Ring Signature Schemes from Lattice Basis Delegation

Jin Wang[1,2,⋆] and Bo Sun[1]

[1] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China
[2] Institute for Advanced Study , Tsinghua University, Beijing 100084, China
jimiwang@mail.tsinghua.edu.cn, {wj,sb}@cert.org.cn

Abstract. In this paper, we propose a set of ring signature (RS) schemes using the lattice basis delegation technique due to [6,7,12]. Our proposed schemes fit with ring trapdoor functions introduced by Brakerski and Kalai [18], and we obtain the first lattice-based ring signature scheme in the random oracle model. Moreover, motivated by Boyen's work [16], our second construction in the standard model achieves in stronger security definitions and shorter signatures than Brakeski-Kalai scheme.

Keywords: Ring signature, lattices, lattice basis delegation.

1 Introduction

Ring Signature. Ring signature [13] is a type of group-oriented signatures which provides anonymity in some scenarios. In a ring signature scheme, a message signer forms a ring of any set of possible signers including him/herself. The message signer can then generate a ring signature using his/her secret key and public keys of other ring members. The generated ring signature can convince an arbitrary verifier that the message was signed by one of the ring members without revealing exactly the singer's identity. Ring signature schemes could be used for many applications such as anonymous membership authentication, whistle blowing etc.

Motivations. Nowadays , most of the existing ring signature constructions are based on hard number theory assumes ranging from the large integer factorization [5,13] and the discrete logarithm problem [1,10] to the diffie-hellman problem of bilinear pairings [15]. However, above number theory problems will be solvable if practical quantum computers become reality, so it implies a potential security threat to these schemes. Thus, a natural question is how to design ring signature systems that are secure in the quantum environment. In recent years, lattices have emerged as a possible alternative to number theory. Lattice-based cryptography began with the seminal work of Ajtai[3], who showed

⋆ Supported by 973 Project (No.2007CB807902), National Natural Science Foundation of China (NSFC Grant No.60910118).

S. Qing et al. (Eds.): ICICS 2011, LNCS 7043, pp. 15–28, 2011.

that it is possible to construct cryptographic functions in which average-case security provably related to the worst-case complexity of hard lattice problems. Lattice-based constructions also enjoy great simplicity and is believed to be secure against quantum computers.

Our Contribution. Following above discussion, in this paper, we focus on constructing ring signature schemes from lattices. The idea behind our constructions is based on the lattice delegation method due to [6,7,12]. In our ring signature scheme, the public/secret key pair of each user is simply a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a corresponding short basis $\mathbf{B} \in \mathbb{Z}^{m \times m}$ for the lattice $\Lambda^{\perp}(\mathbf{A})$. For the ring set R of size l, the singer constructs a public matrix corresponding to the ring set as $\mathbf{A}_R = [\mathbf{A}_1 \| ... \| \mathbf{A}_l]$(for $i \in R, 1 \leq i \leq l$). Following the "hash-and-sign" paradigm as in [9], a message $M \in \{0,1\}^*$ is hashed to some point $\mathbf{y} = H(M)$. Using the basis delegation technique, each member in R should be able to deduce a signature (short vector $\mathbf{e} \in \mathbb{Z}^{lm}$) on M that satisfies $\mathbf{A}_R\mathbf{e} = \mathbf{y} \bmod q$. Since short basis for lattices essentially functions like cryptographic trapdoors [4,9], only the ring members in R can generate the signature successfully. Our ring signature scheme holds anonymity against full key exposure and unforgeability against insider corruption in the random oracle model. Moreover, using the similar technique in [16,17] , we can modify our basic construction to obtain a ring signature scheme in the standard model. To the best of author's knowledge, our constructions consist of the first lattice-based ring signature scheme in the random oracle model and achieve in stronger security definitions and shorter signatures than Brakeski and Kalai's work [18] in the standard model.

1.1 Related Work

Comparing with Brakerski and Kalai's Work. Brakerski and Kalai [18] recently presented a generic framework for constructing ring signature schemes in the standard model, and obtained a corresponding scheme based on SIS assumption. Our proposed ring signature schemes fit with ring trapdoor functions in [18] at a technical level. However, their work does not include ring signature in the random oracle model. For the lattice-based ring signature in the standard model, our construction is motivated by Boyen's work [16] and results in shorter signature than Brakeski-Kalai scheme. Moreover Brakeski-Kalai scheme is proved unforgeable under chosen subring attacks, and our standard model scheme is provable secure under a stronger model (unforgeability with insider corruption).

Lattice-based Signature. Our cryptographic constructions are based on the hardness assumption of the Small Integer Solution Problem (SIS)[11]. For reasonable choices of parameters, SIS is as hard as the shortest vector problem (SVP) in lattices. Gentry, Peikert, and Vaikuntanathan [9] constructed a kind of trapdoor primitive called Pre-image Sampling functions that, given a basis of a hard integer lattice, samples lattice points from a *Discrete Gaussian* probability distribution whose standard deviation is essentially the length of the longest

Gram-Schmidt vector of the basis. As the application of above trapdoors, Gentry et al. [9] constructed "hash and sign" digital signature schemes based on SIS. Another notable recent work is due to Cash et al.[6,7,12] who constructed a basis delegation technique that allows one to derive a short basis of a given lattice using a short basis of a related lattice. Using this basis delegation technique, Cash et al.[6] also constructed a stateless signature of lattice-based constructions.

2 Preliminaries

2.1 Notation

For a positive integer d, $[d]$ denotes the set $\{1, ..., d\}$. For an $n \times m$ matrix $\mathbf{A}$, let $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_m]$, where $\mathbf{a}_i$ denotes the i-th column vector of $\mathbf{A}$. We define $\|\mathbf{a}\|$ for the Euclidean norm of $\mathbf{a}$, and $\|\mathbf{A}\| = \max_{i \in [m]} \|\mathbf{a}_i\|$.

2.2 Lattices

Lattices. Let $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_n\} \in \mathbb{R}^n$ consist of n linearly independent vectors. A n-dimensional lattice Λ generated by $\mathbf{B}$ is defined as

$$\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bc} : \mathbf{c} \in \mathbb{Z}^n\}$$

Here $\mathbf{B}$ is called a *basis* of the lattice $\Lambda = \mathcal{L}(\mathbf{B})$. For a basis $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_n\}$, let $\widetilde{\mathbf{B}}$ denote its *Gram-Schmidt orthogonalization*, defined iteratively as follows: $\widetilde{b_1} = b_1$, and for $i = 2, ..., n$, $\widetilde{b_i}$ is the component of b_i orthogonal to $\text{span}(b_1, ..., b_{i-1})$.

Hard Random Lattices. In this paper our cryptographic constructions will build on a certain family of m-dimensional integer lattices defined by Ajtai [4].

Definition 1. *Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for some integers q, m, n, define:*

1 . $\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{Ae} = 0 \bmod q\}$
2 . $\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{Ae} = \mathbf{y} \bmod q\}$

Observe that $\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}) = \mathbf{t} + \Lambda^{\perp}(\mathbf{A}) \bmod q$ where $\mathbf{t}$ is an arbitrary solution (over $\mathbb{Z}^m$) of the equation $\mathbf{At} = \mathbf{y} \bmod q$. Thus $\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A})$ is the coset of $\Lambda^{\perp}(\mathbf{A})$.

Discrete Gaussians on Lattices. Here we review Gaussian functions used in lattice based cryptographic constructions. For any $r > 0$ the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with deviation parameter r is defined as

$$\forall \mathbf{x} \in \mathbb{R}^n, \rho_{r,\mathbf{c}}(x) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / r^2)$$

For any $\mathbf{c} \in \mathbb{R}^n$, $r > 0$ and m-dimensional lattice Λ, the discrete gaussian distribution over Λ is defined as

$$\forall \mathbf{x} \in \Lambda, D_{\Lambda_{r,\mathbf{c}}}(x) = \frac{\rho_{r,\mathbf{c}}(\mathbf{x})}{\rho_{r,\mathbf{c}}(\Lambda)}$$

Small Integer Solution Problem. The most well known computational problem on lattices is the *shortest vector problem* (SVP), in which given a basis of a lattice Λ and the goal is to find the shortest vector $\mathbf{v} \in \Lambda \backslash \{0\}$. There is a special version of the SVP for the integer lattices, named *small integer solution* problem (SIS).

Definition 2. *The small integer solution problem* SIS *(in the Euclidean norm l_2) is as follows: given an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a real β, find a nonzero integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{e} = 0 \bmod q$ and $\|\mathbf{e}\|_2 \leq \beta$.*

For functions $q(n)$, $m(n)$, and $\beta(n)$, $\mathsf{SIS}_{q,m,\beta}$ is the ensemble over instances $(q(n), \mathbf{A}, \beta(n))$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is uniformly random. In the following of the paper, a variant problem, the *inhomogeneous* small integer solution problem ISIS is also used. The ISIS problem is as follows: given an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathrm{y} \in \mathbb{Z}_q^n$, and a real β, find an integer vector $\mathrm{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathrm{e} = \mathrm{y} \bmod q$ and $\|\mathrm{e}\|_2 \leq \beta$. Using Gaussian techniques, Micciancio and Regev[11] showed that for any poly-bounded m, β=$poly(n)$ and for any prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the average-case problem $\mathsf{SIS}_{q,m,\beta}$ and $\mathsf{ISIS}_{q,m,\beta}$ are as hard as approximating the SIVP problem (a variant of SVP) in the worst case within a factor $\tilde{O}(\beta \cdot \sqrt{n})$.

2.3 Trapdoors and Basis Delegation Functions

It was shown in [17] that if $\mathsf{SIS}_{q,m,2r\sqrt{m}}$ is hard, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ defines a one-way function $f_{\mathbf{A}} : D_n \to R_n$ with $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \bmod q$, where $D_n = \{\mathbf{e} \in \mathbb{Z}^m : \|\mathbf{e}\| \leq r\sqrt{m}\}$ and $R_n = \mathbb{Z}_q^n$. The input distribution is $D_{\mathbb{Z}^m,r}$. A short basis $\mathbf{B} \in \mathbb{Z}^{m \times m}$ for $\Lambda^{\perp}(\mathbf{A})$ can be used as a trapdoor to sample from $f_A^{-1}(\mathbf{y})$ for any $\mathbf{y} \in \mathbb{Z}_q^n$. Knowledge of such a trapdoor makes it easy to solve some hard problems relative to the lattice, such as SIS. Here we briefly introduce such a set of one-way preimage sampleble functions (defined in [9]), denoted as TrapGen,SampleD, SampleDom, SamplePre, which will be used as building blocks in our cryptographic constructions (we refer interested readers to [9] for more details). The following functions take the Gaussian smoothing parameter $r \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log m})$ as a parameter:

- $\mathsf{TrapGen}(1^\lambda)$: Let n, q, m be integers with $q \geq 2$, $m \geq 5n\log q$. $\mathsf{TrapGen}(1^\lambda)$ outputs a pair $(\mathbf{A}, \mathbf{B})$ such that $\mathbf{A}$ is statistically close to uniform on $\mathbb{Z}_q^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}^{m \times m}$ is a good basis of $\Lambda^{\perp}(\mathbf{A})$ such that $\|\widetilde{\mathbf{B}}\| \leq O(\sqrt{n \log q})$ and $\|\mathbf{B}\| \leq O(n \log q)$ with all but $n^{\omega(1)}$ probability. (Ajtai [4] showed how to sample a pair $(\mathbf{A}, \mathbf{B})$ with low Gram-Schmidt norm. Here we use an improved sampling algorithm from Alwen and Peikert[2]).
- $\mathsf{SampleD}(\mathbf{B}, r, \mathbf{c})$: On input of a m-dimensional basis $\mathbf{B}$ of a lattice Λ, a parameter r, and a center vector $\mathbf{c} \in \mathbb{R}^m$, the algorithm SampleD samples from a discrete Gaussian distribution over the lattice Λ around the center $\mathbf{c}$ with the standard deviation r.
- $\mathsf{SampleDom}(\mathbf{A}, r)$: Sample an $\mathbf{x}$ from distribution $D_{\mathbb{Z}^m,r}$ for which the distribution of $f_{\mathbf{A}}(\mathbf{x})$ is uniform over R_n.

– $\mathsf{SamplePre}(\mathbf{A}, \mathbf{B}, \mathbf{y}, r)$: On input of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a good basis $\mathbf{B} \in \mathbb{Z}^{m \times m}$ for $\Lambda^{\perp}(\mathbf{A})$ as above, a vector $\mathbf{y} \in \mathbb{Z}_q^n$ and r; the conditional distribution of the output $\mathbf{e}$ is within negligible statistical distance of $D_{\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}),r}$. The algorithm works as follows. First, choose via linear algebra an arbitrary $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{At} = \mathbf{y} \bmod q$. Then sample $\mathbf{v}$ from the Gaussian distribution $D_{\Lambda^{\perp}(\mathbf{A}),r,-\mathbf{t}}$ using $\mathsf{SampleD}(\mathbf{B}, r, -\mathbf{t})$, and output $\mathbf{e} = \mathbf{t} + \mathbf{v}$.

2.4 Ring Signature

Ring Signature. A ring signature scheme is a tuple of algorithms RS = (KeyGen, Ring-Sign, Ring-Verify) described as follows:

– $\mathsf{KeyGen}(\lambda)$: A probabilistic algorithm takes as input a security parameter λ and outputs a public key pk and a secret key sk.
– $\mathsf{Ring\text{-}Sign}(pk, sk, R, M)$: A probabilistic algorithm takes as input an user's key pair (pk, sk); a set of public keys R of the ring and a message M to be signed (Here $pk \in R$). It returns a ring signature σ of M under sk.
– $\mathsf{Ring\text{-}Verify}(R,M,\sigma)$: A deterministic algorithm takes as input a set of public keys R that constitutes the ring and a ring signature σ on a message M. It outputs "accept" if the ring signature is valid, or "reject" otherwise.

For consistency purposes, we require that for $l \in \mathbb{N}$, all $\{(pk_i, sk_i)_1^l\} \in [\mathsf{KeyGen}(\lambda)]$, all $i \in [l]$ and all $M \in \{0,1\}^*$: $\mathsf{Ring\text{-}Verify}(M, \mathsf{Ring\text{-}Sign}(pk_i, sk_i, R, M)) = 1$ where $R = (pk_1, ..., pk_l)$.

The security of a ring signature scheme consists of two requirements, namely *Anonymity* and *Unforgeability*. Here we follow the strongest security definitions for ring signatures: anonymity against full key exposure and unforgeability with insider corruption, presented by Bender, Katz, and Morselli[5].

***Anonymity*:** Anonymity against full key exposure for a ring signature scheme RS is defined using the following game between a challenger $\mathcal{B}_1$ and an adversary $\mathcal{A}_1$: $\mathcal{B}_1$ firstly runs algorithm KeyGen l times to obtain public/private key pairs $(pk_1, sk_1), ..., (pk_l, sk_l)$. Here l is a game parameter. $\mathcal{A}_1$ is given the public keys $\{pk_i\}_1^l$ and allowed to make ring signing queries and corruption queries. A ring signing query is of the form (i, R, M) where M is the message to be signed, R is a set of public keys, and i is an user index with $pk_i \in R$. The challenger responds with $\sigma = \mathsf{Ring\text{-}Sign}(sk_i, R, M)$. A corruption query is an index i. The challenger provides sk_i to $\mathcal{A}_1$. Finally, $\mathcal{A}_1$ requests a challenge by sending (i_0, i_1, R^*, M^*) to $\mathcal{B}_1$ such that M^* is a message to be signed with the ring R^*, and i_0 and i_1 are indices with $pk_{i_0}, pk_{i_1} \in R^*$. $\mathcal{B}_1$ chooses a bit $b \leftarrow \{0,1\}$, computes the challenge signature $\sigma^* \leftarrow \mathsf{Ring\text{-}Sig}(pk_{i_b}, sk_{i_b}, R^*, M^*)$, and provides $\mathcal{A}_1$ with σ^*. The adversary $\mathcal{A}_1$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

Denote $\mathrm{Adv}_{RS,l}^{anon}(\mathcal{A}_1)$ to be the advantage over $1/2$ of $\mathcal{A}_1$ in the above game. A ring signature scheme RS is anonymous, if for every probabilistic polynomial-time adversary $\mathcal{A}_1$ the advantage $\mathrm{Adv}_{RS,l}^{anon}(\mathcal{A}_1)$ is negligible.

***Unforgeability*:** For a ring signature scheme RS with l public keys, the existential unforgeability (with insider corruption) is defined as the following game between a challenger $\mathcal{B}_2$ and an adversary $\mathcal{A}_2$. $\mathcal{B}_2$ firstly runs the algorithm KeyGen to obtain public/private key pairs $(pk_1, sk_1), ..., (pk_l, sk_l)$. $\mathcal{A}_2$ is given the public keys $L = \{pk_i\}_1^l$ and is allowed to make ring signing queries and corruption queries as in the anonymity game. Finally $\mathcal{A}_2$ outputs a tuple (R^*, M^*, σ^*). $\mathcal{A}_2$ wins the game if all of following conditions satisfied: (1) $R^* \subseteq L \backslash C$, where C is the set of all the corrupted users; (2) (R^*, M^*) has not been submitted to the ring signing oracle; (3) $\mathsf{Ring\text{-}Verify}(R^*, M^*, \sigma^*) = \mathsf{Accept}$.

$\mathcal{A}_2$'s advantage in above game is denoted to be $\mathrm{Adv}^{unfor}_{RS,l,\mathcal{A}_2} = \Pr[\mathcal{A}_2 \text{ wins}]$. A ring signature scheme RS is unforgeable if for every probabilistic polynomial-time adversary $\mathcal{A}_2$ the advantage $\mathrm{Adv}^{unfor}_{RS,l}(\mathcal{A}_2)$ is negligible.

3 Lattice Based Ring Signature in the Random Oracle Model

Our first construction is a lattice based ring signature scheme in the random oracle model. We start with a slight variant of the generalized sampling algorithm GenSamplePre which was first proposed in [7], with different choice of parameters and the structure of the extended lattice. The original algorithm enables the growth of extended matrices in a tree form. In our approach, we will handle with another extension policy better suited for our RS schemes given later.

3.1 Sampling Preimage for Extended Lattice

Let k, k_1, k_2, k_3, k_4 be positive integers and $k = k_1+k_2+k_3+k_4$. Assume without loss of generality that $S = [k]$. We write $\mathbf{A}_S = [\mathbf{A}_{S_1} \| \mathbf{A}_{S_2} \| \mathbf{A}_{S_3} \| \mathbf{A}_{S_4}] \in \mathbb{Z}_q^{n \times km}$, where $\mathbf{A}_{S_1} \in \mathbb{Z}_q^{n \times k_1 m}$, $\mathbf{A}_{S_2} \in \mathbb{Z}_q^{n \times k_2 m}$, $\mathbf{A}_{S_3} \in \mathbb{Z}_q^{n \times k_3 m}$, $\mathbf{A}_{S_4} \in \mathbb{Z}_q^{n \times k_4 m}$. Let $\mathbf{A}_R = [\mathbf{A}_{S_1} \| \mathbf{A}_{S_3}] \in \mathbb{Z}_q^{n \times (k_1+k_3)m}$. Given a short basis $\mathbf{B}_R$ for $\Lambda^{\perp}(\mathbf{A}_R)$ and an integer $r \geq \|\tilde{\mathbf{B}}_R\| \cdot \omega(\sqrt{\log n})$, the algorithm GenSamplePre allows to sample a preimage of the function $f_{\mathbf{A}_S}(\mathbf{e}) = \mathbf{A}_S \mathbf{e} \bmod q$. $\mathsf{GenSamplePre}(\mathbf{A}_S, \mathbf{A}_R, \mathbf{B}_R, \mathbf{y}, r)$ proceeds as follows:

1. Sample $\mathbf{e}_{S_2} \in \mathbb{Z}^{k_2 m}$ from the distribution $D_{\mathbb{Z}^{k_2 m}, r}$ and sample $\mathbf{e}_{S_4} \in \mathbb{Z}^{k_4 m}$ from the distribution $D_{\mathbb{Z}^{k_4 m}, r}$. Parse $\mathbf{e}_{S_2} = [\mathbf{e}_{k_1+1}, ..., \mathbf{e}_{k_1+k_2}] \in (\mathbb{Z}^m)^{k_2}$ and $\mathbf{e}_{S_4} = [\mathbf{e}_{k-k_4+1}, ..., \mathbf{e}_k] \in (\mathbb{Z}^m)^{k_4}$.
2. Let $\mathbf{z} = \mathbf{y} - \mathbf{A}_{S_2}\mathbf{e}_{S_2} - \mathbf{A}_{S_4}\mathbf{e}_{S_4} \bmod q$. Run $\mathsf{SamplePre}(\mathbf{A}_R, \mathbf{B}_R, \mathbf{z}, r)$ to sample a vector $\mathbf{e}_R \in \mathbb{Z}^{(k_1+k_3)m}$ from the distribution $D_{\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A}_S), r}$. Parse $\mathbf{e}_R = [\mathbf{e}_1, ..., \mathbf{e}_{k_1}, \mathbf{e}_{k_1+k_2+1}, ..., \mathbf{e}_{k-k_4}] \in (\mathbb{Z}^m)^{k_1+k_3}$ and let $\mathbf{e}_{S_1} = [\mathbf{e}_1, ..., \mathbf{e}_{k_1}] \in (\mathbb{Z}^m)^{k_1}$, $\mathbf{e}_{S_3} = [\mathbf{e}_{k_1+k_2+1}, ..., \mathbf{e}_{k-k_4}] \in (\mathbb{Z}^m)^{k_3}$.
3. Output $\mathbf{e} \in \mathbb{Z}^{km}$, as $\mathbf{e} = [\mathbf{e}_1, ..., \mathbf{e}_k]$.

Note that by construction, we have $\mathbf{A}_{S_1}\mathbf{e}_{S_1} + \mathbf{A}_{S_3}\mathbf{e}_{S_3} = \mathbf{A}_R\mathbf{e}_R = \mathbf{z} \bmod q$. Thus $\mathbf{A}_S\mathbf{e} = \sum_{i=1}^4 \mathbf{A}_{S_i}\mathbf{e}_{S_i} = \mathbf{y} \bmod q$, and the output vector $\mathbf{e}$ of GenSamplePre is contained in $\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A}_S)$. For the analysis of the output distribution, Theorem 3.4 in [7] showed that $\mathbf{e}$ is within negligible statical distance of $D_{\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A}_S), r}$.

3.2 Basic Construction

Let l, m, n, q, t be positive integers with $q \geq 2$ and $m \geq 5nlogq$. The ring signature scheme shares parameter functions $\widetilde{L}$, r defined in [6] as follows:

- $\widetilde{L} \geq O(\sqrt{n \log q})$: an upper bound on the Gram-Schmidt size of an user's secret basis;
- $r \geq \widetilde{L} \cdot \omega(\sqrt{\log n})$: a Gaussian parameter used to generate the secret basis and short vectors.

The scheme employs a hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^n$. The security analysis will view H_1 as a random oracle.

KeyGen(λ): On input a security parameter λ, an user with index i runs the trapdoor generation algorithm $\mathsf{TrapGen}(1^\lambda)$ (described in section 2.4) to generate $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ with a basis $\mathbf{B}_i \in \mathbb{Z}^{m \times m}$ for $\Lambda^\perp(\mathbf{A}_i)$. Note that by Theorem 3.2 in [2] we have $\|\widetilde{\mathbf{B}}_i\| \leq \widetilde{L}$. The public/private key pair for the user i is $\langle pk_i = \mathbf{A}_i, sk_i = \mathbf{B}_i \rangle$.

Ring-Sign(R, sk_i, M): Given a ring of l individuals with public keys R, assume for notational simplicity that $R = \{\mathbf{A}_1, ..., \mathbf{A}_l\} \in \mathbb{Z}_q^{n \times m}$, an user i's ($i \in [l]$) private key $sk_i = \mathbf{B}_i$ and a message $M \in \{0,1\}^*$, the user i does the following:

- Set $\mathbf{A}_R$=$[\mathbf{A}_1 \| ... \| \mathbf{A}_l] \in \mathbb{Z}_q^{n \times lm}$ and $\mathbf{y} = H_1(M) \in \mathbb{Z}_q^n$. Define a label lab_R that contains information about how $\mathbf{A}_R$ is associated with the sequence of the ring members $\{1, ..., l\}$.
- Generate $\mathbf{e} \leftarrow \mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_i, \mathbf{B}_i, \mathbf{y}, r) \in \mathbb{Z}^{lm}$. Note that $\mathbf{e}$ is distributed according to $D_{\Lambda^\perp_{\mathbf{y}} \mathbf{A}_R, r}$.
- Output the ring signature $\sigma =< \mathbf{e}, lab_R >$.

Ring-Verify(R, M, σ): Given a ring of public keys $R = \{\mathbf{A}_1, ..., \mathbf{A}_l\} \in \mathbb{Z}_q^{n \times m}$, a message M, and a ring signature $\sigma =< \mathbf{e}, lab_R >$, the verifier accepts the signature only if both the following conditions satisfied:

- $0 \leq \|\mathbf{e}\| \leq r\sqrt{lm}$
- $\mathbf{A}_R\mathbf{e}$ mod q=$H_1(M)$.

Otherwise, the verifier rejects.

3.3 Correctness

The scheme's correctness is inherited by the properties of the trapdoor functions [9]. In the signing process, the ring members in R construct an one-way function $f_{A_R} : D_R \rightarrow \mathbb{Z}_q^n$ as $f_{\mathbf{A}_R}(\mathbf{e}) = \mathbf{A}_R\mathbf{e}$ mod q, where $D_R = \{\mathbf{e} \in \mathbb{Z}^{lm} : \|\mathbf{e}\| \leq r\sqrt{lm}\}$ with the following properties:

Correct Distributions: By Lemma 5.1 in [6], the distribution of the syndrome $\mathbf{y} = \mathbf{A}_R\mathbf{e}$ mod q is within statistical distance 2ϵ of uniform over $\mathbb{Z}_q^n$. By Theorem 3.4 in [7], algorithm $\mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_i, \mathbf{B}_i, \mathbf{y}, r)$ samples an element $\mathbf{e}$ from the distribution within negligible statistical distance of $D_{\Lambda^\perp_{\mathbf{y}(\mathbf{A}_R), r}}$.

One-Wayness without Trapdoors: By Theorem 5.9 in [6], inverting a random function $f_{\mathbf{A}_R}$ on an uniform output $\mathbf{y} \in \mathbb{Z}_q^n$ is equivalent to solve the *inhomogeneous small integer solution* problem $\mathsf{ISIS}_{q,lm,r}$.

3.4 Security Analysis

We now prove that our ring signature scheme is anonymous against the full key exposure and unforgeable against the insider corruption following the definitions in section 2.4.

Full Anonymity: Before proving the fully anonymity, we prepare the following lemma on our ring signature scheme.

Lemma 1. *Let (i_0, i_1, R, M) be a tuple such that $M \in \{0,1\}^*$ is a message to be signed with the ring $R = \{\mathbf{A}_1, ..., \mathbf{A}_l\} \in \mathbb{Z}_q^{n \times m}$, i_0 and i_1 are indices with $\mathbf{A}_{i_0}, \mathbf{A}_{i_1} \in R$. If $\mathsf{ISIS}_{q,lm,r}$ is hard, $\sigma_{i_0} \leftarrow$Ring-Sign(sk_{i_0}, R, M) and $\sigma_{i_1} \leftarrow$Ring-Sign(sk_{i_1}, R, M) are computationally indistinguishable.*

Proof: The proof is straightforward from the algorithm Ring-Sign. Recall that in the signing process, the ring signature σ_{i_0} and σ_{i_1} are only vectors in $\mathbb{Z}^{lm}$, they have the same distribution of the domain in $f_{\mathbf{A}_R}$ within negligible statistical distance of $D_{\Lambda^{\perp}_{H_1(M)}(\mathbf{A}_R),r}$ and it implies that σ_{i_0} and σ_{i_1} are computationally indistinguishable.

Theorem 1. *Let $q \geq 2$ and $m \geq 5n \log q$. If H_1 is modeled as a random oracle, the ring signature scheme above is fully-anonymous assuming that $\mathsf{ISIS}_{q,lm,r}$ is hard.*

Proof. Assume that there exists an adaptive adversary $\mathcal{A}_1$ attacking our ring signature scheme following the definition of anonymity against full key exposure. We construct a poly-time algorithm $\mathcal{B}_1$ to simulate the attacking environment for $\mathcal{A}_1$. Both $\mathcal{A}_1$ and $\mathcal{B}_1$ are given as input q_E, the total number of extraction queries that can be issued by $\mathcal{A}_1$. To respond to $\mathcal{A}_1$'s queries in the random oracle, $\mathcal{B}_1$ will maintain two lists H_1 and $\mathcal{G}$, which are initialized to be empty and will store tuples of values.

In the Setup phase, $\mathcal{B}_1$ runs the algorithm TrapGen q_E times to generate $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ with the corresponding short basis $\mathbf{B}_i \in \mathbb{Z}^{m \times m}$ $(1 \leq i \leq q_E)$. $\mathcal{B}$ stores the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i \rangle (1 \leq i \leq q_E)$ in list $\mathcal{G}$ and the system parameters $< \mathbf{A}_1 \| ... \| \mathbf{A}_{q_E}] >$ are given to $\mathcal{A}_1$. In the query phase, $\mathcal{B}_1$ answers the hash queries, corruption queries and signing queries of $\mathcal{A}_1$ as follows:

- *Hash Query to $H_1(M_j)$* : $\mathcal{B}_1$ returns a random $\mathbf{y}_j \in \mathbb{Z}_q^n$ to $\mathcal{A}_1$ and stores $\langle M_j, \mathbf{y}_j \rangle$ in list-H_1.
- *Corruption Query* (i): $\mathcal{B}_1$ looks for the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i \rangle$ in list $\mathcal{G}$ and returns $\mathbf{B}_i$ to $\mathcal{A}_1$.
- *Signing Query*(i, M_j, R_j): It can be assumed, without loss of generality, that $\mathcal{A}_1$ has made a H_1 query on M_j. $\mathcal{B}_1$ searches the tuple $\langle M_j, \mathbf{y}_j \rangle$ in list-H_1 and returns $\mathbf{e}_j \leftarrow$GenSamplePre $(\mathbf{A}_{R_j}, \mathbf{A}_i, \mathbf{B}_i, \mathbf{y}_j, r)$ to $\mathcal{A}_1$.

At some point, $\mathcal{A}_1$ provides $< i_0, i_1, R^*, M^* >$ such that M^* is a message to be signed with the ring R^*, and i_0 and i_1 are indices with $pk_{i_0}, pk_{i_1} \in R^*$. $\mathcal{B}_1$ chooses a bit $b^* \leftarrow \{0,1\}$, and retrieves the tuple $\langle M^*, \mathbf{y}^* \rangle$ in list-H_1. Then $\mathcal{B}_1$ computes the challenge signature $\mathbf{e}^* \leftarrow$ GenSamplePre$(\mathbf{A}_{R^*}, \mathbf{A}_{i_{b^*}}, \mathbf{B}_{i_{b^*}}, \mathbf{y}^*, r)$,

and provides $\mathcal{A}_1$ with $\mathbf{e}^*$. Finally, the adversary $\mathcal{A}_1$ outputs a guess $b' \in \{0,1\}$. In the view of $\mathcal{A}_1$, the behavior of $\mathcal{B}_1$ is statistically close to the one provided by the real anonymity security experiment. Observe that the ring members in R^* construct a one-way function $f_{\mathbf{A}_{R^*}}(\mathbf{e}) = \mathbf{A}_{R^*}\mathbf{e} \bmod q$: with the domain $D_{R^*} = \{\mathbf{e} \in \mathbb{Z}^{lm} : \|\mathbf{e}\| \leq r\sqrt{lm}\}$ and $\mathbb{Z}_q^n$. If $\mathcal{A}_1$ exhibits a different success probability in distinguishing between i_0 and i_1 with non-negligible probability, it will contradict with the lemma 1. Hence, we claim that the adversary $\mathcal{A}_1$ in the anonymity game under the simulated environment has negligible advantage to guess the correct identity.

Unforgeability: The unforgeability proof closely follows the proof for the original lattice signature scheme given by Gentry, Peikert, and Vaikuntanathan [9].

Theorem 2. *Our ring signature scheme is unforgeable with regard to the insider corruption assuming that H_1 is collision resistant and $\mathsf{SIS}_{q,lm,2r}$ is hard (l is a guess for the size of the challenge ring).*

Proof: Let $\mathcal{A}_2$ be an adversary that attacks the unforgeability of the ring signature scheme. We construct a poly-time adversary $\mathcal{B}_2$ that solves $\mathrm{SIS}_{q,lm,2r}$ with probability

$$\mathsf{Adv}^{\mathrm{SIS}}_{q,lm,2r}(\mathcal{B}_2) \geq \frac{\mathsf{Adv}^{unfor}_{RS,l}(\mathcal{A}_2)}{q_E \mathrm{C}_{q_E}^{q_E/2}} - \mathrm{negl}$$

Both the adversary $\mathcal{A}_2$ and the challenger $\mathcal{B}_2$ are given as input q_E, the total number of extraction queries that can be issued by $\mathcal{A}_2$. $\mathcal{B}_2$ interacts with $\mathcal{A}_2$ as follows:

Setup: $\mathcal{B}_2$ chooses $l \in [q_E]$, a guess for the size of the challenge ring. Next $\mathcal{B}_2$ obtains an instance $\mathbf{A}_{R_t} \in \mathbb{Z}_q^{n \times lm}$ from the SIS oracle and parses it as $\mathbf{A}_{i^*} \in \mathbb{Z}_q^{n \times m}$ $(1 \leq i^* \leq l)$. $\mathcal{B}_2$ then picks a vector $\mathbf{t} = (t_1, ..., t_l) \in [q_E]$ and sets $R_t = \{t_1, ..., t_l\}$. To respond to $\mathcal{A}_2$'s hash queries and corruption queries in the random oracle, $\mathcal{B}_2$ will maintain two lists H_1 and $\mathcal{G}$, which are initialized to be empty and will store tuples of values. For $1 \leq i \leq q_E$ and $i \notin \mathbf{t}$, $\mathcal{B}_2$ runs the algorithm $\mathsf{TrapGen}$ to generate $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ with the corresponding short basis $\mathbf{B}_i \in \mathbb{Z}^{m \times m}$ and stores the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i \rangle$ in list $\mathcal{G}$. For $1 \leq i \leq q_E$ and $i = t_{i^*} \in \mathbf{t}$, $\mathcal{B}_2$ sets $\mathbf{A}_i = \mathbf{A}_{i^*} \in \mathbb{Z}_q^{n \times m}$. The system parameters $< \mathbf{A}_1 \| ... \| \mathbf{A}_{q_E}] >$ are given to $\mathcal{A}_2$.

Query Phase: $\mathcal{B}_2$ answers the hash queries, corruption queries and signing queries of $\mathcal{A}_2$ as follows:

- *Hash Query to $H_1(M_j)$.* $\mathcal{B}_2$ chooses a random $\mathbf{e}_j \leftarrow D_{lm,r}$ by running the algorithm $\mathsf{SampleDom}(\mathbf{A}_{R_t}, r)$, returns $\mathbf{y}_j \leftarrow \mathbf{A}_{R_t}\mathbf{e}_j \bmod q \in \mathbb{Z}_q^n$ to $\mathcal{A}_2$ and stores $\langle M_j, \mathbf{e}_j, \mathbf{y}_j \rangle$ in list-H_1.
- *Corruption Query* (i). If $i \notin \mathbf{t}$, $\mathcal{B}_2$ looks for the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i \rangle$ in list $\mathcal{G}$ and returns $\mathbf{B}_i$ to $\mathcal{A}_2$. Otherwise, $\mathcal{B}_2$ aborts.
- *Signing Query*(i, M_j, R_j). It can be assumed, without loss of generality, that $\mathcal{A}_2$ has made a H_1 query on M_j. If $R_j = R_t$, $\mathcal{B}_2$ searches the tuple $\langle M_j, \mathbf{e}_j \rangle$ in list-H_1 and returns $\mathbf{e}_j$ to $\mathcal{A}_2$. Otherwise if the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i \rangle$

contains in list $\mathcal{G}$, $\mathcal{B}_2$ retrieves the tuple $\langle M_j, \mathbf{e}_j, \mathbf{y}_j \rangle$ in list-H_1 and then returns $\sigma_j \leftarrow \mathsf{GenSamplePre}(\mathbf{A}_{R_j}, \mathbf{A}_i, \mathbf{B}_i, \mathbf{y}_j, r)$ to $\mathcal{A}_2$. Otherwise, $\mathcal{B}_2$ looks for a $k \in R_j$ such that $\langle k, \mathbf{A}_k, \mathbf{B}_k \rangle$ contains in list $\mathcal{G}$. $\mathcal{B}_2$ then computes $\sigma_j \leftarrow \mathsf{GenSamplePre}(\mathbf{A}_{R_j}, \mathbf{A}_k, \mathbf{B}_k, \mathbf{y}_j, r)$ and returns σ_j to $\mathcal{A}_2$.

Challenge: Finally, $\mathcal{A}_2$ outputs a forgery $\langle i^*, M^*, \sigma^*, R^* \rangle$. If $R^* \neq R_t$, $\mathcal{B}_2$ aborts. Otherwise, $\mathcal{B}_2$ looks up the tuple $\langle M^*, \mathbf{e}^*, \mathbf{y}^* \rangle$ in list-H_1 and outputs $\mathbf{e}_0 = \sigma^* - \mathbf{e}^*$ as a solution to the SIS instance $f_{\mathcal{A}_{R_t}}$.

Analysis. In above process, the probability of an abort is at most $1 - \frac{1}{q_E C_{q_E}^{q_E/2}}$. We claim that the view of $\mathcal{A}_2$ in the unforgeability attack is identical to its view as provided by $\mathcal{B}_2$. For each distinct query M_j to H_1, the value returned by $\mathcal{B}_2$ is $f_{\mathcal{A}_{R_t}}(\mathbf{e}_j) \in \mathbb{Z}_q^n$ where $\mathbf{e}_j \leftarrow \mathsf{SampleDom}(\mathbf{A}_{R_t}, r)$; by the uniform output property of the constructed function, this is identical to the uniformly random value of $H_1(M_j) \in \mathbb{Z}_q^n$ in the real environment. Therefore $\mathcal{A}_2$ outputs a valid forgery $\langle M^*, \sigma^* \rangle$ with probability (negligibly close to) ϵ. Since σ^* is a valid signature of the ring on M^*, we have $\sigma^* < r\sqrt{lm}$ and $f_{\mathcal{A}_{R_t}}(\sigma^*) = H_1(M^*) = f_{\mathcal{A}_{R_t}}(\mathbf{e}^*)$, and they form a collision in $f_{\mathcal{A}_{R_t}}$. Let $\mathbf{e}_0 = \sigma^* - \mathbf{e}^*$, we have $\|\mathbf{e}_0\| \leq 2r\sqrt{lm}$. The probability that $\mathbf{e}_0 = 0$ is at most $n^{-\omega(1)}$. Thus, $\mathcal{B}_2$ solves the SIS instance $\mathrm{SIS}_{q,lm,2r}$.

4 Ring Signature in the Standard Model

Recently, Boyen et al.[16] proposed a framework for fully secure lattice-based signatures in the standard model. Let $\mathbf{A}$ and $\mathbf{S}$ be matrices in $\mathbb{Z}_q^{n \times m}$ and let $\mathbf{R} \in \mathbb{Z}^{m \times m}$ have some distribution with $\|\tilde{\mathbf{R}}\| \leq \omega(\sqrt{\log m})\sqrt{m}$. The key construction in their work is the matrices of the form $\mathbf{F} = [\mathbf{A}|\mathbf{AR} + \mathbf{S}] \in \mathbb{Z}_q^{n \times 2m}$. Given a short basis for either $\Lambda^{\perp}(\mathbf{A})$ or $\Lambda^{\perp}(\mathbf{S})$, they showed that $\mathbf{F}$ is a two-sided preimage samplable function. Using the similar method as in [16,17], we can extend our basic construction in section 3.2 to a ring signature scheme in the standard model. Our construction involves a variant of $\mathbf{F}$ as follows.

Lemma 2. *Fix a matrix $\mathbf{S} \in \mathbb{Z}_q^{n \times m}$ with a short basis $\mathbf{T} \in \mathbb{Z}^{m \times m}$ for $\Lambda^{\perp}(\mathbf{S})$ and $\|\widetilde{\mathbf{T}}\| \leq O(\sqrt{n \log q})$. For $(\mathbf{A}, \mathbf{F})$ such that $\mathbf{A}$ is statistically close to uniform on $\mathbb{Z}_q^{n \times lm}$, $\mathbf{R} \in \mathbb{Z}^{lm \times m}$ with $\|\tilde{\mathbf{R}}\| \leq \omega(\sqrt{\log m})\sqrt{lm}$; $\mathbf{F} = [\mathbf{A}|\mathbf{AR} + \mathbf{S}] \in \mathbb{Z}_q^{n \times (l+1)m}$ is a preimage-samplable function in the sense of section 2.3.*

Proof: The lemma differs from the lemma 23 in [16] in the choice of parameters of the matrices $(\mathbf{A}, \mathbf{R}, \mathbf{F})$, the proof can be deduced using the similar method as in the original one [16] and is omitted here.

4.1 Our Construction in the Standard Model

The scheme involves parameter functions $\widetilde{L}$, r as in section 3.2. For some integers l and d, the following construction assumes that messages are arbitrary d+1-bit strings in $\{0\} \times \{0,1\}^d$. The public parameters also include d+1 independent matrices $\mathbf{C}_0, ..., \mathbf{C}_d \in \mathbb{Z}_q^{n \times m}$.

KeyGen(λ): As in the basic construction in section 3.2.

Ring-Sign(R, sk_i, M): Given a ring with public keys $R = \{\mathbf{A}_1, ..., \mathbf{A}_l\} \in \mathbb{Z}_q^{n \times m}$, an user i's($i \in [l]$) private key $sk_i = \mathbf{B}_i \in \mathbb{Z}^{m \times m}$ ($\|\widetilde{\mathbf{B}}_i\| \leq \widetilde{L}$), and a message $M \in \{0\} \times \{0,1\}^d$, the user i does the following:

- Set $\mathbf{C}_M = \sum_{i=0}^{d} (-1)^{M[i]} \mathbf{C}_i \in \mathbb{Z}_q^{n \times m}$.
- Set $\mathbf{A}_R = [\mathbf{A}_1 \| ... \| \mathbf{A}_l \| \mathbf{C}_M] \in \mathbb{Z}_q^{n \times (l+1)m}$. Define a label lab_R that contains information about how $\mathbf{A}_R$ is associated with the sequence of the ring numbers $\{1, ..., l\}$.
- Generate $\mathbf{e} \leftarrow \mathsf{GenSamplePre}(\mathbf{A}_R, \mathbf{A}_i, \mathbf{B}_i, 0, r) \in \mathbb{Z}^{(l+1)m}$. Note that $\mathbf{e}$ is distributed according to $D_{\Lambda^{\perp}_{\mathbf{A}_R}, r}$.
- Output the ring signature $\sigma = < \mathbf{e}, lab_R >$.

Ring-Verify(R, M, σ): Given a ring of public keys $R = \{\mathbf{A}_1, ..., \mathbf{A}_l\} \in \mathbb{Z}_q^{n \times m}$, a message $M \in \{0\} \times \{0,1\}^d$ and a ring signature $\sigma = < \mathbf{e}, lab_R >$, the verifier accepts the signature only if both the following conditions satisfied:

- $0 \leq \|\mathbf{e}\| \leq r\sqrt{(l+1)m}$
- $[\mathbf{A}_1 \| ... \| \mathbf{A}_l \| \sum_{i=0}^{d} (-1)^{M[i]} \mathbf{C}_i] \mathbf{e} = 0 \bmod q$.

Otherwise, the verifier rejects.

Above ring signature scheme fits with ring trapdoor functions in the standard model proposed by Brakerski and Kalai in [18]. Moreover, our construction is motivated by Boyen's work [16] and results in shorter signatures than Brakeski-Kalai scheme. The anonymity of the scheme can be proved using the similar method as in Theorem 1. Since the concurrent ring signature scheme is based on the signature scheme proposed by Boyen [16] which is existentially unforgeable under a chosen message attacks, it can also be proved unforgeable using the similar way as in [16].

Theorem 3. *Our ring signature scheme is unforgeable with regard to the insider corruption assuming that* $\mathsf{SIS}_{q,lm,r}$ *is hard (l is a guess for the size of the challenge ring).*

The unforgeability proof closely follows the combination of the methods in the proof of Theorem 2 and the proof of Theorem 23 in [16]. The proof sketch is given in Appendix A.

5 Conclusion

In this paper, we propose a set of ring signature schemes using the lattice basis delegation technique. To the best of author's knowledge, our constructions consist of the first lattice-based ring signature in the random oracle model and achieve in stronger security models and shorter signatures than Brakeski and Kalai's work in the standard model.

References

1. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n Signatures from a Variety of Keys. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 415–432. Springer, Heidelberg (2002)
2. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: Proc. of STACS 2009, pp. 75–86 (2009)
3. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: STOC, pp. 284–293 (1997)
4. Ajtai, M.: Generating Hard Instances of the Short Basis Problem. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999)
5. Bender, A., Katz, J., Morselli, R.: Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 60–79. Springer, Heidelberg (2006)
6. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
7. Cash, D., Hofheinz, D., Kiltz, E.: How to delegate a lattice basis. In: Halevi, S. (ed.) CRYPTO rumption (2009). Cryptology ePrint Archive, Report 2009/351 (2009), `http://eprint.iacr.org/2009/351`
8. Chow, S.S.M., Wei, V.K., Liu, J.K., Yuen, T.H.: Ring Signatures without Random Oracles. In: ASIACCS 2006: Proceedings of the 2006 ACM Symposium on Information, Taipei, Taiwan. Computer and Communications Security, pp. 297–302. ACM Press, New York (2006)
9. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)
10. Herranz, J., Sáez, G.: Forking Lemmas for Ring Signature Schemes. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 266–279. Springer, Heidelberg (2003)
11. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. 37(1), 267–302 (2007); Preliminary version in FOCS 2004
12. Peikert, C.: Bonsai Trees: Arboriculture in Lattice-Based Cryptography (2009) (in manuscript)
13. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001)
14. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93 (2005)
15. Shacham, H., Waters, B.: Efficient Ring Signatures without Random Oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 166–180. Springer, Heidelberg (2007)
16. Boyen, X.: Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 499–517. Springer, Heidelberg (2010)
17. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
18. 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)

A Proof of Theorem 3

Proof (Sketch): Let $\mathcal{A}_3$ be an adversary that attacks the unforgeability of the ring signature scheme. We construct a poly-time adversary $\mathcal{B}_3$ that solves $\mathrm{SIS}_{q,lm,\beta}$ with probability

$$\mathsf{Adv}^{\mathrm{SIS}}_{q,lm,\beta}(\mathcal{B}_3) \geq \frac{\mathsf{Adv}^{unfor}_{RS,l}(\mathcal{A}_3)}{qq_E C^{q_E/2}_{q_E}} - \mathrm{negl}$$

The proof takes $\beta = (1+\sqrt{d+1})\sqrt{lm}\omega\sqrt{\log lm})\sqrt{(l+1)m}r$ as a parameter. Both the adversary $\mathcal{A}_3$ and the challenger $\mathcal{B}_3$ are given as input q_E, the total number of extraction queries that can be issued by $\mathcal{A}_3$. $\mathcal{B}_3$ interacts with $\mathcal{A}_3$ as follows:

Setup: $\mathcal{B}_3$ constructs the system's public parameters:

1. Choose $l \in [q_E]$, a guess for the size of the challenge ring.
2. Obtain an instance $\mathbf{A}_{R_t} \in \mathbb{Z}_q^{n\times lm}$ from the SIS oracle and parses it as $\mathbf{A}_{i^*} \in \mathbb{Z}_q^{n\times m}$ $(1 \leq i^* \leq l)$. Then pick a vector $\mathbf{t} = (t_1, ..., t_l) \in [q_E]$ and set $R_t = \{t_1, ..., t_l\}$.
3. Run the algorithm $\mathsf{TrapGen}(1^\lambda)$ to generate $\mathbf{S}_0 \in \mathbb{Z}_q^{n\times m}$ with the corresponding short basis $\mathbf{T}_0 \in \mathbb{Z}^{m\times m}$ for $\Lambda^\perp(\mathbf{S}_0)$.
4. Construct a list $\mathcal{G}$ which is initialized to be empty. For $1 \leq i \leq q_E$ and $i \notin \mathbf{t}$, run the algorithm $\mathsf{TrapGen}(1^\lambda)$ to generate $\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}$ with the corresponding short basis $\mathbf{B}_i \in \mathbb{Z}^{m\times m}$ and store the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i\rangle$ in list $\mathcal{G}$. For $1 \leq i \leq q_E$ and $i = t_{i^*} \in \mathbf{t}$, set $\mathbf{A}_i = \mathbf{A}_{i^*} \in \mathbb{Z}_q^{n\times m}$.
5. Pick $l+1$ short random matrices $\mathbf{R}_0, ..., \mathbf{R}_l \in \mathbb{Z}^{lm\times m}$. Fix $\mathbf{h}_0 = 1 \in \mathbb{Z}_q$ and pick uniformly random scalars $\mathbf{h}_1, ..., \mathbf{h}_l \in \mathbb{Z}_q$.
6. The system parameters $< \mathbf{A}_1, ..., \mathbf{A}_{q_E}, \mathbf{C}_0 = \mathbf{A}_{R_t}\mathbf{R}_0 + \mathbf{h}_0\mathbf{S}_0 \mod q, \mathbf{C}_1 = \mathbf{A}_{R_t}\mathbf{R}_1 + \mathbf{h}_1\mathbf{S}_0 \mod q, ..., \mathbf{C}_l = \mathbf{A}_{R_t}\mathbf{R}_l + \mathbf{h}_l\mathbf{S}_0 \mod q >$ are given to $\mathcal{A}_3$.

Query Phase: $\mathcal{B}_3$ answers the corruption queries and signing queries of $\mathcal{A}_3$ as follows:

- *Corruption Query* (i). If $i \notin \mathbf{t}$, $\mathcal{B}_3$ looks for the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i\rangle$ in list $\mathcal{G}$ and returns $\mathbf{B}_i$ to $\mathcal{A}_3$. Otherwise, $\mathcal{B}_3$ aborts.
- *Signing Query*(i, M_j, R_j). If $R_j = R_t$, $\mathcal{B}_3$ computes the matrix $\mathbf{R}_{M_j} = \sum_{k=0}^{d}(-1)^{M_j[k]}\mathbf{R}_k \in \mathbb{Z}^{(l+1)m}$ and $\mathbf{h}_{M_j} = \sum_{k=0}^{d}(-1)^{M_j[k]}\mathbf{h}_k \in \mathbb{Z}_q$. If $\mathbf{h}_{M_j} \neq 0$, $\mathcal{B}_3$ then constructs the matrix $\mathbf{F}_i = [\mathbf{A}_{R_t} \| \mathbf{A}_{R_t}\mathbf{R}_{M_j} + \mathbf{h}_{M_j}\mathbf{S}_0] \in \mathbb{Z}_q^{n\times(l+1)m}$ and finds a short random $\mathbf{e} \in \Lambda^\perp(\mathbf{F}_i) \subset \mathbb{Z}^{(l+1)m}$ using the trapdoor $\mathbf{T}_0$. Otherwise if the tuple $\langle i, \mathbf{A}_i, \mathbf{B}_i\rangle$ contains in list $\mathcal{G}$, $\mathcal{B}_3$ constructs $\mathbf{F}_j = [\mathbf{A}_{R_j} \| \sum_{i=0}^{d}(-1)^{M_j[i]}\mathbf{C}_i]$ and then returns $\sigma_j \leftarrow \mathsf{GenSamplePre}\ (\mathbf{F}_j, \mathbf{A}_i, \mathbf{B}_i, 0, r)$ to $\mathcal{A}_3$. Otherwise, $\mathcal{B}_3$ looks for a $k \in R_j$ such that $\langle k, \mathbf{A}_k, \mathbf{B}_k\rangle$ contains in list $\mathcal{G}$. $\mathcal{B}_3$ returns $\sigma_j \leftarrow \mathsf{GenSamplePre}\ (\mathbf{F}_j, \mathbf{A}_k, \mathbf{B}_k, 0, r)$ to $\mathcal{A}_3$.

Challenge: Finally, $\mathcal{A}_3$ outputs a forgery $\langle i^*, M^*, \sigma^*, R^*\rangle$. If $R^* \neq R_t$, $\mathcal{B}_3$ aborts. Otherwise, $\mathcal{B}_3$ does the following:

1. Compute $\mathbf{R}_{M^*} = \sum_{k=0}^{d}(-1)^{M^*[k]}\mathbf{R}_k$ and $\mathbf{h}_{M^*} = \sum_{k=0}^{d}(-1)^{M^*[k]}\mathbf{h}_k$.

2. If $\mathbf{h}_{M^*} \neq 0 \mod q$, abort the simulation.
3. Separate $\sigma^* \in \mathbb{Z}^{(l+1)m}$ into $\sigma_1^* \in \mathbb{Z}^{lm}$ and $\sigma_2^* \in \mathbb{Z}^m$ such that $\sigma^* = \begin{pmatrix} \sigma_1^* \\ \sigma_2^* \end{pmatrix}$
4. Return $\mathbf{e}^* = \sigma_1^* + \mathbf{R}_{M^*}\sigma_2^* \in \mathbb{Z}^{lm}$.

Analysis. Let $\mathbf{C}_{M^*} = \sum_{k=0}^{d}(-1)^{M^*[k]}\mathbf{C}_i = \sum_{k=0}^{d}(-1)^{M^*[k]}(\mathbf{A}_{\mathbf{R}^*}\mathbf{R}_k + \mathbf{h}_k\mathbf{S}_0)$. If $\mathbf{h}_{M^*} = 0 \mod q$, we have $\mathbf{C}_{M^*} = \mathbf{A}_{\mathbf{R}^*}\mathbf{R}_{M^*} \mod q$ and then $\mathbf{A}_{\mathbf{R}^*}\mathbf{e}^* = \mathbf{A}_{\mathbf{R}^*}(\sigma_1^* + \mathbf{R}_{M^*}\sigma_2^*) = [\mathbf{A}_{\mathbf{R}^*}|\mathbf{A}_{\mathbf{R}^*}\mathbf{R}_{M^*}]\begin{pmatrix} \sigma_1^* \\ \sigma_2^* \end{pmatrix} = [\mathbf{A}_{\mathbf{R}^*}|\mathbf{C}_{M^*}]\mathbf{e}^* = [\mathbf{A}_{\mathbf{R}_t}|\mathbf{C}_{M^*}]\mathbf{e}^* = 0 \mod q$. Using the similar method as in lemma 26 in [16], we can prove that $\mathbf{e}^*$ is a short non-zero vector with high probability as a solution to the given SIS instance. The probability of an abort in above simulation process is at most $1 - \frac{1}{qq_E C_{q_E}^{q_E/2}}$. We claim that the view of $\mathcal{A}_3$ in the unforgeability attack is identical to its view as provided by $\mathcal{B}_3$. We also choose parameter β similar as in lemma 26,27 in [16] and the complete proof will be given in the full version of the paper.