A Lattice-based Ring Signature Scheme Secure against Key Exposure

Xiaoling Yu and Yuntao Wang*

College of Data Science, Taiyuan University of Technology, Taiyuan, China
Graduate School of Engineering, Osaka University, Osaka, Japan

Abstract. A ring signature scheme allows a group member to generate a signature on behalf of the whole group, while the verifier can not tell who computed this signature. However, most predecessors do not guarantee security from the secret key leakage of signers. In 2002, Anderson proposed the forward security mechanism to reduce the effect of such leakage. In this paper, we construct the first lattice-based ring signature scheme with forward security. Our scheme combines the binary tree and lattice basis delegation technique to realize a key evolution mechanism, where secret keys are ephemeral and updated with generating nodes in the binary tree. Thus, the adversary cannot forge the past signature even if the users' present secret keys are revealed. Moreover, our scheme can offer unforgeability under standard models. Furthermore, our proposed scheme is expected to realize post-quantum security due to the underlying Short Integer Solution (SIS) problem in lattice-based cryptography.

Keywords: Ring signature · Lattice · Forward security · Key exposure · Post-quantum secure.

1 Introduction

Ring signature [31] allows one group member to generate signatures on behalf of this group, where the verifier can confirm that the signer belongs to this group but can not identify the signer. Thus, ring signatures can provide anonymity on the signer's identity and have broad applications, such as Blockchain, ad-hoc networks, anonymous transactions, anonymous whistle-blowing, and so on.

In practical applications, secret keys of singers are revealed easily because of the careless store or internet attacks, etc. Moreover, once a secret key of a member of the group is exposed, an adversary can forge a valid signature on behalf of this group. Thus, the damage from the key exposure is particularly critical in ring signatures. In 2002, Anderson [6] proposed the forward security mechanism for signature schemes to reduce the impact caused by secret key exposure. Specifically, forward security of signatures guarantees that the exposure of a present secret key cannot affect the preceding generated signatures. Its core idea is a key evolution mechanism, where the lifetime of signature schemes is

* Corresponding author: yuxiaoling12@gmail.com; wang@comm.eng.osaka-u.ac.jp.

divided into τ discrete time periods. When a time period is updated to the next one, a new secret key is also computed from the current one by this one-way key evolution, while the current secret key is deleted. Since the key evolution is one-way, the previously generated signature is still secure even if an adversary obtains a current secret key. Therefore, how to design a proper key evolution mechanism is the point of a forward secure ring signature.

On the other hand, current ring signatures are constructed based on the hardness of some number-theoretical problems, such as prime factorization problems, discrete logarithm problems, bilinear maps problems, etc. However, Shor's quantum algorithm [33] shows that all these classical problems can be solved in polynomial time in a practical quantum computer. So Post-Quantum Cryptography (PQC) is widely studied to withstand the attack from quantum computers. In fact, some international standards organizations such as NIST, ISO, and IETF have been conducting PQC standardization projects for a long time. Generally, three primitives are focused on: Public-Key Encryption algorithms (PKE), Key Encapsulation Mechanisms (KEM), and digital signature (DS) schemes. Among the several categories, lattice-based cryptography is considered the most promising candidate for its robust security strength, comparative light communication cost, desirable efficiency, and excellent adaptation capabilities [1]. Indeed, NIST announced three lattice-based PKE/KEM/signature algorithms over four candidate finalists in 2022 [1].

1.1 Contributions and approaches

In this paper, we proposed the first lattice-based ring signature scheme with forward security, which is expected to resist the attack from quantum computers. Under the inspiration of [35,26], the proposed scheme is proved secure under standard models. In this scheme, we combine the binary tree structure and lattice basis delegation technique to realize a key evolution mechanism. Based on this mechanism, secret keys are updated as the change of time periods, which is able to satisfy forward security.

Inspired by the works [22,3,13], we use the leaf nodes in a binary tree structure of the depth l to discretize the lifetime into 2^l intervals. The lattice trapdoor generation algorithm is used to obtain a matrix A_k along with a basis T_{A_k} of lattice $\Lambda_q^{\perp}(A_k)$ as the public key and the initial secret key of group member k, respectively. Without losing generality, assume that the user with index i is the real signer, then A_i is the corresponding matrix of the **root** node of the binary tree. Then we choose $2l$ randomly uniform matrices $A_j^{(b_j)}$ of the size as A_i for $j \in \{1, 2, \ldots, l\}$ and $b_j \in \{0, 1\}$. For each node $\Theta^{(j)} = (\theta_1, \ldots, \theta_k, \ldots, \theta_j)$ with $\theta_k \in \{0, 1\}$ and $k \in \{1, 2, \ldots, j\}$, we set the corresponding matrix $F_{\Theta^{(j)}} = [A_i||A_1^{(\theta_1)}||\ldots||A_j^{(\theta_j)}]$. We employ the lattice basis extension algorithm to compute the trapdoor of any node, inputting the corresponding matrix and the trapdoor of the **root** node (or the trapdoor of its ancestor node). According to the property of the basis extension algorithm, the computation of lattice trapdoors can not be operated inversely, which realizes the one-way key evolution.

After arranging the trapdoor of each node, we apply the minimal cover set to guarantee the signer's secret key $sk_{i,t}$ in time period t includes the ancestor trapdoor for time periods t' ($t' \geq t$) and does not include any trapdoor for time periods t'' ($t'' < t$).

1.2 Related works

Forward security: Anderson[6] first introduced forward security in signatures, which protects the use of past secret keys even if the current key is revealed. Bellare et al. [7] further formalized the definition of forward secure signatures and provided a construction based on the hardness assumption of the integer factorization problem. Then, Abdalla et al. [2] and Itkis et al. [20] did respectively some work to improve the efficiency of [7]. Besides, many forward secure cryptosystems were given, such as forward secure public key encryption systems [9,12,14], forward secure group signatures [24,23,30], forward secure blind signatures [22,15,21], forward secure ring signatures [25,26], forward secure linkable ring signature [11], etc.

Lattice-based Signature: In 2008, Gentry et al. [17] proposed a lattice-based signature scheme using a preimage sampling algorithm. On the one hand, this work showed a "hash-and-sign" paradigm that can achieve high computing speed with a compact design and owns a shorter output size. On the other hand, this paradigm has some shortcomings, i.e., limitations to parameter sets, difficulty in conducting high-speed implementation, and inability to withstand side-channel attacks [28]. In 2010, Cash et al. [13] designed a lattice basis delegation technique that allows obtaining a short basis of a designated lattice from a short basis of a related lattice. They also showed a lattice-based signature scheme with this technique. Many current lattice-based signature schemes adopt this delegation technique to expand the lattice bases. In 2011, Wang et al. [35] constructed a lattice-based ring signature using the delegation algorithm. In 2011, Yu et al.[36] constructed an identity-based signature scheme with forward security. Further, Ling et al. [24] proposed the first forward secure group signature from lattices in 2019. Then, Le et al. [22] gave the first forward secure blind signature from lattices. Simultaneously, Feng et al.[16] gave a traceable ring signature from lattices. In 2022, Hu et al. [19] gave a lattice-based linkable ring signature scheme with standard models.

Ring Signature: Rivest et al. [31] first proposed a ring signature in 2001. Then many ring signature schemes [8,18,34,32] were constructed, whose security models do not rely on random oracles. However, the above schemes do not consider forward security and post-quantum security either. In 2012, Tian et al. [27] gave an efficient lattice-based ring signature scheme that can achieve strong unforgeability under the standard model. In 2008, Liu et al. [25] first proposed a forward secure ring signature to reduce the damage from the key exposure, and they also gave a construction under the random oracle model. Further, Liu et al. [26] showed a forward secure ring signature based on the bilinear maps without random oracles in 2011.

To sum up, due to the apparent resistance to quantum computing attacks, lattice-based cryptography has attracted more and more attention. In particular, the forward security of signatures is considered one of the most promising ways to minimize the damage caused by secret key exposure. However, to the authors' knowledge, there is no lattice-based ring signature scheme with forward security. The work in this paper aims to fill this gap.

1.3 Organization

The rest of the paper is organized as follows. Section 2 shows preliminaries on lattice, hardness assumptions, and related algorithms. Then, we introduce the syntax of ring signature with forward security in Section 3. In Section 4, the specific construction in lattices is given. Finally, we conclude our work in Section 5.

2 Preliminaries

2.1 Notations

For a positive integer n, $[n]$ denotes a set of $\{1, 2, \ldots, n\}$. Without special description, bold lower-case letters, for example, $\mathbf{v}$, denote column vectors. $\|A\|$ and $\|\mathbf{v}\|$ denote the Euclidean norm of a matrix A and a vector $\mathbf{v}$, respectively. $\|\widetilde{A}\|$ denotes the Gram-Schmidt norm of the matrix A, where $\widetilde{A}$ is the Gram-Schmidt orthogonalization of A. $[A_1||A_2] \in \mathbb{Z}_q^{n\times(m_1+m_2)}$ means the concatenation of a matrix $A_1 \in \mathbb{Z}_q^{n\times m_1}$ and a matrix $A_2 \in \mathbb{Z}_q^{n\times m_2}$. $f(x) = O(g(x))$ means there exist positives c and m such that for any $x > m$, $f(x) \leqslant c \cdot g(x)$. $f(x) = \omega(g(x))$ means $\lim\limits_{x\to\infty} \dfrac{f(x)}{g(x)} = \infty$. $\{0, 1\}^l$ denotes a binary string with the length of l.

2.2 Lattice

Definition 1 (Lattice). *Given positive integers n, m and some linearly independent vectors $\boldsymbol{b}_i \in \mathbb{R}^m$ for $i \in [n]$, the set generated by the above vectors $\Lambda(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \{\Sigma_{i=1}^n x_i \boldsymbol{b}_i | x_i \in \mathbb{Z}\}$ is a lattice.*

From the above definition, the set $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is a lattice basis. m is the dimension and n is the rank. One lattice is full-rank if its dimension equals to the rank, namely, $m = n$.

Definition 2. *For positive integers n, m and a prime q, a matrix $A \in \mathbb{Z}_q^{n\times m}$ and a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, define two sets:*

$$\Lambda_q^{\perp}(A) := \{\boldsymbol{e} \in \mathbb{Z}^m | A\boldsymbol{e} = \boldsymbol{0} \mod q\}$$
$$\Lambda_q^{\boldsymbol{u}}(A) := \{\boldsymbol{e} \in \mathbb{Z}^m | A\boldsymbol{e} = \boldsymbol{u} \mod q\}.$$

Assuming that $T \in \mathbb{Z}^{m\times m}$ is a basis of $\Lambda_q^{\perp}(A)$, T is a basis of $\Lambda_q^{\perp}(BA)$ for a full-rank $B \in \mathbb{Z}_q^{n\times n}$.

2.3 Hardness assumptions

Definition 3 (Small integer solution, SIS problem). *Given an integer q, a matrix $A \in \mathbb{Z}_q^{n\times m}$ and a real $\beta > 0$, find a nonzero integer vector $\boldsymbol{e} \in \mathbb{Z}^m$ such that $A\boldsymbol{e} = 0 \mod q$ and $\|\boldsymbol{e}\| \leq \beta$.*

The SIS problem [17,29] has been proved as hard as approximating the worst-case Gap-SVP (smallest vector problem) and SIVP with certain factors.

2.4 Lattice algorithms

Definition 4 (Gaussian distribution). *Given parameter $\sigma \in \mathbb{R}^+$, a vector $\boldsymbol{c} \in \mathbb{R}^m$ and a lattice Λ, $\mathbf{D}_{\Lambda,\sigma,\boldsymbol{c}}$ is a discrete gaussian distribution over Λ with a center $\boldsymbol{c}$ and a parameter σ, denoted by $\mathbf{D}_{\Lambda,\sigma,\boldsymbol{c}} = \dfrac{\rho_{\sigma,\mathbf{c}(x)}}{\rho_{\sigma,\mathbf{c}(\Lambda)}}$ for $\forall x \in \Lambda$, where $\rho_{\sigma,\mathbf{c}(\Lambda)} = \sum_{x\in\Lambda} \rho_{\sigma,\boldsymbol{c}(x)}$ and $\rho_{\sigma,\boldsymbol{c}(x)} = \exp(-\pi\dfrac{\|x-\boldsymbol{c}\|^2}{\sigma^2})$. When $\boldsymbol{c} = 0$, $\mathbf{D}_{\Lambda,\sigma,0}$ can be abbreviated as $\mathbf{D}_{\Lambda,\sigma}$.*

Lemma 1 (TrapGen algorithm). *[4,17,5] Given integers n, m, q with $q > 2$ and $m \geqslant 6n\log q$ as the input, there is a probabilistic polynomial-time (PPT) algorithm $TrapGen$, outputs a matrix $A \in \mathbb{Z}_q^{n\times m}$ along with a basis T_A of the lattice $\Lambda_q^{\perp}(A)$, namely, $A \cdot T_A = 0 \mod q$, where the distribution of A is statistically close to uniform on $\mathbb{Z}_q^{n\times m}$, and $\|\widetilde{T_A}\| \leqslant O(\sqrt{n\log q})$*

Lemma 2 (Preimage sample algorithm). *Given a matrix $A \in \mathbb{Z}_q^{n\times m}$ with a basis $T_A \in \mathbb{Z}_q^{m\times m}$, a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, and a parameter $\sigma \geqslant \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log m})$ as the input, where $m \geqslant 2n\lceil\log q\rceil$, there is a PPT algorithm $SamplePre$, outputs a sample $\boldsymbol{e} \in \mathbb{Z}_q^m$ distributed in $\mathbf{D}_{\Lambda_q^{\boldsymbol{u}}(A),\sigma}$, such that $A\boldsymbol{e} = \boldsymbol{u} \mod q$.*

Lemma 3 (ExtBasis algorithm). *Given an arbitrary matrix $A \in \mathbb{Z}_q^{n\times m}$ whose columns generate the group $\mathbb{Z}_q^n$, an arbitrary basis $S \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^{\perp}(A)$ and an arbitrary matrix $A' \in \mathbb{Z}_q^{n\times m'}$, there is a deterministic ploynomial-time algorithm $ExtBasis$ which can output a basis S'' of $\Lambda_q^{\perp}(A'') \subseteq \mathbb{Z}^{m\times m''}$ such that $\|\widetilde{S}\| = \|\widetilde{S''}\|$, where $A'' = A||A'$, $m'' = m + m'$. Moreover, the above results apply to the situation that the columns of A' are prepended to A. This algorithm can be denoted by $S'' \leftarrow ExtBasis(A'', S)$.*

Lemma 4 (GenSamplePre algorithm). *Given a matrix $A_R = [A_1||A_3]$ and a short basis B_R of the lattice $\Lambda_q^{\perp}(A_R)$, a vector $\boldsymbol{y} \in \mathbb{Z}_q^n$, a parameter $\delta \geq \|\widetilde{B_R}\| \cdot \omega(\sqrt{\log n})$, there is an algorithm $GenSamplePre(A_S, A_R, B_R, \boldsymbol{y}, \delta)$ to sample a preimage $\boldsymbol{e}$ which is within negligible statical distance of $D_{\Lambda_q^{\boldsymbol{y}}(A_S),\delta}$, where $A_1 \in \mathbb{Z}_q^{n\times k_1 m}$, $A_2 \in \mathbb{Z}_q^{n\times k_2 m}$, $A_3 \in \mathbb{Z}_q^{n\times k_3 m}$, $A_4 \in \mathbb{Z}_q^{n\times k_4 m}$, $A_S = [A_1||A_2||A_3||A_4]$, and k_1, k_2, k_3, k_4 are positive integers.*

The *TrapGen* algorithm will be used to generate the public-secret key pairs in the following scheme. And the *GenSamplePre* algorithm can be achieved by invoking *preimage sample* algorithm which was introduced in [35]. Actually, the perimage $\mathbf{e}$ satisfies $A_S\mathbf{e} = \mathbf{y} \mod q$. The *ExtBasis* algorithm will be used to update keys as the change of time periods.

3 Syntax of forward secure ring signature

In this section, we show the model of forward secure ring signature and its security model which was first proposed in [26]. The security of ring signatures is required with two points, anonymity and unforgeability. Especially, for the definition of forward secure ring signatures in Liu et al [26], the model on anonymity is similar to *anonymity against full key exposure* and its forward security is similar to *unforgeability w.r.t. insider corruption.*

3.1 System model

One forward secure ring signature scheme consists of five algorithms, $\Pi =$**(Setup, KeyGen, KeyUpdate, Sign, Verify)**, which was first introduced by Liu et al.[26].

- $pp \leftarrow$ **Setup**(λ): Given the security parameter λ as the input, the setup algorithm outputs the system public parameter pp.
- $(pk_i, sk_{i,0}) \leftarrow$ **KeyGen**(pp): Given the public parameter pp, the key generation algorithm outputs the public-secret key pair $(pk_i, sk_{i,0})$ of user i at the original time, namely, the time period $t = 0$.
- $sk_{i,t+1} \leftarrow$ **KeyUpdate**$(sk_{i,t}, t)$: Given the secret key $sk_{i,t}$ of user i with the time period t as the input, this key update algorithm generates a new secret key $sk_{i,t+1}$ at the time period $t + 1$, and deletes the previous secret key sk_t.
- $\sigma_t \leftarrow$ **Sign**$(sk_{i,t}, \mathbf{m}, R, t)$: Given a time period t, the secret key $sk_{i,t}$, a set R of public keys (represents the ring of users) and the message $\mathbf{m}$ as the input, this algorithm returns a signature σ_t.
- **Verify**$(R, \mathbf{m}, \sigma_t, t)$: Given the public keys set R, the signature σ_t, the message $\mathbf{m}$, as well as the time period t as the input, the algorithm outputs 1 for accept, namely, the signature is valid for this message. Otherwise returns 0 for reject.

3.2 Anonymity

The anonymity of ring signature implies an adversary cannot tell which member of a ring generates signatures. Here we show a game to describe the anonymity against full key exposure [8] on forward secure ring signature, which was first introduced in [26]. Compared with the definition of anonymity in the standard ring signature, the adversary in this model is given the secret keys with the original time period instead of having the right to access a corruption oracle, which means the adversary can obtain the secret keys of all users for any time period. The definition of anonymity for forward secure ring signature is shown in the following game between a challenge $\mathscr{C}$ and an adversary $\mathscr{A}$:

- **Setup**: The challenger $\mathscr{C}$ runs **KeyGen** algorithm for n' times to get public-secret key pairs $(pk_1, sk_{1,0}), \ldots, (pk_{n'}, sk_{n',0})$, then $\mathscr{C}$ sends the public key set $R = \{pk_1, \ldots, pk_{n'}\}$ and the secret key set $\{sk_{1,0}, \ldots, sk_{n',0}\}$ at original time period to the adversary $\mathscr{A}$.
- **Query 1**: $\mathscr{A}$ can query the signing oracle adaptively. $\mathscr{A}$ submits a message $\mathbf{m}$, a time period t, a ring set R with group members' public keys, a public key $pk_i \in R$, then the challenger $\mathscr{C}$ runs the **Sign** algorithm to respond the signing oracle queries.
- **Challenge**: $\mathscr{A}$ chooses a time t^*, a group size n^*, a message $\mathbf{m}^*$, a set R^* of n^* public keys which satisfies two public keys $pk_{i_0}, pk_{i_1} \in R$ are included in R^*, and sends them to $\mathscr{C}$. $\mathscr{C}$ selects randomly a bit $b \in \{0, 1\}$ and runs $\sigma^*_{t^*} \leftarrow \mathbf{Sign}(t^*, n^*, R^*, sk_{i_b,t^*}, \mathbf{m}^*)$. The challenger sends the signature $\sigma^*_{t^*}$ to $\mathscr{A}$.
- **Query 2**: $\mathscr{A}$ is allowed to query the signing oracle adaptively.
- **Guess**: $\mathscr{A}$ returns a guess b'.

$\mathscr{A}$ wins this game if $b' = b$ holds. The advantage that $\mathscr{A}$ wins this game for the security parameter λ is

$$\mathbf{Adv}^{Anon}_{\mathscr{A}}(\lambda) = |Pr[b = b'] - \frac{1}{2}|.$$

Definition 5. *A forward secure ring signature scheme is anonymous, if for any PPT adversary $\mathscr{A}$, the defined advantage $\boldsymbol{Adv}^{Anon}_{\mathscr{A}}(\lambda)$ is negligible.*

3.3 Forward security

The forward security of ring signature schemes is described by the following game which was first introduced in [26]. Here an adversary cannot output a valid signature $\sigma^*_{t^*}$ for a message $\mathbf{m}^*$, a ring R^*, and a time period t^*, such that $Verify(\mathbf{m}^*, \sigma^*_{t^*}, t^*) = 1$ unless either one of public keys in R^* is generated by the adversary or a user whose public key contains in R^* signs $\mathbf{m}^*$. The details of this game are as follows:

- **Setup**: The challenger runs **KeyGen** algorithm for n' times and obtains some public key and original secret key pairs $(pk_1, sk_{1,0}), \ldots, (pk_{n'}, sk_{n',0})$, then he sends the set of public keys $S = (pk_1, \ldots, pk_{n'})$ to the adversary.
- **Query phase**: $\mathscr{A}$ queries the following oracles adaptively.
  - *Corruption oracle query* ($sk_{i,t} \leftarrow CO(pk_i, t)$): Inputting a public key $pk_i \in S$ and a time t, the oracle outputs the corresponding secret key $sk_{i,t}$.

- *Signing oracle query* $SO(t, n, R, pk_i, \boldsymbol{m})$: Inputting a time t, a group size n, a set of n public keys R, a public key $pk_i \in R$ and a message $\mathbf{m}$, this oracle outputs a signature σ_t with the time t.

– **Output**: $\mathscr{A}$ outputs a signature $\sigma^*_{t^*}$, a ring R^* with the number n^* of users, a time t^* and a message $\mathbf{m}^*$.

$\mathscr{A}$ wins the game if the following conditions holds:

1. $Verify(\mathbf{m}^*, \sigma^*_{t^*}, t^*) = 1$,
2. $R^* \subseteq S$,
3. for all $pk^*_i \in R^*$, there is no $CO(pk^*_i, t')$ query with time $t' \leqslant t^*$,
4. there is no $SO(t^*, n^*, R^*, \mathbf{m}^*)$ query.

Definition 6. *A ring signature scheme is unforgeable with forward security, if for all PPT adversary $\mathscr{A}$, the advantage $Adv^{fs}_{\mathscr{A}}(\lambda)$ that $\mathscr{A}$ wins the above game is negligible on the security parameter λ.*

4 A lattice-based ring signature scheme with forward security

In this section, we first show a framework how to generally assign time periods, and generate the corresponding lattice trapdoor for each node in a binary tree. Then, we propose a lattice-based forward secure ring signature scheme.

4.1 Time periods enroll and update in Binary tree

Our construction employs binary tree structure and lattice basis delegation technique, *ExtBasis* algorithm, to realize the update of secret keys with the change of time periods. The details are described as follows.

– **Time arrangement in Binary Tree:**
  - We assign the time periods $t \in \{0, 1, \ldots, 2^l - 1\}$ to the leaf nodes of a binary tree with depth l from leaf to right. Here we show an example as **Fig.**1, where the depth of the tree is $l = 3$ and the number of time intervals is 8.
  - On each time period t, there is an unique path $t = (t_1, \ldots, t_l)$ from the **root** node to leaf node. And for the ith level, $t_i = 0$ if the node in this path is left node, otherwise if the node in this path is right node, $t_i = 1$. Similarly, for the ith level node ($i \neq l$, namely, the non-leaf node), its path from the **root** node to this node is denoted uniquely by $\Theta^{(i)} = (\theta_1, \ldots, \theta_i)$, where $\theta_i \in \{0, 1\}$ is defined as same as t_i.

– **Update of lattice trapdoor of nodes:**

- *TrapGen* algorithm is run to obtain a random matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ and the lattice basis T_{A_0} of lattice $\Lambda^{\perp}(A_0)$. Then we define the corresponding matrix $F_{\Theta^{(i)}} = [A_0||A_1^{(\theta_1)}||\dots||A_i^{(\theta_i)}]$ for $\Theta^{(i)}$, and the matrix $F_t = [A_0||A_1^{(t_1)}||\dots||A_l^{(t_l)}]$ for a time period t, where $A_i^{(b)}$ are random matrices for $i \in \{1, 2, \dots, l\}$ and $b \in \{0, 1\}$. Thus, A_0 can be regarded as the corresponding matrix of **root** node and T_{A_0} is a lattice trapdoor for **root** node.
- Considering the computation of a corresponding lattice trapdoor $T_{\Theta^{(i)}}$ for the node $\Theta^{(i)}$ of the binary tree, we employs lattice basis extension algorithm *ExtBasis*. There are two following situations.
  - ∗ Given the original lattice trapdoor T_{A_0}, the trapdoor $T_{\Theta^{(i)}}$ can be computed as follows.

$$T_{\Theta^{(i)}} \leftarrow ExtBasis(F_{\Theta^{(i)}}, T_{A_0}),$$

    where $F_{\Theta^{(i)}} = [A_0||A_1^{(\theta_1)}||\dots||A_i^{(\theta_i)}]$.
  - ∗ The trapdoor $T_{\Theta^{(i)}}$ can also be computed from its any ancestor's trapdoor. For example, given $T_{\Theta^{(k)}}$,

$$T_{\Theta^{(i)}} \leftarrow ExtBasis(F_{\Theta^{(i)}}, T_{\Theta^{(k)}}),$$

    where $F_{\Theta^{(i)}} = [A_0||A_1^{(\theta_1)}||\dots||A_i^{(\theta_i)}]$ and $\Theta^{(i)} = (\theta_1, \dots, \theta_k, \theta_{k+1}, \dots, \theta_i)$ for $k < i$.
    That is to say, the trapdoor $T_{\Theta^{(i)}}$ is a basis of the lattice $\Lambda^{\perp}(F_{\Theta^{(i)}})$.
- The above methods are also suitable for computing the lattice trapdoor for time periods (i.e., leaf nodes), if its ancestor's lattice trapdoor is known.

4.2 Our lattice-based proposal

Here, we show the lattice-based construction which uses the key evolution (KV) mechanism on the binary tree to achieve the key update and forward secure.

- **Setup**(λ): Given security parameter λ as input, set the number of time period $\tau = 2^l$ where l is the depth of the binary tree, set system parameters n, m, q, d, δ, where n, m are integer, q is prime, d represents the length of the signed messages, δ is the parameter of sampling algorithm, the maximum number of users max, the setup algorithm performs as follows:
  - Choose $2l$ random matrices $A_1^{(0)}, A_1^{(1)}, \dots, A_l^{(0)}, A_l^{(1)} \in \mathbb{Z}_q^{n\times m}$,
  - Choose random and independent vectors $C_0, C_1, \dots, C_d \in \mathbb{Z}_q^{n\times m}$,
  - Outputs the public parameter $pp = (q, n, m, d, \delta, \tau, max, A_1^{(0)}, A_1^{(1)}, \dots, A_l^{(0)}, A_l^{(1)}, C_0, C_1, \dots, C_d)$.

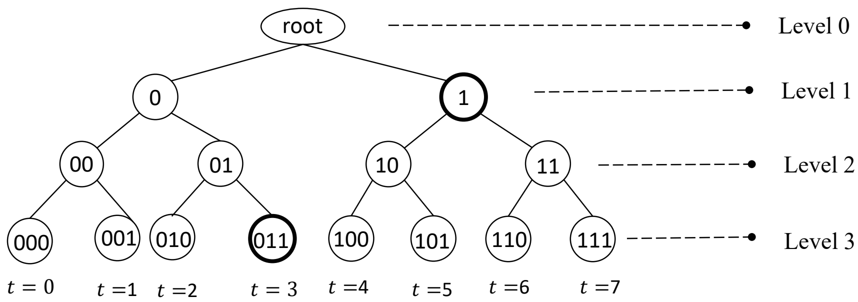

Fig. 1. Binary tree of depth $l = 3$: without losing generality, assume that the singer is the user with the index i in the group, then the corresponding matrix for **root** node is A_i and its trapdoor is T_{A_i}.

- **KeyGen**(pp): Given the public parameter pp, the key generation algorithm performs as follows.
  - For the user with index i $(1 \leq i \leq max)$, run $TrapGen(n, m, q)$ algorithm to obtain a random matrix A_i and a basis T_{A_i} of lattice $\Lambda^{\perp}(A_i)$,
  - Returns the public-secret key $(pk_i, sk_{i,0}) = (A_i, T_{A_i})$ for user i.
- **KeyUpdate**$(pp, sk_{i,t}, pk_i)$: Given the public parameter pp, a secret key $sk_{i,t}$ with the time period t and public key $pk_i = A_i$ of a user with the index i as input, the key update algorithm invokes $ExtBasis$ algorithm combining with the binary tree, and returns the updated secret key $sk_{i,t+1}$ in the time period $t + 1$. The key evolution mechanism is needed to achieve the secret key update as follows.
  - For any leaf node t in the binary tree, a minimal cover $Node(t)$ represents the smallest set that contains an ancestor of all leaves in $\{t, t+1, \ldots, \tau - 1\}$ but does not contains any ancestors of any leaf in $\{0, 1, \ldots, t-1\}$. For example, as shown in **Fig.**1, $Node(0) = \{\textbf{root}\}$, $Node(1) = \{001, 01, 1\}$, $Node(2) = \{01, 1\}$, $Node(3) = \{011, 1\}$, $Node(4) = \{1\}$, $Node(5) = \{101, 11\}$, $Node(6) = \{11\}$, $Node(7) = \{111\}$.
  - Based on the rules in the Section 4.1, each node in the binary tree owns the corresponding trapdoor, for example, for the node "01" in Level 1, its lattice trapdoor is denoted by T_{01} which is a basis of lattice $\Lambda_q^{\perp}(F_{01})$ and $F_{01} = [A_i||A_1^{(0)}||A_2^{(1)}]$. Then the secret key sk_t at the time period t consists of trapdoors of all nodes in the set $Node(t)$. In **Fig.**1, we have $sk_{i,0} = \{T_{A_i}\}$, $sk_{i,1} = \{T_{001}, T_{01}, T_1\}$, where T_{001}, T_{01}, T_1 are the corresponding trapdoor (basis) for $F_{001} = [A_i||A_1^{(0)}||A_2^{(0)}||A_3^{(1)}]$, $F_{01} = [A_i||A_1^{(0)}||A_2^{(1)}]$, $F_1 = [A_i||A_1^{(1)}]$, respectively.

- To realize the update from $sk_{i,t}$ to $sk_{i,t+1}$, the singer determines firstly the minimal cover $Node(t+1)$, then grabs all trapdoors of nodes which are in $Node(t+1) \backslash Node(t)$ by using the methods introduced in Section 4.1, and deletes the trapdoors of nodes in $Node(t) \backslash Node(t+1)$. Finally, the singer can determine the secret key $sk_{i,t+1}$. For example, given $sk_{i,1} = \{T_{001}, T_{01}, T_1\}$, then $sk_{i,2} = \{T_{01}, T_1\}$, where $Node(2) \backslash Node(1) = \{01, 1\}$ and $Node(1) \backslash Node(2) = \{001\}$.
- Based on the above steps, this algorithm outputs the secret key $sk_{i,t+1}$ of the singer with index i in the time period $t+1$, and deletes the secret key $sk_{i,t}$.

– **Sign**$(\mathbf{m}, sk_{i,t}, R, t)$: Given a ring of N users with public keys $R = \{A_1, A_2, \ldots, A_N\}$, the message $\mathbf{m} \in \{0\} \times \{0,1\}^d$ with the length of $d+1$[1], the user i with the secret key $sk_{i,t}$ at the time period t generates a signature as follows:
  - The user i checks firstly if $sk_{i,t}$ contains the trapdoor $T_{\Theta^{(t)}}$. Otherwise, he runs $ExtBasis(F_{\Theta^{(t)}}, T_{\Theta^{(k)}})$ to compute $T_{\Theta^{(t)}}$, where $T_{\Theta^{(k)}}$ is an ancestor basis of $T_{\Theta^{(t)}}$ in the secret key $sk_{i,t}$[2],
  - Set $C_{\mathbf{m}} = \sum_{j=0}^{d} (-1)^{\mathbf{m}[j]} C_j \in \mathbb{Z}_q^{n \times m}$, where $\mathbf{m}[j]$ is the jth bit of the message $\mathbf{m}$,
  - Runs $GenSamplePre(A_{R,t}, F_{i,t}, T_{\Theta^{(t)}}, \mathbf{0}, \delta)$ to obtain $\mathbf{e} \in \mathbb{Z}_q^{[N(l+1)+1]m}$ which satisfies $A_{R,t} \cdot \mathbf{e} = \mathbf{0} \mod q$, where $F_{i,t} = [A_i || A_1^{(t_1)} || \ldots || A_l^{(t_l)}]$, $A_{R,t} = [F_{1,t} || F_{2,t} || \ldots || F_{N,t} || C_{\mathbf{m}}]$,
  - Returns $\sigma_t = \mathbf{e}$ as the ring signature of $\mathbf{m}$ during the time period t.

– **Verify**$(R, \mathbf{m}, \sigma_t, t)$: The verify algorithm performs as follows:
  - Compute $C_{\mathbf{m}} = \sum_{j=0}^{d} (-1)^{\mathbf{m}[j]} C_j$,
  - Verify if $A_{R,t} \cdot \mathbf{e} = \mathbf{0} \mod q$ holds and $\|\mathbf{e}\| \leqslant \delta \sqrt{[N(l+1)+1]m}$, receive this signature. Otherwise, reject it.

4.3 Correctness

According to the GenSamplePre algorithm, the vector $\mathbf{e}$ satisfies $A_{R,t} \cdot \mathbf{e} = \mathbf{0} \mod q$ and $\|\mathbf{e}\| \leqslant \delta \sqrt{[N(l+1)+1]m}$ with overwhelming probability. $\mathbf{e}$ is within negligible statical distance of $D_{\Lambda_q^{\perp}(A_{R,t}),\delta}$.

[1] We can note that any messages can be transformed the fixed length string by a proper collision resistant hash function $H : \{0,1\}^* \rightarrow \{0\} \times \{0,1\}^d$, thus we set the signed messages with fixed length here, which can guarantee various summations can include easily a constant term of index 0 [10].

[2] There are two situations on $T_{\Theta^{(t)}}$. If $T_{\Theta^{(t)}}$ contains in $sk_{i,t}$, for example, $T_{001} \in sk_{i,1}$ as **Fig.**1, then T_{001} can be used directly to generate signature; On the other hand, if $T_{\Theta^{(t)}}$ does not contain in $sk_{i,t}$, for example, when the time period $t = 2$, $T_{010} \notin sk_{i,2} = \{T_{01}, T_1\}$, the $ExtBasis(A_i || A_1^{(0)} || A_2^{(1)} || A_3^{(0)}, T_{01})$ algorithm will be invoked to return T_{010} .

4.4 Security analysis

Theorem 1. *The proposed ring signature scheme is fully-anonymous, if $SIS_{q,N(l+1)m,\delta}$ problem is intractable, where N is the size of ring.*

Proof. Assume that there is an adaptive adversary $\mathscr{A}$ attacking the ring signature of anonymity. And a challenger $\mathscr{C}$ can simulate some operations to respond to $\mathscr{A}$'s queries.

- **Setup**: The challenger $\mathscr{C}$ chooses $2l$ random matrices $A_1^{(0)}, A_1^{(1)}, \ldots, A_l^{(0)}, A_l^{(1)} \in \mathbb{Z}_q^{n\times m}$, $d+1$ random and independent matrices $C_0, C_1, \ldots, C_d \in \mathbb{Z}_q^{n\times m}$, and runs $TrapGen(n, m, q)$ for n' times to get public-secret key pairs $(pk_1, sk_{1,0})$, $\ldots, (pk_{n'}, sk_{n',0})$. Then $\mathscr{C}$ sends the public key set $S = \{pk_1, \ldots, pk_{n'}\}$, the original secret key set $\{sk_{1,0}, \ldots, sk_{n',0}\}$ and the public parameters $(A_1^{(0)}, A_1^{(1)}, \ldots, A_l^{(0)}, A_l^{(1)}, C_0, C_1, \ldots, C_d)$ to the adversary $\mathscr{A}$.
- **Query 1**: $\mathscr{A}$ submits a ring R, a time period t, a public key pk_i and a message $\mathbf{m}$, the challenger $\mathscr{C}$ runs the **sign** algorithm and returns a valid signature σ_t to $\mathscr{A}$.
- **Challenge**: $\mathscr{A}$ chooses a time $t^* = (t_1^*, t_2^*, \ldots, t_l^*)$, a group size n^*, a message $\mathbf{m}^* \in \{0\} \times \{0,1\}^d$, a set $R^* \subset S$ of n^* public keys which satisfies two public keys $pk_{i_0}, pk_{i_1} \in R^*$, and sends $(pk_{i_0}, pk_{i_1}, R^*, \mathbf{m}^*, t^*)$ to $\mathscr{C}$. Then $\mathscr{C}$ selects randomly a bit $b \in \{0,1\}$, and computes $C_{\mathbf{m}^*} = \sum_{j=0}^{d} (-1)^{\mathbf{m}^*[j]} C_j \in \mathbb{Z}_q^{n\times m}$, and runs $\mathbf{e}^* \leftarrow GenSamplePre(A_{R^*,t^*}, F_{i_b,t^*}, T_{\Theta(t^*)}, \mathbf{0}, \delta)$. The challenger sends the signature $\mathbf{e}^*$to $\mathscr{A}$.
- **Query 2**: $\mathscr{A}$ is allowed to query the signing oracle adaptively.
- **Guess**: $\mathscr{A}$ returns a guess $b' \in \{0,1\}$ about b.

In the view of $\mathscr{A}$, the operations of the challenger $\mathscr{C}$ are statistical close to the real anonymity experiments. According to the **Sign** algorithm, for the ring R^*, any valid signature $\mathbf{e}$ and $\mathbf{e}'$, the verification matrix is $A_{R^*,t^*} = [F_{1,t^*}||F_{2,t^*}||\ldots||F_{n^*,t^*}]$, where $F_{i,t^*} = [A_i||A_1^{(t_1^*)}||\ldots||A_l^{(t_l^*)}]$ for $i \in \{1, 2, \ldots, n^*\}$. Actually, this is an one-way function $f_{A_{R^*,t^*}}(\mathbf{e}) = A_{R^*,t^*}\mathbf{e} \mod q$. The signatures $\mathbf{e}$ and $\mathbf{e}'$ are vectors in $\mathbb{Z}_q^{n^*(l+1)m}$ and have the same distribution within negligible statistical distance of $D_{\Lambda^\perp(A_{R^*,t^*}),\delta}$, which means $\mathbf{e}$ and $\mathbf{e}'$ are computationally indistinguishable. Thus the advantage that $\mathscr{A}$ distinguishes signer is negligible, the proposed scheme satisfies the anonymity.

Theorem 2. *The proposed ring signature is unforgeable with forward security, if $SIS_{q,N(1+2l)m,\delta}$ problem is hard, where N is the size of the challenge ring.*

Proof. Assuming that $\mathscr{A}$ is an adversary attacking the unforgeability of the proposed ring signature scheme, then a challenger $\mathscr{C}$ can solve the SIS problem. The process is as follows.

- **Setup**: $\mathscr{C}$ constructs the system parameters as follows:

- Let the maximum number of users be max, the depth of binary tree l, choose $N \leq max$ as the size of the challenge ring.
- **Instance**: Assume $\mathscr{A}$ intends to solve the $SIS_{q,N(1+2l)m,\delta}$ problem

$$F \cdot \mathbf{v} = \mathbf{0} \mod q, \|\mathbf{v}\| \leq \beta, F \in \mathbb{Z}_q^{n\times N(1+2l)m} \tag{1}$$

  Here F satisfies it can be parsed as $F = [F_1||F_2||\dots||F_N]$, where $F_{i^*} = [A_{i^*}||U_1^{(0)}||U_1^{(1)}||\dots||U_l^{(0)}||U_l^{(1)}]$ for $i^* \in \{1, 2, \dots, N\}$, and $A_{i^*}, U_j^{(b_j)} \in \mathbb{Z}_q^{n\times m}$ for $i^* \in \{1, 2, \dots, N\}$, $j \in \{1, 2, \dots, l\}$ and $b_j \in \{0, 1\}$.
- Pick a vector $\mathbf{u} = (u_1, u_2, \dots, u_N) \in \{1, 2, \dots, max\}$ and set a ring $R_{\mathbf{u}} = \{u_1, u_2, \dots, u_N\}$.
- Construct a list $\mathscr{L}$ that is empty initially. For the index i, if $1 \leq i \leq max$ and $i \notin R_{\mathbf{u}}$, run the $TrapGen$ algorithm to generate $A_i \in \mathbb{Z}_q^{n\times m}$ with the corresponding short basis $B_i \in \mathbb{Z}^{m\times m}$ for $\Lambda^{\perp}(A_i)$ and store the tuple $< i, A_i, B_i >$ in list $\mathscr{L}$. For $1 \leq i \leq max$ and $i = u_{i^*} \in R_{\mathbf{u}}$, set $A_i = A_{i^*} \in \mathbb{Z}_q^{n\times m}$.
- Set $A_i^{(b_i)} = U_i^{(b_i)}$, where $i \in \{1, 2, \dots, l\}$ and $b_i \in \{0, 1\}$.
- Select randomly $d+1$ short matrices $D_i \in \mathbb{Z}^{N(l+1)m\times m}$, fix $h_0 = 1 \in \mathbb{Z}_q$ and pick d uniform and random scalars $h_i \in \mathbb{Z}_q$.
- Invoke the $TrapGen$ algorithm to obtain $E \in \mathbb{Z}_q^{n\times m}$ with a short basis $B_E \in \mathbb{Z}^{m\times m}$ for $\Lambda_q^{\perp}(E)$.
- $\mathscr{C}$ guesses the target time period $t^* \in \{0, 1, 2, \dots, \tau - 1\}$ that $\mathscr{A}$ intends to attack and denotes it as $t^* = (t_1^*, t_2^*, \dots, t_l^*)$, where t_i^* is the value of the ith node of the path in the binary tree. For $i \in [l]$, set $A_i^{(t_i^*)} = U_i^{(t_i^*)}$, and let $A_{R_{\mathbf{u}},t^*} = [F_1^*||F_2^*||\dots||F_N^*]$, where $F_i^* = [A_i||A_1^{(t_1^*)}||\dots||A_l^{(t_l^*)}]$ for $i \in \{1, 2, \dots, N\}$.
- Set the set of public keys $\{A_1, A_2, \dots, A_{max}\}$ and system parameters $< A_1^{(0)}, A_1^{(1)}, \dots, A_l^{(0)}, A_l^{(1)}, C_0 = A_{R,t^*}D_0 + h_0E \mod q, C_1 = A_{R,t^*}D_1 + h_1E \mod q, \dots, C_d = A_{R,t^*}D_d + h_dE \mod q >$, and send them to $\mathscr{A}$.

– **Query**: $\mathscr{A}$ proceeds the following queries adaptively.

- **Corruption query:** For a time period $t = (t_1, t_2, \dots, t_l)$ and a sign user $pk_i = A_i$, if $pk_i \notin R_{\mathbf{u}}$ or this user pk_i is the challenged signer, $\mathscr{C}$ aborts. If $t \leq t^*$, $\mathscr{C}$ aborts. Otherwise, assume $k \leq l$ is the minimum index such that $t_k \neq t_k^*$, the challenger $\mathscr{C}$ computes the key T_{t_k} for the node t_k using the trapdoor $T_{A_k^{(t_k)}}$ as follows:

  $T_{t_k} \leftarrow ExtBasis(G||A_k^{(t_k)}, T_{A_k^{(t_k)}})$, where $G = [A_i||A_1^{(t_1)}||\dots||A_{k-1}^{(t_{k-1})}]$.

  Then $\mathscr{C}$ computes all keys in $sk_{i,t}$ as the real KeyUpdate algorithm.
- **Signing query:** For the time period $t = (t_1, t_2, \dots, t_l)$, the ring R, the message $\mathbf{m} \in \{0, 1\} \times \{0, 1\}^d$, the signer pk_i with the index i,

* If $R = R_{\mathbf{u}}$, $\mathscr{C}$ constructs the matrix $A_{R,t} = [F_1||F_2||\ldots||F_N]$, where $F_i = [A_i||A_1^{(t_1)}||\ldots||A_l^{(t_l)}]$ for $i \in \{1, 2, \ldots, N\}$. $\mathscr{C}$ computes $D_{\mathbf{m}} = \sum_{i=0}^{d}(-1)^{\mathbf{m}[i]}D_i$ and $h_{\mathbf{m}} = \sum_{i=0}^{d}(-1)^{\mathbf{m}[i]}h_i$. If $\sum_{i=0}^{d}(-1)^{\mathbf{m}[i]}h_i = 0$, $\mathscr{C}$ aborts. Otherwise, $\mathscr{C}$ constructs the matrix $F' = [A_{R,t}||A_{R,t}D_{\mathbf{m}} + h_{\mathbf{m}}E]$, and finds a short vector $\mathbf{e} \in \Lambda_q^{\perp}(F')$ using B_E.
* Otherwise, if the tuple $< i, A_i, B_i >$ belongs to the list $\mathscr{L}$, $\mathscr{C}$ constructs a matrix $F_j = [A_{R_j,t}|| \sum_{i=0}^{d}(-1)^{\mathbf{m}[i]}C_i]$ for a ring R_j and returns $\mathbf{e}_j \to GenSamplePre(F_j, A_i, B_i, \mathbf{0}, \delta)$ to $\mathscr{A}$.
* Otherwise, the challenger a user with index $k \in R_j$ such that the tuple $< k, A_k, B_k >$ belongs to the list $\mathscr{L}$. And $\mathscr{C}$ returns $\mathbf{e}'_j \to GenSamplePre(F_j, A_k, B_k, \mathbf{0}, \delta)$ to $\mathscr{A}$.

– **Challenge**: The adversary $\mathscr{A}$ outputs a forgery $< \sigma^*, \mathbf{m}^*, R^*, A_i^*, t^* >$. If $R^* \neq R_{\mathbf{u}}$, $\mathscr{C}$ aborts. Otherwise, $\mathscr{C}$ processes the following steps:
  - Compute $D_{\mathbf{m}^*} = \sum_{i=0}^{d}(-1)^{\mathbf{m}^*_{[i]}}D_i$ and $h_{\mathbf{m}^*} = \sum_{i=0}^{d}(-1)^{\mathbf{m}^*_{[i]}}h_i$;
  - If $h_{\mathbf{m}^*} \neq 0 \mod q$, $\mathscr{C}$ aborts.
  - Separate $\sigma^* \in \mathbb{Z}^{(N(l+1)+1)m}$ into $\sigma_1^* \in \mathbb{Z}^{N(l+1)m}$ and $\sigma_2^* \in \mathbb{Z}^m$, such that $\sigma^* = \begin{pmatrix} \sigma_1^* \\ \sigma_2^*. \end{pmatrix}$
  - Return $\mathbf{e}^* = \sigma_1^* + D_{\mathbf{m}^*}\sigma_2^* \in \mathbb{Z}^{N(l+1)m}$.

Analysis: $C_{\mathbf{m}^*} = \sum_{i=0}^{d}(-1)^{\mathbf{m}^*_{[i]}}C_i = \sum_{i=0}^{d}(-1)^{\mathbf{m}^*_{[i]}}(A_{R^*,t^*}D_i + h_iE)$. If $h_{\mathbf{m}^*} = 0 \mod q$, we have that $C_{\mathbf{m}^*} = A_{R^*,t^*}D_{\mathbf{m}^*}$ and then $A_{R^*,t^*}\mathbf{e}^* = A_{R^*,t^*}(\sigma_1^* + D_{\mathbf{m}^*}\sigma_2^*) = [A_{R^*,t^*}|A_{R^*,t^*}D_{\mathbf{m}^*}]\begin{pmatrix} \sigma_1^* \\ \sigma_2^* \end{pmatrix} = [A_{R^*,t^*}|C_{\mathbf{m}^*}]\mathbf{e}^* = [A_{R_{\mathbf{u}},t^*}|C_{\mathbf{m}^*}]\mathbf{e}^* = \mathbf{0} \mod q$. We can obtain the matrix F from $A_{R_{\mathbf{u}},t^*}$ by inserting the remain matrices $\{U_i^{(1-b_i)}\}_{i=1}^{l}$ for $b_i \in \{0, 1\}$. Then we insert vectors $\mathbf{0}$ into the corresponding positions of $\mathbf{e}^*$ to obtain the solution $\mathbf{v}$ of the equation (1). And we can get the fact that $F \cdot \mathbf{v} = \mathbf{0} \mod q$ and $\|\mathbf{v}\| = \|\mathbf{e}^*\|$. Thus, $\mathscr{C}$ solves the $SIS_{q,N(1+2l)m,\delta}$ problem.

5 Conclusion

This paper shows the first lattice-based ring signature scheme with forward security under the standard model. Our proposal combines lattice delegation techniques with a binary tree structure to realize a key evolution mechanism. Based on this one-way evolution mechanism, secret keys can be updated timely with generating nodes in the binary tree, which guarantees that the exposure of a current secret key can not threaten the past signatures. Moreover, our scheme is expected to be post-quantum secure due to its underlying security assumption on the hardness of the SIS problem in lattice theory.

References

1. US Department of Commerce, National Institute of Standards and Technology. Post-Quantum Cryptography (2020), https://csrc.nist.gov/projects/post-quantum-cryptography/
2. Abdalla, M., Reyzin, L.: A new forward-secure digital signature scheme. In: Advances in Cryptology - ASIACRYPT, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7. vol. 1976, pp. 116–129 (2000), https://doi.org/10.1007/3-540-44448-3_10
3. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Advances in Cryptology - CRYPTO, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19. vol. 6223, pp. 98–115 (2010), https://doi.org/10.1007/978-3-642-14623-7_6
4. Ajtai, M.: Generating hard instances of the short basis problem. In: Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15. vol. 1644, pp. 1–9 (1999), https://doi.org/10.1007/3-540-48523-6_1
5. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: 26th International Symposium on Theoretical Aspects of Computer Science, STACS, February 26-28, 2009, Freiburg, Germany. vol. 3, pp. 75–86 (2009), https://doi.org/10.4230/LIPIcs.STACS.2009.1832
6. Anderson, R.: Two remarks on public key cryptology. Technical report, University of Cambridge, Computer Laboratory (2002)
7. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In: Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19. vol. 1666, pp. 431–448 (1999), https://doi.org/10.1007/3-540-48405-1_28
8. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. J. Cryptol. **22**(1), 114–138 (2009), https://doi.org/10.1007/s00145-007-9011-9
9. Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. In: Advances in Cryptology - EUROCRYPT, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26. vol. 3494, pp. 440–456 (2005), https://doi.org/10.1007/11426639_26
10. Boyen, X.: Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In: Public Key Cryptography - PKC, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28. vol. 6056, pp. 499–517 (2010), https://doi.org/10.1007/978-3-642-13013-7_29
11. Boyen, X., Haines, T.: Forward-secure linkable ring signatures. In: Information Security and Privacy - 23rd Australasian Conference, ACISP, Wollongong, NSW, Australia, July 11-13. vol. 10946, pp. 245–264 (2018), https://doi.org/10.1007/978-3-319-93638-3_15
12. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Advances in Cryptology - EUROCRYPT, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8. vol. 2656, pp. 255–271 (2003), https://doi.org/10.1007/3-540-39200-9_16
13. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Advances in Cryptology - EUROCRYPT, 29th Annual In-

ternational Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3. vol. 6110, pp. 523–552 (2010), https://doi.org/10.1007/978-3-642-13190-5_27

14. Dodis, Y., Katz, J., Xu, S., Yung, M.: Key-insulated public key cryptosystems. In: Advances in Cryptology - EUROCRYPT, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2. vol. 2332, pp. 65–82 (2002), https://doi.org/10.1007/3-540-46035-7_5
15. Duc, D.N., Cheon, J.H., Kim, K.: A forward-secure blind signature scheme based on the strong RSA assumption. In: Information and Communications Security, 5th International Conference, ICICS, Huhehaote, China, October 10-13. vol. 2836, pp. 11–21 (2003), https://doi.org/10.1007/978-3-540-39927-8_2
16. Feng, H., Liu, J., Wu, Q., Li, Y.: Traceable ring signatures with post-quantum security. In: Topics in Cryptology - CT-RSA - The Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, February 24-28. vol. 12006, pp. 442–468 (2020), https://doi.org/10.1007/978-3-030-40186-3_19
17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proc. ACM Symposium on Theory of Computing. pp. 197–206 (2008), https://doi.org/10.1145/1374376.1374407
18. Gritti, C., Susilo, W., Plantard, T.: Logarithmic size ring signatures without random oracles. IET Inf. Secur. **10**(1), 1–7 (2016), https://doi.org/10.1049/iet-ifs.2014.0428
19. Hu, M., Liu, Z.: Lattice-based linkable ring signature in the standard model. IACR Cryptol. ePrint Arch. p. 101 (2022), https://eprint.iacr.org/2022/101
20. Itkis, G., Reyzin, L.: Forward-secure signatures with optimal signing and verifying. In: Advances in Cryptology - CRYPTO, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23. vol. 2139, pp. 332–354 (2001), https://doi.org/10.1007/3-540-44647-8_20
21. Lai, Y., Chang, C.: A simple forward secure blind signature scheme based on master keys and blind signatures. In: 19th International Conference on Advanced Information Networking and Applications (AINA), 28-30 March, Taipei, Taiwan. pp. 139–144. IEEE Computer Society (2005), https://doi.org/10.1109/AINA.2005.63
22. Le, H.Q., Duong, D.H., Susilo, W., Tran, H.T.N., Trinh, V.C., Pieprzyk, J., Plantard, T.: Lattice blind signatures with forward security. In: Information Security and Privacy - 25th Australasian Conference, ACISP, Perth, WA, Australia, November 30 - December 2. vol. 12248, pp. 3–22 (2020), https://doi.org/10.1007/978-3-030-55304-3_1
23. Libert, B., Yung, M.: Dynamic fully forward-secure group signatures. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS, Beijing, China, April 13-16. pp. 70–81. ACM (2010), https://doi.org/10.1145/1755688.1755698
24. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Forward-secure group signatures from lattices. In: Post-Quantum Cryptography - 10th International Conference, PQCrypto, Chongqing, China, May 8-10. vol. 11505, pp. 44–64 (2019), https://doi.org/10.1007/978-3-030-25510-7_3
25. Liu, J.K., Wong, D.S.: Solutions to key exposure problem in ring signature. Int. J. Netw. Secur. **6**(2), 170–180 (2008), http://ijns.jalaxy.com.tw/contents/ijns-v6-n2/ijns-2008-v6-n2-p170-180.pdf

26. Liu, J.K., Yuen, T.H., Zhou, J.: Forward secure ring signature without random oracles. In: Information and Communications Security - 13th International Conference, ICICS, Beijing, China, November 23-26. vol. 7043, pp. 1–14 (2011), https://doi.org/10.1007/978-3-642-25243-3_1
27. Miaomiao Tian, L.H., Yang, W.: Efficient lattice-based ring signature scheme. Chinese journal of computers **35**(4), 712–718 (2012)
28. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Advances in Cryptology - EUROCRYPT - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19. vol. 7237, pp. 700–718 (2012), https://doi.org/10.1007/978-3-642-29011-4_41
29. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007), https://doi.org/10.1137/S0097539705447360
30. Nakanishi, T., Hira, Y., Funabiki, N.: Forward-secure group signatures from pairings. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **93-A**(11), 2007–2016 (2010), https://doi.org/10.1587/transfun.E93.A.2007
31. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Advances in Cryptology - ASIACRYPT, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13. vol. 2248, pp. 552–565 (2001), https://doi.org/10.1007/3-540-45682-1_32
32. Shacham, H., Waters, B.: Efficient ring signatures without random oracles. In: Public Key Cryptography - PKC, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20. vol. 4450, pp. 166–180 (2007), https://doi.org/10.1007/978-3-540-71677-8_12
33. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November. pp. 124–134. IEEE Computer Society (1994), https://doi.org/10.1109/SFCS.1994.365700
34. Tang, F., Li, H.: Ring signatures of constant size without random oracles. In: Information Security and Cryptology - 10th International Conference, Inscrypt, Beijing, China, December 13-15. vol. 8957, pp. 93–108 (2014), https://doi.org/10.1007/978-3-319-16745-9_6
35. Wang, J., Sun, B.: Ring signature schemes from lattice basis delegation. In: Information and Communications Security - 13th International Conference, ICICS, Beijing, China, November 23-26. vol. 7043, pp. 15–28 (2011), https://doi.org/10.1007/978-3-642-25243-3_2
36. Yu, J., Hao, R., Kong, F., Cheng, X., Fan, J., Chen, Y.: Forward-secure identity-based signature: Security notions and construction. Inf. Sci. **181**(3), 648–660 (2011), https://doi.org/10.1016/j.ins.2010.09.034