfy that $\|f\|_2 \leq \beta$, $\|\boldsymbol{g}\|_\infty \leq \gamma$, then $\|f\boldsymbol{g}\|_2 \leq \sqrt{v}\nu\beta\gamma$.

We require a subset D of R_q which consists of short invertible elements such that the difference of any two distinct elements from this set is also invertible. It was shown in [35] that as long as q is a prime that satisfies $q = 17 \mod 32$ and $q > 2^{20}$, then the set $D = \{d \in R_q | \|d\|_\infty \leq 1, \|d\|_1 \leq \kappa\}$ satisfies this requirement. We use $\bar{D}$ to denote the set of values $D + D$ excluding 0. By setting $\kappa = 60$, the set D, for all practical instantiations, has size at least 2^{256}.

2.1 Normal Distribution and Rejection Sampling

The continuous normal distribution over $\mathbb{R}^\nu$ centered at $\boldsymbol{u} \in \mathbb{R}^\nu$ with standard deviation σ has probability density function

$$\rho^\nu_{\boldsymbol{u},\sigma}(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(\frac{-||\boldsymbol{x} - \boldsymbol{u}||_2^2}{2\sigma^2}\right)$$

The *discrete normal distribution* over R^v centered at $\boldsymbol{u} \in R^v$ with standard deviation σ is given by the distribution function (for all $\boldsymbol{x} \in R^v$)

$$\mathcal{N}_{\boldsymbol{u},\sigma}(\boldsymbol{x}) = \rho^{v\cdot\nu}_{\boldsymbol{u},\sigma}(\boldsymbol{x})/\rho^{v\cdot\nu}_\sigma(R^v),$$

where we omit the subscript $\boldsymbol{u}$ when it is zero. We use the following tail-bound due to Banaszczyk [3]:

Lemma 1. *Let $\mathcal{N}_{\boldsymbol{u},\sigma}$ be defined as above. Then*

$$\Pr\left[\|\boldsymbol{z}\|_2 > 2\sigma\sqrt{v\nu} \mid \boldsymbol{z} \leftarrow \mathcal{N}_{\sigma}^{v}\right] < 2^{-v\nu}$$

For our ring signature scheme, we use rejection sampling to hide the secret signing key. The basic idea of rejection sampling is to abort the protocol with a certain probability such that the distribution of the response is independent of the secret input. We adopt the rejection sampling lemma from [32]:

Lemma 2. *Let V be a subset of R^v in such that all elements have $\|\cdot\|_2$-norms less than T, $\sigma \in \mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log(v\nu)})$, and $h : V \to \mathbb{R}$ be a probability distribution. Then there exists an $M = O(1)$ such that the output distribution of the following two algorithms $\mathcal{A}$, $\mathcal{S}$ is within statistical distance $2^{-\omega(\log(v\nu))}/M$:*

$\mathcal{A}$**:**

1. $\boldsymbol{u} \leftarrow h$
2. $\boldsymbol{z} \leftarrow \mathcal{N}_{\boldsymbol{u},\sigma}^{v}$
3. *output* $(\boldsymbol{u}, \boldsymbol{z})$ *with probability* $\min\left(\frac{1}{M}\frac{\mathcal{N}_{\sigma}^{v}(\boldsymbol{z})}{\mathcal{N}_{\boldsymbol{u},\sigma}^{v}(\boldsymbol{z})}, 1\right)$

$\mathcal{S}$**:**

1. $\boldsymbol{u} \leftarrow h$
2. $\boldsymbol{z} \leftarrow \mathcal{N}_{\sigma}^{v}$
3. *output* $(\boldsymbol{u}, \boldsymbol{z})$ *with probability* $1/M$

Moreover, the probability that $\mathcal{A}$ outputs a value is at least $\frac{1-2^{-\omega(\log(v\nu))}}{M}$.

2.2 Module-SIS and Module-LWE

The security of our linkable ring signature scheme will be based on the hardness of two problems, Module-SIS and Module-LWE [27]. These problems are variants of the well-known SIS [2] and LWE [41] problems, but over modules that are defined over polynomial rings. This is a generalized version of the Ring-SIS and Ring-LWE problems [34, 33, 40].

The homogeneous Module-SIS problem consists of finding a vector $\boldsymbol{r}$ of small norm such that $\boldsymbol{A}\boldsymbol{r} = 0$ for a given, structured matrix $\boldsymbol{A}$.

Definition 1. $(MSIS_{h,v,\beta})$ *Given $\boldsymbol{A} \leftarrow R_q^{h\times v}$, find $\boldsymbol{r} \in R^v$ such that $\boldsymbol{A}\boldsymbol{r} = 0$ and $0 < \|\boldsymbol{r}\|_2 \leq \beta$.*

Our scheme also uses the Decisional Module-LWE problem. In $D\text{-}MLWE$, the problem consists of distinguishing noisy linear equations from random.

Definition 2. $(D\text{-}MLWE_{h,v,\beta})$ *Let $\boldsymbol{A} \leftarrow R_q^{h\times v}$. Then distinguish the distributions*

$$(\boldsymbol{A}, \boldsymbol{A}\boldsymbol{r}) \quad \textit{and} \quad (\boldsymbol{A}, \boldsymbol{u})$$

where $\boldsymbol{r} \leftarrow S_{\beta}^{v}$ and $\boldsymbol{u} \leftarrow R_q^h$.

Here, we use a special instance of the Module-LWE problem where the secret has the same distribution as the noise[5].

If two samples (with different matrices, but same secret vector $\boldsymbol{r}$) are issued by the challenger, then this can still be related to a $D\text{-}MLWE$ instance but with different parameters, as the following proposition shows.

Proposition 1. *Let* $\boldsymbol{A}, \boldsymbol{B} \leftarrow R_q^{h \times v}$, $\boldsymbol{r} \leftarrow S_\beta^v$ *and* $\boldsymbol{c}, \boldsymbol{d} \leftarrow R_q^h$. *Then*

$$(\boldsymbol{A}, \boldsymbol{Ar}, \boldsymbol{B}, \boldsymbol{Br}) \approx_c (\boldsymbol{A}, \boldsymbol{c}, \boldsymbol{B}, \boldsymbol{d})$$

given the $D\text{-}MLWE_{2h,v,\beta}$*-problem is hard.*

Proof. Consider the matrices $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \end{bmatrix}$, and $\boldsymbol{Er} = \begin{bmatrix} \boldsymbol{Ar} \\ \boldsymbol{Br} \end{bmatrix}$. Then distinguishing the above distributions is equivalent to distinguishing

$$(\boldsymbol{E}, \boldsymbol{Er}) \approx_c \left(\boldsymbol{E}, \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} \right)$$

This is the definition of the $D\text{-}MLWE_{2h,v,\beta}$ problem. □

Our construction will moreover rely on a third problem, which is related to $D\text{-}MLWE$ but is not as well-known, namely the Search Module-LWE problem. It can be seen as an inhomogeneous $MSIS$ instance where the target is known to have a short preimage under $\boldsymbol{A}$.

Definition 3. ($S\text{-}MLWE_{h,v,\beta}$) *Sample* $\boldsymbol{r} \leftarrow S_\beta^v$ *uniformly at random. Given* $\left(\boldsymbol{A} \leftarrow R_q^{h \times v}, \boldsymbol{s} = \boldsymbol{Ar}\right)$ *find* $\boldsymbol{r'} \in R^v$ *such that* $\boldsymbol{Ar'} = \boldsymbol{s}$ *and* $0 < \|\boldsymbol{r'}\|_\infty \leq \beta$.

Langlois and Stehlé [27] showed that, for certain parameter sets, $S\text{-}MLWE$ can be reduced to $D\text{-}MLWE$.

2.3 Linkable Ring Signatures

The formal syntax and security model of linkable ring signatures, sometimes also called linkable spontaneous anonymous group signatures, can be found in [30, 19]. Definitions of linkable ring signatures with adaptation to the cryptocurrency scenario can be found in [39]. Our definitions are in the spirit of [22, 30, 19].

Definition 4 (Linkable Ring Signature). *A linkable ring signature scheme consists of five algorithms:*

[5] This equivalent formulation is possible in our setting, as only one LWE sample will be issued per secret. The definition might seem unusual at first, as one regularly defines the LWE distribution as $\boldsymbol{As}_1 + \boldsymbol{s}_2$. We can use the following transformation, which is well-known: note that the given equation is equivalent to writing $\boldsymbol{As}_1 + \boldsymbol{I}_h \boldsymbol{s}_2$ instead. By aligning this into a single matrix product of $\boldsymbol{A'}$ with $(\boldsymbol{s}_1 | \boldsymbol{s}_2)$ and multiplying the resulting challenge with a uniformly random $r \in R_q$, we obtain Definition 2.

- **Setup**(1^λ)*: Generates and outputs public parameters PP available to all users.*
- **KGen**(PP)*: Generates a public verification key PK and a private signing key SK.*
- $\mathbf{Sign}_{PP,SK_\ell}(m, L)$*: Outputs a signature* Ω *on the message* $m \in \{0,1\}^\star$ *with respect to the ring* $L = (PK_1, \ldots, PK_N)$*. Here,* (PK_ℓ, SK_ℓ) *is a valid key pair output by* **KGen**(PP)*, and* $PK_\ell \in L$*.*
- **Vfy**(m, L, Ω)*: Verifies a purported ring signature* Ω *on a message* m *with respect to the ring of public keys* L*. It outputs a bit* $b \in \{0,1\}$*.*
- **Link**$(m_1, m_2, \Omega_1, \Omega_2)$[6]*: Takes as inputs two messages* m_1, m_2 *as well as two signatures* Ω_1 *and* Ω_2 *and outputs* $b \in \{0,1\}$*.*

The above algorithms form a linkable ring signature scheme if the following three definitions of correctness, signer anonymity, linkability and exculpability are fulfilled.

Definition 5 (Correctness). *Let* $N \geq 1$*. Then* $\forall t \in [N],\ \forall \{i_1, ..., i_t\} \subset [N]$*,* $k \in \{i_1, ..., i_t\}$ *and* $\forall m \in \{0,1\}^*$ *it holds that*

$$\Pr\left[\mathbf{Vfy}(m, L, \Omega) = 0 \;\middle|\; \begin{array}{l} PP \leftarrow \mathbf{Setup}() \\ \wedge \{PK_i \leftarrow \mathbf{KGen}(PP)\}_{i\in[N]} \\ \wedge L = (PK_{i_1}, \ldots, PK_{i_t}) \\ \wedge \Omega = \mathbf{Sign}_{PP,SK_k}(m, L) \end{array}\right] \leq \mathrm{negl}(\lambda)$$

Signer anonymity captures the intuition that if the targeted signer is not corrupted, then the probability that the adversary can identify him as the true signer among all uncorrupted parties is negligible.

Definition 6 (Signer Anonymity). *Let* $L = (PK_1, \ldots, PK_N)$ *be a list of public keys and* D_t *be any set of* $0 \leq t < N$ *signing keys such that* $\forall SK_i \in D_t\ \exists PK_i \in L :\ (PK_i, SK_i)$ *is generated by* **KGen***. A ring signature scheme is signer anonymous if for any PPT algorithm* $\mathcal{E}$*, on inputs of any message* m*, sets* L, D_t *as defined above and any valid signature* Ω *on* L *and* m *generated using* $SK_\ell \notin D_t$*, then*

$$\left|\Pr\left[\mathcal{E}\left(m, L, D_t, \Omega\right) = \ell\right] - \frac{1}{N-t}\right| = \mathrm{negl}(\lambda).$$

Let $PP \leftarrow \mathbf{Setup}(1^\lambda)$. For the following two definitions we assume the existence of two oracles $\mathcal{O}_K, \mathcal{O}_S$:

Key generation oracle $\mathcal{O}_K$: On input of a bit b it generates a random keypair $(PK, SK) \leftarrow \mathbf{KGen}(PP)$. If $b = 0$ then it outputs PK, otherwise (PK, SK).

[6] Different from the definition of **Link** algorithm in the existing linkable ring signature schemes [30, 19], our definition does not take L as inputs since we are talking about one-time linkable ring signature.

Signing oracle $\mathcal{O}_S$: On input (L, m, i) where $L = (PK_1, \ldots, PK_N)$ is a list of public keys generated by $\mathcal{O}_K$, $i \in [N]$ and $\mathcal{O}_K$ did not output SK_i and $m \in \{0,1\}^*$, it outputs $\Omega \leftarrow \mathbf{Sign}_{PP,SK_i}(m, L)$. If a key in L was not queried before, then it outputs $\perp$.

The idea behind the Linkability definition is as follows: if the same signer generates two signatures, then the algorithm **Link** will identify this with overwhelming probability. It is important that this not only holds against honest use of the algorithm **Sign**, but arbitrary adversaries.

Definition 7 (Linkability). *Let $\mathcal{A}$ be a PPT algorithm with oracle access to $\mathcal{O}_K, \mathcal{O}_S$. $\mathcal{A}$ is given 1^λ and PP as input and outputs a list $L \subseteq \overline{L}$ (where $\overline{L}$ is the set of all keys queried from $\mathcal{O}_K$) of length N together with $N+1$ values $\{(m_i, \Omega_i)\}_{i \in [N+1]}$. Then the scheme is linkable if, for every such $\mathcal{A}$,*

$$\Pr\begin{bmatrix}\forall i \in [N+1]\, \mathbf{Vfy}(m_i, L, \Omega_i) = 1 \wedge \\ \forall i, j \in [N+1],\, i \neq j,\, \mathbf{Link}(m_i, m_j, \Omega_i, \Omega_j) = 0\end{bmatrix} \leq \mathrm{negl}(\lambda).$$

The above only talks about the setting of generating signatures without being traceable. Equally important is the setting where signatures are signed by two different parties, where we require that their tags must be distinct. This then, of course, in particular includes the case of the **Sign** algorithm. This property is important in the setting of cryptocurrencies where one might otherwise be able to issue fake transactions on behalf of another party.

Definition 8 (Exculpability). *Let $\mathcal{A}$ be a PPT algorithm with oracle access to $\mathcal{O}_K, \mathcal{O}_S$. $\mathcal{A}$ is given 1^λ and PP as input and outputs a list $L \subseteq \overline{L}$ (where $\overline{L}$ is the set of all keys queried from $\mathcal{O}_K$) of length N together with two pairs $(m_1, \Omega_1), (m_2, \Omega_2)$ with $\mathbf{Vfy}(m_1, L, \Omega_1) = \mathbf{Vfy}(m_2, L, \Omega_2) = 1$, not both queried to $\mathcal{O}_S$. Let $M \subset L$ be set of PK_i for which $\mathcal{A}$ did not obtain SK_i from $\mathcal{O}_K$. Then*

$$\Pr\left[\mathbf{Link}(L, m_1, m_2, \Omega_1, \Omega_2) = 1 \,\middle|\, \begin{array}{l}\exists PK_i \in M\, \exists m \in \{0,1\}^* \\ \exists j \in \{1,2\} : \\ \left[\Omega \leftarrow \mathbf{Sign}_{PP,SK_i}(m, L)\right. \\ \left.\wedge \mathbf{Link}(m, m_j, \Omega, \Omega_j) = 1\right]\end{array}\right] \leq \mathrm{negl}(\lambda).$$

Remark 2. In our scheme, we do not give a definition and proof for existential unforgeability. As was observed in [19] the above definitions imply this property, as any algorithm breaking existential unforgeability can be used in a black-box setting to break exculpability (see [19, Theorem 2.6]).

3 Constructing Linkable Ring Signatures

In this section, we will describe our linkable ring signature scheme and prove its security. Our proposed scheme can be considered as an adaption of the linkable

ring signature scheme proposed in [30] to the lattice setting. However, while most linkable signature schemes such as the one proposed in [18] require the use of a pseudorandom function to achieve linkability, our scheme demonstrates that the linkability for one-time ring signature schemes can be obtained without using a pseudorandom function to generate the tag.

If a scheme is not one-time, then this PRF is evaluated on the secret (or public) key of the signing party and a description of the actual ring L. In our case, it is not necessary to include the ring L into the tag computation (as the scheme is one-time) and we attach a tag derived from the secret key only. Concretely, each party will have a private key $\boldsymbol{r}_i$ together with a public key $PK_i = \boldsymbol{A}\boldsymbol{r}_i$, where $\boldsymbol{A}$ is a random length-compressing matrix and $\boldsymbol{r}_i$ is a vector of small norm. Thus, PK_i is an evaluation of the public collision-resistant hash function $f_{\boldsymbol{A}}(\cdot) : \boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$ on the private input $\boldsymbol{r}_i$.

During the signing process, the signer will generate two rings of signatures (similar to [42, 30] but twice): the first is a ring consisting of signatures for all the N public keys and generated using $f_{\boldsymbol{A}}$ whereas the second ring uses a different CRHF $f_{\boldsymbol{B}}$. This function $f_{\boldsymbol{B}}(\cdot) : \boldsymbol{x} \mapsto \boldsymbol{B}\boldsymbol{x}$ uses a different public matrix $\boldsymbol{B}$ having the same dimensions as $\boldsymbol{A}$. The crucial point to interleave these rings is that they are built simultaneously, using the same challenges and blinding value in each step. For this to be verifiable, the signer must now include his I_i in the signature, which serves the same purpose as the public key PK_i in the first ring. We will show that the signer is bound to use his own value I_i if he wants to generate a valid signature and will therefore produce a collision if a second signature is revealed.

Let $H : \{0,1\}^* \to D$ be a cryptographic hash function where D is the challenge space defined in Section 2. The algorithms of our scheme are defined as follows:

Setup(1^λ)**:** Sample two random matrices $\boldsymbol{A}, \boldsymbol{B} \leftarrow R_q^{h \times v}$ and set $PP = (\boldsymbol{A}, \boldsymbol{B})$.

KGen(PP) Sample $\boldsymbol{r} \leftarrow S_\beta^v$ and then generate the public key $PK = \boldsymbol{A}\boldsymbol{r}$ as well as the signing key $SK = \boldsymbol{r}$.

Sign$_{PP,SK_\ell}(m, L)$**:**

1. Compute the tag $I_\ell = \boldsymbol{B}\boldsymbol{r}_\ell$.
2. Sample $\boldsymbol{u} \leftarrow \mathcal{N}_\sigma^v$ and set $d_{\ell+1} \leftarrow H(L, I_\ell, m, \boldsymbol{A}\boldsymbol{u}, \boldsymbol{B}\boldsymbol{u})$.
3. For each $i = \ell+1, \ldots, N, 1, \ldots, \ell-1$:
   (a) Sample $\boldsymbol{r}_{z,i} \leftarrow \mathcal{N}_\sigma^v$.
   (b) Set $t_{i,1} = \boldsymbol{A}\boldsymbol{r}_{z,i} - d_i PK_i$ and $t_{i,2} = \boldsymbol{B}\boldsymbol{r}_{z,i} - d_i I_\ell$ as well as $d_{(i \mod N)+1} \leftarrow H(L, I_\ell, m, t_{i,1}, t_{i,2})$.
4. Compute $\boldsymbol{r}_{z,\ell} = \boldsymbol{u} + d_\ell \boldsymbol{r}_\ell$.
5. Abort with probability $1 - \min\left(1, \frac{\mathcal{N}_\sigma^v(\boldsymbol{r}_{z,\ell})}{M \cdot \mathcal{N}_{d_\ell \boldsymbol{r}_\ell, \sigma}^v(\boldsymbol{r}_{z,\ell})}\right)$, otherwise output the signature $\Omega = \left(d_1, (\boldsymbol{r}_{z,i})_{i \in [N]}, I_\ell\right)$.

Vfy(m, L, Ω)**:**

1. For $i \in [N]$, check whether $\|\boldsymbol{r}_{z,i}\|_2 \leq 2\sigma\sqrt{\nu v}$, else output 0.
2. For $i \in [N]$, compute $t'_{i,1} = \boldsymbol{A}\boldsymbol{r}_{z,i} - d_i PK_i$, $t'_{i,2} = \boldsymbol{B}\boldsymbol{r}_{z,i} - d_i I_\ell$ as well as $d_{i+1} = H\left(L, I_\ell, m, t'_{i,1}, t'_{i,2}\right)$.
3. If $d_1 = H\left(L, I_\ell, m, t'_{N,1}, t'_{N,2}\right) = d_{N+1}$ then output 1, else output 0.

Link(Ω_1, Ω_2)**:** Given

$$\Omega_1 = \left(d_1^{(1)}, \left(\boldsymbol{r}_{z,i}^{(1)}\right)_{i\in[N]}, I_\ell^{(1)}\right) \text{ and } \Omega_2 = \left(d_1^{(2)}, \left(\boldsymbol{r}_{z,i}^{(2)}\right)_{i\in[N]}, I_\ell^{(2)}\right),$$

return 1 if $I_\ell^{(1)} = I_\ell^{(2)}$ and 0 otherwise.

Correctness can easily be verified using Lemma 1 and Lemma 2 and the proof is left as an exercise to the reader.

3.1 Security

We first provide a theorem about signer anonymity. As its proof follows a similar structure as in [30, 42], it is moved to the appendix.

Theorem 1 (Signer Anonymity). *The proposed ring signature scheme provides signer anonymity in the (programmable) random oracle model assuming hardness of the $D\text{-}MLWE_{2h,v,\beta}$-problem.*

Proof. See Appendix A.

The proofs of Linkability and Exculpability use the Forking Lemma. Due to space limitations, we separate the rewinding part (as it also rather directly follows from previous work) from the rest and defer it to the appendix. Instead, for the proof of Linkability, we now give an exact description of what we need in the following lemma.

Lemma 3. *Let $\mathcal{A}$ be a PPT algorithm with oracle access to $\mathcal{O}_K, \mathcal{O}_S, H$ which queries H at most q_H times and $\mathcal{O}_S$ at most q_S times. Moreover, $\mathcal{A}$ on input $1^\lambda, PP$ outputs $L, \{(m^{(i)}, \Omega^{(i)})\}_{i\in[N+1]}$ as defined in Definition 7 with probability ϵ in time at most s. Then there exists an algorithm $\mathcal{R}$ that outputs $(I, \boldsymbol{r}, \hat{\boldsymbol{r}}, d, \hat{d}, \pi)$*

- *such that $\pi \in [N]$, $d, \hat{d} \in D$ and $||\boldsymbol{r}||, ||\hat{\boldsymbol{r}}|| \leq 2\sigma\sqrt{\nu v}$,*
- *it holds that $dPK_\pi = \boldsymbol{A}\boldsymbol{r}$, $\hat{d}PK_\pi = \boldsymbol{A}\hat{\boldsymbol{r}}$, $dI = \boldsymbol{B}\boldsymbol{r}$ and $\hat{d}I = \boldsymbol{B}\hat{\boldsymbol{r}}$,*
- *where I is not a tag for any honestly generated PK_j from L,*
- *with probability $\left(\epsilon - \frac{1}{|D|-q_H-Nq_S}\right)^2 / \left((N^2+N)q_H\right)^2$,*
- *in time $O(N^2 \cdot q_H \cdot s)$.*

Proof. See Appendix A.

Using the above lemma, the proof of Linkability now works as follows: we embed a $MSIS$-instance into the public key and then use the algorithm $\mathcal{R}$. This gives us values $d, \hat{d}, \boldsymbol{r}, \hat{\boldsymbol{r}}$ which are used to obtain a $MSIS$-preimage. It remains to show that this preimage is within the bounds of the theorem and non-zero.

Theorem 2 (Linkability). *Let $\mathcal{A}$ be defined as in Lemma 3. Then there exists an algorithm $\mathcal{M}$ that breaks a $MSIS_{h,v,t}$-instance*

- *with probability* $\left(\epsilon - \frac{1}{|\overline{D}|-q_H-Nq_S}\right)^2 / \left((N^2+N)q_H\right)^2$,
- *in time* $O(N^2 \cdot q_H \cdot s)$,
- *where* $t = 4\sigma\sqrt{v \cdot \nu} + 2 \cdot \kappa \cdot v \cdot \nu^{1.5} \cdot \beta$.

For simplicity, we assume in the reduction that N is fixed, which means that we can easily construct a reduction for an upper-bounded L.

Proof. Consider the following algorithm $\mathcal{M}$:

1. Query for an $MSIS$-challenge $\boldsymbol{A} \in R_q^{h \times v}$. Sample $\boldsymbol{B} \leftarrow R_q^{h \times v}$ uniformly at random. Set $PP = (\boldsymbol{A}, \boldsymbol{B})$.
2. Set up oracles $\mathcal{O}_K, \mathcal{O}_S$ for $\mathcal{A}$ and simulate these by running **KGen**, **Sign** honestly.
3. Run $\mathcal{R}$ from Lemma 3 with $\mathcal{A}$.
4. If $\mathcal{R}$ does not output $\perp$ then return $\boldsymbol{s} = (d - \hat{d})\boldsymbol{r}_\pi - (\hat{\boldsymbol{r}} - \boldsymbol{r})$.

Observe that if $\mathcal{R}$ generates output, then we obtain $d, \hat{d}, \boldsymbol{r}, \hat{\boldsymbol{r}}, \pi$ such that

$$(d - \hat{d})PK_\pi = \boldsymbol{A}(\boldsymbol{r} - \hat{\boldsymbol{r}}) \text{ and } (d - \hat{d})I = \boldsymbol{B}(\boldsymbol{r} - \hat{\boldsymbol{r}}).$$

PK_π was generated honestly by $\mathcal{O}_K$ and we have $\boldsymbol{r}_\pi$ such that $PK_\pi = A\boldsymbol{r}_\pi$. Rewrite the above as

$$\boldsymbol{A}(d - \hat{d})\boldsymbol{r}_\pi = \boldsymbol{A}(\boldsymbol{r} - \hat{\boldsymbol{r}})$$

Assume that $(d - \hat{d})\boldsymbol{r}_\pi = (\boldsymbol{r} - \hat{\boldsymbol{r}})$ then by the invertibility of $(d - \hat{d})$ it holds that

$$I_\pi = \boldsymbol{B}\boldsymbol{r}_\pi = \boldsymbol{B}\left((\boldsymbol{r} - \hat{\boldsymbol{r}}) \cdot (d - \hat{d})^{-1}\right) = I$$

which contradicts the assumption that I is different from all honestly generated tags. Hence $(d - \hat{d})\boldsymbol{r}_\pi \neq (\hat{\boldsymbol{r}} - \boldsymbol{r})$ and thus $\boldsymbol{s} \neq 0$, while $0 = \boldsymbol{A}\boldsymbol{s}$ which yields a solution $\boldsymbol{s}$ to the $MSIS$-instance obtained in Step 1 as required in Definition 1.

The runtime of $\mathcal{M}$ is clearly dominated by the runtime of $\mathcal{R}$. To give an upper-bound on the size of $\boldsymbol{s}$: the vector $\boldsymbol{r}_\pi$ was generated honestly as in **KGen**, therefore $||\boldsymbol{r}_\pi||_\infty \leq \beta$. Using $d, \hat{d} \in D$ we obtain that $||(d - \hat{d})\boldsymbol{r}_\pi||_2 \leq 2 \cdot \kappa \cdot \beta \cdot v \cdot \nu^{1.5}$. By the triangular inequality, the bound on $\boldsymbol{s}$ follows. □

For the Exculpability proof, we need a different extractor which is also based on the Forking Lemma. Again, the actual construction can be found in the appendix. In the proof of Exculpability, we use the following result:

Lemma 4. *Let $\mathcal{A}$ be a PPT algorithm with oracle access to $\mathcal{O}_K, \mathcal{O}_S, H$ which queries H at most q_H times and $\mathcal{O}_S$ at most q_S times. Moreover, $\mathcal{A}$ on input $1^\lambda, PP$ outputs $L, (m^{(1)}, \Omega^{(1)}), (m^{(2)}, \Omega^{(2)})$ as defined in Definition 8 with probability ϵ in time s. Then there exists an algorithm $\mathcal{R}'$ that outputs $(I, \boldsymbol{r}, \hat{\boldsymbol{r}}, d, \hat{d}, \pi)$*

- *such that $\pi \in [N]$, $d, \hat{d} \in D$ and $||\boldsymbol{r}||, ||\hat{\boldsymbol{r}}|| \leq 2\sigma\sqrt{\nu v}$,*

– *it holds that* $dPK_\pi = \boldsymbol{A}\boldsymbol{r}$, $\hat{dPK}_\pi = \boldsymbol{A}\hat{\boldsymbol{r}}$, $dI = \boldsymbol{B}\boldsymbol{r}$ *and* $\hat{dI} = \boldsymbol{B}\hat{\boldsymbol{r}}$,
– *with probability* $\left(\epsilon - \frac{1}{|\overline{D}|-q_H-Nq_S}\right)^2 / \left((N^2+N)(q_H+N\cdot q_S)\right)^2$,
– *in time* $O(N \cdot q_H \cdot s)$.

Proof. See Appendix A.

We do not follow the exact same outline as in the proof of Theorem 2, as the same argument that the computed $MSIS$-solution is non-zero does not hold anymore. We resolve this by having a simulator that - indistinguishably for the adversary $\mathcal{A}$ and the extractor $\mathcal{R}'$ - randomly embeds one out of two possible, but different, challenges which it then breaks.

Theorem 3 (Exculpability). *Let algorithm $\mathcal{A}$ be defined as in Lemma 4. Then there exists an algorithm $\mathcal{M}$ that either breaks an $S\text{-}MLWE_{2h,v,\beta}$ instance or an $MSIS_{h,v,t}$-instance*

– *with probability* $\left(\frac{(N-1)\epsilon}{N} - \frac{1}{|\overline{D}|-q_H-Nq_S}\right)^2 / \left((N^2+N)(q_H+N\cdot q_S)\right)^2$,
– *in time* $O(N \cdot q_H \cdot s)$,
– *where* $t = 4\sigma\sqrt{v\cdot\nu} + 2\cdot\kappa\cdot v\cdot\nu^{1.5}\cdot\beta$.

Proof. The algorithm $\mathcal{M}$ which we will construct in the course of this proof will either use the matrix $\boldsymbol{A}$ in **Setup** to implant an $MSIS$-challenge or alternatively choose $\boldsymbol{A}, \boldsymbol{B}$ from an $S\text{-}MLWE$ instance. Whereas in the former case the proof works as before, in the latter one we use a randomly chosen public key and its corresponding tag to embed an $S\text{-}MLWE$ challenge. This then means that we cannot correctly simulate the $\mathcal{O}_S$-oracle as we would need the secret key for it - which is the secret we want to extract! Instead, the proof uses a version of the simulator $\mathcal{S}$ from signer anonymity.

With respect to the **Link** algorithm from our construction, the definition translates into the requirement that the tags $I^{(1)}, I^{(2)}$ from Ω_1, Ω_2 are equal. Moreover, each $I^{(i)}$ must be identical to an honestly generated identification tag for one of the public keys in L, and $\mathcal{A}$ did not obtain both signatures from $\mathcal{O}_S$ and does not possess the secret key for this public key. Let $I = I^{(1)} = I^{(2)}$. The algorithm $\mathcal{M}$ will first fairly flip a bit $b \leftarrow \mathcal{B}_{1/2}$. Then it does the following, based on the value of b:

$b = 0$**:** $\mathcal{M}$ will take a $S\text{-}MLWE$ instance $(\boldsymbol{D}, \boldsymbol{t})$ where $\boldsymbol{D} = \begin{pmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{pmatrix} \in R_q^{2h\times v}$ and $\boldsymbol{t} = \begin{pmatrix}\boldsymbol{t}_0\\ \boldsymbol{t}_1\end{pmatrix} \in R_q^{2h}$ such that $\boldsymbol{A}, \boldsymbol{B} \in R_q^{h\times v}$ and $\boldsymbol{t}_0,\ \boldsymbol{t}_1 \in R_q^h$. Assign $PP = (\boldsymbol{A}, \boldsymbol{B})$ and choose an index $k \in [N]$. For $j \in [N]$ set

$$(PK_j, SK_j) = \begin{cases} (\boldsymbol{A}\boldsymbol{r}_j, (\boldsymbol{r}_j, \boldsymbol{B}\boldsymbol{r}_j)) & \text{if } k \neq j \text{ and for } \boldsymbol{r}_j \leftarrow S_\beta^v \\ (\boldsymbol{t}_0, (\perp, \boldsymbol{t}_1)) & \text{if } k = j \end{cases}$$

We then set the counter $j = 1$. Whenever $\mathcal{A}$ requests a public key from $\mathcal{O}_K$, then output PK_j and increase j by 1. If $j = k$ and $\mathcal{A}$ requests the secret key

then abort. Whenever $\mathcal{O}_S$ is queried, then sign the signature for the queried key s correctly if $s \neq k$, otherwise use the back-patching simulator from the Signer Anonymity proof[7], but with $I_j = \boldsymbol{t}_1$.

$b = 1$**:** $\mathcal{M}$ will take a $MSIS$ instance $\boldsymbol{A} \in R_q^{h \times v}$ as input, sample $\boldsymbol{B} \leftarrow R_q^{h \times v}$ uniformly at random and set $PP = (\boldsymbol{A}, \boldsymbol{B})$. It will additionally choose $k \in [N]$ uniformly at random. $\mathcal{O}_K$ will generate all keys honestly, but abort if $\mathcal{A}$ queries SK_k. $\mathcal{O}_S$ will run **Sign** honestly.

Assume that $\mathcal{A}$ does not query for SK_k, then the output of $\mathcal{A}$ will be independent of the choice of b due to Theorem 1. If $b = 0$ then $\mathcal{A}$ will be stopped if SK_k is queried, but observe that this abort probability is the same in case $b = 1$ as the key PK_k is perfectly indistinguishable from honestly generated public key PK_j. Moreover, the abort probability in the presence of $\mathcal{O}_S$ is identical due to the construction of the oracle, so the probability that $\mathcal{A}$ outputs something is independent of b. This output probability is $\epsilon' = \epsilon \cdot (N-1)/N$ by the random choice of k.

In the next step, $\mathcal{M}$ now runs $\mathcal{A}$ using the algorithm $\mathcal{R}'$ from Lemma 4. If $\mathcal{R}'$ does not output $\perp$ then we obtain values $d, \hat{d}, \boldsymbol{r}, \hat{\boldsymbol{r}}, \pi$ such that

$$(d - \hat{d})\boldsymbol{A}\boldsymbol{r}_\pi = (d - \hat{d})PK_\pi = \boldsymbol{A}(\boldsymbol{r} - \hat{\boldsymbol{r}}) \text{ and } (d - \hat{d})I = \boldsymbol{B}(\boldsymbol{r} - \hat{\boldsymbol{r}})$$

where $\boldsymbol{r}_\pi$ is the secret key belonging to PK_π. We might either have that $(d - \hat{d})\boldsymbol{r}_\pi = \boldsymbol{r} - \hat{\boldsymbol{r}}$ or that inequality holds. Now if the values are not equal, then we can use the same argument as in linkability to extract a $MSIS$ solution (this covers the case when $b = 1$). But in case of equality the approach does not work - unless we are in the setting where the algorithm $\mathcal{M}$ chose $b = 0$. Now we know that equality holds and $\boldsymbol{r}_\pi$ is known to exist as PK_π is a S-$MLWE$ challenge, which we can therefore extract.

More formally, if $b = 0$ and $k = \pi$ then $\mathcal{M}$ will output $\boldsymbol{r}_\pi = (\boldsymbol{r} - \hat{\boldsymbol{r}}) \cdot (d - \hat{d})^{-1}$ as $d - \hat{d} \in D'$. If $b = 1$ then it will instead output $(d - \hat{d})\boldsymbol{r}_\pi + \hat{\boldsymbol{r}} - \boldsymbol{r}$.

We now calculate the probability that the algorithm $\mathcal{M}$ will output a correct answer to either of the two challenges. Therefore, denote with $\mathsf{X}_=$ the event that $(d - \hat{d})\boldsymbol{r}_\pi = \boldsymbol{r} - \hat{\boldsymbol{r}}$, and with $\mathsf{X}_\neq$ the opposite event. Let M denote the event that $\mathcal{M}$ outputs something. As our goal is to lower-bound the probability that the output of $\mathcal{M}$ is correct, we need to determine

$$\Pr[\mathcal{M} \text{ gives correct output}] = \Pr[\mathsf{X}_=, b = 0 | \mathsf{M}] + \Pr[\mathsf{X}_\neq, b = 1 | \mathsf{M}]$$

[7] The anonymity simulation does only provide computational indistinguishability as it uses Proposition 1. Here the correctly generated I_j is known and the simulation is statistically indistinguishable, not just computationally.

If $b = 0$, then by the choice of k, the probability that $\pi = k$ is at least $1/|L|$ and therefore $\Pr[\mathsf{M}|\mathsf{X}_=, b = 0] \geq 1/N$. Using Bayes' Theorem, we obtain that

$$\begin{aligned}\Pr[\mathsf{X}_=, b = 0|\mathsf{M}] &= \frac{\Pr[\mathsf{M}|\mathsf{X}_=, b = 0] \cdot \Pr[\mathsf{X}_=, b = 0]}{\Pr[\mathsf{M}]} \\ &\geq \Pr[\mathsf{M}|\mathsf{X}_=, b = 0] \cdot \Pr[\mathsf{X}_=, b = 0] \\ &\geq 1/N \cdot \Pr[\mathsf{X}_=] \cdot \Pr[b = 0] = 1/2N \cdot \Pr[\mathsf{X}_=]\end{aligned}$$

where we use in the last step that the occurrence of $\mathsf{X}_=$ is independent of b.

In case of $b = 1$ we always give output, so we have that $\Pr[\mathsf{M}|\mathsf{X}_{\neq}, b = 1] = 1$. Using the same reasoning as above, we obtain that

$$\Pr[\mathsf{X}_{\neq}, b = 1|\mathsf{M}] \geq 1/2 \cdot \Pr[\mathsf{X}_{\neq}]$$

which yields an overall bound of $\Pr[\mathcal{M} \text{ gives correct output}] \geq 1/2N$.

For the size t of the extracted $MSIS$-solution we can use the same argument as in the proof of Theorem 2. The runtime is dominated by the runtime of $\mathcal{R}'$, except that the success probability of $\mathcal{A}$ is ϵ' instead of ϵ due to the adjustment of the oracles. □

The proof of Theorem 3 can be simplified such as to rely only on the $MSIS$ assumption if one can guarantee that the event $\mathsf{X}_{\neq}$ will occur with good probability (where multiple solutions for the given $MSIS$ instance exist and the $\boldsymbol{r}_\pi$ used in the signature is hidden). See [31] for details.

4 Discussion

In this section, we discuss questions surrounding the practicality of our scheme and hint at future research directions.

Practical Considerations. The runtime of **Vfy** is essentially the N-fold runtime of the verification of a regular lattice-based signature scheme. For signing, the computation and sampling of $I_\ell, \boldsymbol{u}$ as well as $\boldsymbol{r}_{r,j}, \boldsymbol{A}\boldsymbol{r}_{z,j}, \boldsymbol{B}\boldsymbol{r}_{z,j}$ for $j \neq \ell$ can be done offline. The size of the total signature is approximately the size of N individual lattice-based signatures, which we estimate will outperform schemes such as [13] for small and moderate-size rings ($N \lessapprox 50$) in practice.

As the basis of our construction, we chose a simple signature scheme without optimizations. Following the outline of our algorithms, one can instantiate it with e.g. [16] and then use their key-compression technique: this optimization is important when it comes to signature size.

Parameter Selection. As a simplification of our assumptions, note that the D-$MLWE$-instance from Theorem 1 and the S-$MLWE$-instance in Theorem 3 have the same dimensions and bounds. Moreover, any algorithm that solves the

S-$MLWE$ problem in time t with success probability ϵ can be turned into a distinguisher for D-$MLWE$ for the same dimension with essentially the same runtime and success probability. It thus suffices to look at the D-$MLWE$-instance only.

Unfortunately, it seems like the security reduction cannot be used for e.g. the choice of H, as it is inherently non-tight: from the proofs in Section 3, we see that the reductions have a huge loss in terms of success probability (both due to the use of the Forking Lemma and because the runtime is proportional to the number of queries of $\mathcal{A}$ to H). If one attempts to obtain a good success probability of the reduction, the estimated runtime gets rather large. We leave a proof with a tighter reduction that can be used to instantiate our construction as an open problem.

Post-Quantum Security. It is widely believed that hardness assumptions used in our scheme may offer security in a post-quantum era. On the other hand, it is unlikely that our security proofs carry over to the Quantum Random Oracle Model (QROM, see e.g. [7]): we use adaptive programming of the RO H in Theorem 1, and adaptive rewinding in Theorem 2 and 3. Both of these proof techniques are somewhat inherent to the construction.

We note though that other candidate constructions in the QROM such as [16, 12] also use a form of RO programming (even though not adaptively). Moreover, though it seems unlikely that the Forking Lemma can be proven in the QROM, there exist no attacks on protocols using these proof techniques which stem from this use of the RO, to the best of our knowledge. We leave the construction of an efficient linkable ring-signature scheme with a proof of security in the QROM as an interesting open question.

References

1. Divesh Aggarwal, Gavin K Brennen, Troy Lee, Miklos Santha, and Marco Tomamichel. Quantum attacks on bitcoin, and how to protect against them. *arXiv preprint arXiv:1710.10377*, 2017.
2. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.
3. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
4. Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In *CRYPTO 1989*, pages 194–211, 1989.
5. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *TCC*, volume 6, pages 60–79. Springer, 2006.
6. Daniel J Bernstein and Tanja Lange. Post-quantum cryptography-dealing with the fallout of physics success. *IACR Cryptology ePrint Archive*, 2017:314, 2017.
7. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Asiacrypt*, volume 7073, pages 41–69. Springer, 2011.

8. Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography*, volume 6056, pages 499–517. Springer, 2010.
9. Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086, 2010. http://eprint.iacr.org/2010/086.
10. Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP*, pages 423–434, 2007.
11. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.
12. Rafaël del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler. Practical quantum-safe voting from lattices. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1565–1581, 2017.
13. David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. Cryptology ePrint Archive, Report 2017/1154, 2017. https://eprint.iacr.org/2017/1154.
14. Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT*, pages 609–626, 2004.
15. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology–CRYPTO 2013*, pages 40–56. Springer, 2013.
16. Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. Crystals – dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Report 2017/633, 2017. http://eprint.iacr.org/2017/633.
17. Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In *International Cryptology Conference*, pages 335–352. Springer, 2014.
18. Matthew Franklin and Haibin Zhang. A framework for unique ring signatures. Cryptology ePrint Archive, Report 2012/577, 2012. http://eprint.iacr.org/2012/577.
19. Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. *Public Key Cryptography–PKC 2007*, pages 181–200, 2007.
20. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.
21. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Annual International Cryptology Conference*, pages 112–131. Springer, 1997.
22. Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT (2)*, volume 9057, pages 253–280. Springer Berlin Heidelberg, 2015.
23. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, volume 7428, pages 530–547. Springer, 2012.
24. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph Silverman, and William Whyte. Ntrusign: Digital signatures using the ntru lattice. *Topics in cryptology?CT-RSA 2003*, pages 122–140, 2003.

25. Jeffrey Hoffstein, Jill Pipher, and Joseph Silverman. Nss: An ntru lattice-based signature scheme. *Advances in Cryptology?Eurocrypt 2001*, pages 211–228, 2001.
26. Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT 2008*, pages 372–389, 2008.
27. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, 2015.
28. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–31. Springer, 2016.
29. Benoit Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based prfs and applications to e-cash. Asiacrypt 2017, to appear, 2017. http://eprint.iacr.org/2017/856.
30. Joseph K Liu, Victor K Wei, and Duncan S Wong. Linkable and anonymous signature for ad hoc groups. In *ACISP04, LNCS 3108, pp. 325–335*. Citeseer, 2004.
31. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 598–616. Springer, 2009.
32. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 738–755. Springer, 2012.
33. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP*, pages 144–155, 2006.
34. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010.
35. Vadim Lyubashevsky and Gregor Seiler. Partially splitting rings for faster lattice-based zero-knowledge proofs. Cryptology ePrint Archive, Report 2017/523, 2017. https://eprint.iacr.org/2017/523.
36. Carlos Aguilar Melchor, Slim Bettaieb, Xavier Boyen, Laurent Fousse, and Philippe Gaborit. Adapting lyubashevsky's signature schemes to the ring signature setting. In *International Conference on Cryptology in Africa*, pages 1–25. Springer, 2013.
37. Mohamed Saied Emam Mohamed and Albrecht Petzoldt. Ringrainbow–an efficient multivariate ring signature scheme. In *International Conference on Cryptology in Africa*, pages 3–20. Springer, 2017.
38. Lan Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA*, pages 275–292, 2005.
39. Shen Noether, Adam Mackenzie, et al. Ring confidential transactions. *Ledger*, 1:1–18, 2016.
40. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proceedings of the Third conference on Theory of Cryptography*, pages 145–166. Springer-Verlag, 2006.
41. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
42. Ronald Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. *Advances in Cryptology - ASIACRYPT 2001*, pages 552–565, 2001.
43. Patrick P. Tsang and Victor K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC*, pages 48–60, 2005.

44. Nicolas van Saberhagen. Cryptonote v 2.0. *https://cryptonote.org/whitepaper.pdf*, 2013.
45. Jin Wang and Bo Sun. Ring signature schemes from lattice basis delegation. In *ICICS 2011*, pages 15–28, 2011.
46. Rupeng Yang, Man Ho Au, Junzuo Lai, Qiuliang Xu, and Zuoxia Yu. Lattice-based techniques for accountable anonymity: Composition of abstract stern's protocols and weak prf with efficient protocols from lwr. Cryptology ePrint Archive, Report 2017/781.

Appendix A Simulation and Knowledge Extraction

In this appendix we explain how to simulate signatures for our scheme and how to extract knowledge from the specific adversaries as is needed in the proofs in Section 3.

A.1 Simulation

The proof of Theorem 1 follows a similar pattern as in [30, 42] and is essentially an OR-proof [11]: the **Sign** algorithm first simulates $N-1$ individual signatures for the corresponding public keys, before using the challenge that is obtained for the last signature to *close the ring*. A simulator without any private keys can thus generate all N individual signatures this way and close the ring by reprogramming the random oracle on one specific input.

Proof (Theorem 1). We will argue that there exists a simulator $\mathcal{S}$ that outputs signatures for a signer ℓ without knowledge of the secret key $SK_\ell = \boldsymbol{r}_\ell$ of ℓ in the programmable random oracle model. We will then argue that its output distribution is computationally indistinguishable from the input to the distinguisher in Definition 6.

The algorithm $\mathcal{S}$ obtains m, L, D_t as input and then does the following:

1. Choose a $PK_\ell \in L$ such that $\forall SK_i \in D_t : \ (PK_\ell, SK_i)$ is not generated by **KGen**.
2. Sample a random $\boldsymbol{r} \in S_\beta^v$ and compute $I_\ell = \boldsymbol{B}\boldsymbol{r}$.
3. Use the back-patching approach as in [42, 30] to generate d_1 and answer the relevant random oracle queries to H:
   (a) For $i \in [N]$, choose $\boldsymbol{r}_{z,i} \leftarrow \mathcal{N}_\sigma^v$.
   (b) Sample $d_1 \leftarrow D$ uniformly at random.
   (c) For $i \in [N]$ compute $t_{i,1} = \boldsymbol{A}\boldsymbol{r}_{z,i} - d_i PK_i$ and $t_{i,2} = \boldsymbol{B}\boldsymbol{r}_{z,i} - d_i I_\ell$ as well as $d_{i+1} = H(L, I_\ell, m, t_{i,1}, t_{i,2})$.
   (d) Set the output of $H(L, I_\ell, m, t_{N,1}, t_{N,2})$ to be d_1.
4. Output $\Omega = \left(d_1, (\boldsymbol{r}_{z,i})_{i\in[N]}, I_\ell\right)$.

$\mathcal{S}$ does not need to abort the simulation and output this event, as a verifier will never see aborting signatures in practical use cases.

Note that the simulator created the signature exactly in such a way that it will pass verification: all equations hold trivially, and the check on the bound of $\boldsymbol{r}_{z,i}$ holds with overwhelming probability (and the same probability as for honestly generated signatures) due to Lemma 1. All that remains to show is that the signature follows the right distribution.

In the real signature, d_1 is assumed to be uniformly random in D as it is chosen as output of H. The simulator chooses d_1 with the same distribution. The real $r_{z,i}$, $i \neq \ell$ are drawn from $\mathcal{N}_\sigma^v$, whereas the final $r_{z,\ell}$ is computed as $\boldsymbol{u} + d_\ell \boldsymbol{r}_\ell$. However, Lemma 2 ensures that the distribution of $r_{z,\ell}$ is statistically close to $\mathcal{N}_\sigma^v$. Hence, all simulated $r_{z,i}$ follow a distribution that is statistically

close to the real one. Finally, we note that here I_ℓ is not obtained from the same secret input $\boldsymbol{r}_\ell$ that is used to derive PK_ℓ since the simulator does not know SK_ℓ. However, an adversary cannot distinguish between I_ℓ and the correctly generated counterpart due to Proposition 1. □

If the above simulator is used in an environment where it is queried adaptively and repetitvely (or where PK_ℓ is fixed), we store the choice $I_\ell, \boldsymbol{r}$ in Step 2 in a list to provide signatures that are consistent and linkable if necessary.

A.2 Rewinding

We now give proofs for the Lemmas 3 and 4.

Proof (Lemma 3). By the requirements of Linkability as in Definition 7, each signature $\Omega^{(i)}$ which is generated by the adversary $\mathcal{A}$ must contain an identification tag $I^{(i)}$ and all these identification tags are different. Therefore, at least one identification tag will not correspond to the public keys from the list L. Our goal is now to extract those values mentioned in the definition of the Lemma.

Assume that $\mathcal{A}$ is run with some certain input and that it generates a set of signatures. $\mathcal{A}$ makes queries to both the random oracle H and to the two oracles $\mathcal{O}_K, \mathcal{O}_S$ in order to generate these signatures. We construct an algorithm $\mathcal{R}$ which will run $\mathcal{A}$ with multiple inputs and will attempt to rewind it on one of these inputs with different outputs from the random oracle. During a run, $\mathcal{A}$ will be allowed to make q_H queries to the random oracle directly, but also $\mathcal{O}_S$ indirectly[8] makes $N \cdot q_S$ queries to H to generate all the queried signatures. Now for each of the signatures that $\mathcal{A}$ generates, we have that such a signature $\Omega^{(i)}$ can be verified by **Vfy**. In order to do so, **Vfy** makes N subsequent queries to H, which we denote as

$$d_{j+1}^{(i)} = H\left(L, I^{(i)}, m^{(i)}, \boldsymbol{A}\boldsymbol{r}_{z,j}^{(i)} - d_j^{(i)} PK_j, \boldsymbol{B}\boldsymbol{r}_{z,j}^{(i)} - d_j^{(i)} I^{(i)}\right) \qquad \text{for } j \in [N]$$

where we additionally must have that $d_{N+1}^{(i)} = d_1^{(i)}$. Each d_j from these queries is uniformly random from the set $|\overline{D}|$, so an adversary that generates a signature which is accepted by **Vfy** must either use the random oracle H or guess the output of at least one non-queried input. Therefore, if $\mathcal{A}$ generates a forgery with probability ϵ, then $\mathcal{A}$ generates this forgery while obtaining all the values $d_j^{(i)}$ from H with probability at least $\mu = \epsilon - \frac{1}{|\overline{D}| - q_H - N \cdot q_S}$.

Now consider an output τ of $\mathcal{A}$, then for the queries that $\mathcal{A}$ makes to generate the output, we write the first occurrences of the N queries to **Vfy** for $m^{(i)}, \Omega^{(i)}$ as $X_1, \ldots, X_N$ where

$$X_{i_r} = H\left(L, I^{(i)}, m^{(i)}, t_{j,1}^{(i)}, t_{j,2}^{(i)}\right)$$

[8] These indirect queries are not important when we discuss a signature that does not correspond to any public key.

with $1 \leq i_1 < i_2 < \cdots < i_N \leq q_H$. Moreover, for X_{i_N} we write $X_{i_N} = H(L, I^{(i)}, m^{(i)}, t^{(i)}_{\pi-1,1}, t^{(i)}_{\pi-1,2})$. We call a signature a (ℓ, π)-signature if $\ell = i_1$ and π is defined as above. We call an output τ a (ℓ, π, i)-output if its output signature $\Omega^{(i)}$ is a (ℓ, π) signature. For such a (ℓ, π)-signature we must have that $X_{i_1} = H(L, I^{(i)}, m^{(i)}, \boldsymbol{A}\boldsymbol{r}^{(i)}_{z,\pi} - d^{(i)}_{\pi} PK_{\pi}, \boldsymbol{B}\boldsymbol{r}^{(i)}_{z,\pi} - d^{(i)}_{\pi} I^{(i)})$ where $d^{(i)}_{\pi} = X_{i_N}$, and the overall goal of rewinding will be to obtain two forgeries for the same X_{i_1}, π and i where $I^{(i)}$ does not belong to a public key, which will allow extraction.

We are now ready to introduce the algorithms $\mathcal{R}$, which iterates over all possible $\ell \in [q_H], \pi \in [N], i \in [N+1]$ and will first run $\mathcal{A}$ so it generates a well-formed output. If so, then it checks if it is a (ℓ, π, i)-output. In this case, it will rewind $\mathcal{A}$:

1. Fix a key PP according to **Setup**. For each $\ell \in [q_H], \pi \in [N], i \in [N+1]$ do the following:
   (a) Sample new randomness r for the algorithm $\mathcal{A}$, which includes preparing responses of $\mathcal{O}_K, \mathcal{O}_S$. Moreover, initialize the random oracle H.
   (b) Run $\mathcal{A}$ on r, the oracle H and simulate the oracles $\mathcal{O}_K, \mathcal{O}_S$ truthfully. This will generate an output τ.
   (c) Add (ℓ, π, i, τ, r) to the list Z if:
      – $\tau = (L, \{(m^{(j)}, \Omega^{(j)})\}_{j \in [N+1]})$ and fulfills Definition 7.
      – Let $I^{(j)}$ be the identification tag used in $\Omega^{(j)}$. Moreover, let $F = (I'_1, \ldots, I'_N)$ be the honestly generated identification tags belonging to $L = (PK_1, \ldots, PK_N)$. Add the item to the list if $I^{(i)} \notin F$, and if $\Omega^{(i)}$ is a (ℓ, π) signature.
2. For each $(\ell, \pi, i, \tau, r) \in Z$, we do the following:
   (a) Rewind $\mathcal{A}$ on r until right after the ℓth query to H. Then, for each new query to H sample a new output value.
   (b) For the new output $\hat{\tau}$ check if
      – $\hat{\tau} = (L, \{(\hat{m}^{(j)}, \hat{\Omega}^{(j)})\}_{j \in [N+1]})$ and fulfills Definition 7.
      – Check if $\hat{\Omega}^{(i)}$ is a (ℓ, π)-signature[9].
   (c) If the check fails, continue with the next element from Z. Else output $(I^{(i)}, \boldsymbol{r}^{(i)}, \hat{\boldsymbol{r}}^{(i)}, d^{(i)}, \hat{d}^{(i)}, \pi)$.
3. Output $\perp$.

We now examine the probability that $\mathcal{R}$ outputs a pair in step 2: there exists (ℓ, π, i) such that the probability that $\mathcal{A}$ generates an output τ that will be added to Z is at least $\frac{\mu}{(N^2+N) \cdot q_H}$. Then by the proof of [30, Theorem 1] and their Rewind-on-Success Lemma we get that $\mathcal{R}$ will output $(I^{(i)}, \boldsymbol{r}^{(i)}, \hat{\boldsymbol{r}}^{(i)}, d^{(i)}, \hat{d}^{(i)})$ with probability at least $\left(\frac{\mu}{(N^2+N) \cdot q_H}\right)^2$. The runtime of $\mathcal{R}$ can be approximated by the number of iterations of its loops, which is $2 \cdot (N^2 + N) \cdot q_H \cdot s$. □

[9] This is in fact enough as the input to H for this query will be fixed. This means that the used tag is the same (as it is input to the random oracle) and also the list of public keys, which means that the tag will still not belong to a public key.

Proof (Lemma 4). With respect to the **Link** algorithm from our construction, the definition translates into the requirement that the tags $I^{(1)}, I^{(2)}$ from $\Omega^{(1)}$, $\Omega^{(2)}$ are equal. Moreover, each $I^{(i)}$ must be identical to an honestly generated identification tag for one of the public keys in L. Let $I = I^{(1)} = I^{(2)}$.

We describe the algorithm now for completeness, but will neither give intuition nor analysis of the success probability as it follows directly from the algorithm $\mathcal{R}$ from the Lemma 3. $\mathcal{R}'$ works as follows:

1. For each $\ell \in [q_H], \pi \in [N], i \in \{1, 2\}$ do the following:
   (a) Run $\mathcal{A}$ on fresh randomness r, the oracle H and simulate the oracles $\mathcal{O}_K, \mathcal{O}_S$ using **KGen**, **Sign**. This will generate an output τ.
   (b) Add (ℓ, π, i, τ, r) to the list Z if:
      – $\tau = (L, (m^{(1)}, \Omega^{(1)}), (m^{(2)}, \Omega^{(2)}))$ and fulfills Definition 8.
      – $\Omega^{(i)}$ is a (ℓ, π)-signature, SK_i was not queried and $\Omega^{(i)}$ was not an output of $\mathcal{O}_S$.
2. For each $(\ell, \pi, i, \tau, r) \in Z$, we do the following:
   (a) Rewind $\mathcal{A}$ on r until right after the ℓth query to H. Then, for each new query to H sample a new output value.
   (b) For the new output $\hat{\tau}$ check if
      – $\hat{\tau} = (L, (\hat{m}^{(1)}, \hat{\Omega}^{(1)}), (\hat{m}^{(2)}, \hat{\Omega}^{(2)}))$ and fulfills Definition 8.
      – $\hat{\Omega}^{(i)}$ is a (ℓ, π)-signature, SK_i was not queried and $\hat{\Omega}^{(i)}$ was not an output of $\mathcal{O}_S$.
   (c) If the check fails, continue with the next element from Z. Else output $(I, \boldsymbol{r}^{(i)}, \hat{\boldsymbol{r}}^{(i)}, d^{(i)}, \hat{d}^{(i)}, \pi)$.
3. Output $\perp$.

Using the same argument as in the linkability proof, the probability that $\mathcal{R}'$ succeeds in not outputting $\perp$ is at least $\left(\frac{\mu}{2 \cdot N(q_H + N q_S)}\right)^2$ for $\mu = \epsilon - \frac{1}{|\overline{D}| - q_H - N \cdot q_S}$. The runtime of $\mathcal{R}'$ is dominated by the number of iterations of the inner loops and can be approximated as $O(N \cdot q_H \cdot s)$. □