10-2020

Hierarchical identity-based signature in polynomial rings

Zhichao YANG

Dung H. DUONG

Willy SUSILO

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Chao LI

See next page for additional authors

Citation

YANG, Zhichao; DUONG, Dung H.; SUSILO, Willy; YANG, Guomin; LI, Chao; and CHEN, Rongmao. Hierarchical identity-based signature in polynomial rings. (2020). *Computer Journal*. 63, (10), 1490-1499.
Available at: https://ink.library.smu.edu.sg/sis_research/7328

Advance Access publication on 28 April 2020
doi: 10.1093/comjnl/bxaa033

Hierarchical Identity-Based Signature in Polynomial Rings

Zhichao Yang[1], Dung H. Duong[2], Willy Susilo[2], Guomin Yang[2], Chao Li[1,*] and Rongmao Chen[1]

[1]*College of Computer, National University of Defence Technology, Hunan, 410000 Changsha, P. R. China*

[2]*School of Computing and Information Technology, University of Wollongong, 2500 Wollongong, Australia*

*****Corresponding author: academic_lc@163.com*

Hierarchical identity-based signature (HIBS) plays a core role in a large community as it significantly reduces the workload of the root private key generator. To make HIBS still available and secure in post-quantum era, constructing lattice-based schemes is a promising option. In this paper, we present an efficient HIBS scheme in polynomial rings. Although there are many lattice-based signatures proposed in recent years, to the best of our knowledge, our HIBS scheme is the first ring-based construction. In the center of our construction are two new algorithms to extend lattice trapdoors to higher dimensions, which are non-trivial and of independent interest. With these techniques, the security of the new scheme can be proved, assuming the hardness of the Ring-SIS problem. Since operations in the ring setting are much faster than those over integers and the new construction is the first ring-base HIBS scheme, our scheme is more efficient and practical in terms of computation and storage cost when comparing to the previous constructions.

Keywords: HIBS; Lattice; Ring-SIS; Post-Quantum.

Received 26 August 2019; Revised 19 January 2020; Editorial Decision 5 March 2020
Handling editor: Joseph Liu

1. INTRODUCTION

Generally speaking, the public verification key in a classical signature scheme is not directly linked to the user's identity. A receiver needs to firstly obtain the sender's verification key to verify the received signature. In this case, a public-key infrastructure is required, which usually causes the certificate management problem. In 1984, Shamir introduced the notion of identity-based cryptography and proposed an identity-based signature (IBS) in [1]. In identity-based cryptography, the receiver can easily deduce the public key based on the sender's public identity, e.g. an email address. Hence, it eliminates the problem caused by the public-key infrastructure. However, for the sender, the private signing key can only be generated by a third party called private-key generator (PKG), which takes the master secret key and the sender's identity as inputs and returns the private key. It means that the sender has to authenticate himself to the PKG through some secure channels, and the system will be completely broken if the master secret key is revealed to the attacker.

In practice, the IBS scheme [2–6] is efficient when there are only a few users in the system. However, the PKG will be overloaded when it comes to a large network since the PKG has to establish secure channels for each sender and issues lots of user private keys. Besides, proofs of each identity have to be verified, which makes PKG a bottleneck. Hierarchical identity-based signature (HIBS) is a method to solve these problems. In an HIBS scheme, a lower-level PKG acquires delegation from the higher-level PKG to generate private key and authenticate the identity. Hence, it effectively distributes the root PKG's workload. Moreover, an adversary with lower-level private key cannot recover any higher-level private key, which is called damage control. Gentry and Silverberg proposed the first HIBS scheme in [7] where they constructed the scheme from pairings. Currently, most signature schemes rely on the hardness of integer factorization problem or discrete logarithm problem [8]. In 1999, Shor [9] proposed a powerful quantum algorithm, which can solve those hard problems on quantum computers in polynomial time. It shows that, in the quantum era, the schemes based on classic hard problems are not secure any more. As alternatives, schemes based on decoding problems, solving multivariate equation problem and lattice problems are proposed, since it has been believed that these problems [10] can

still resist the quantum computing attack. In [11], Ajtai proved that if an adversary solves an average-case problem underlying the lattice-based cryptography, he can also solve a related worst-case problem in random lattices. That is an interesting property which has not been found elsewhere in cryptography. It makes lattice-based schemes the most popular ones among those post-quantum cryptography candidates. Since then, many signatures based on lattice problems have been constructed. Some are defined in the random oracle model (ROM) [12, 13] and others are considered in the standard model [14, 15].

Rückert [16] first proposed two HIBS schemes from lattices whose security can be proved in standard model, and both of them are bonsai tree signature schemes. Liu *et al.* [17] constructed an IBS scheme on lattice and then obtained an HIBS scheme without security proof by extending it. Later, Tian *et al.* [18] gave an HIBS scheme over the standard short integer solution (SIS) assumption [11] and showed that its security can be proved in the standard model. In 2013, Tian *et al.* [19] introduced another HIBS scheme on lattices. The new scheme has smaller secret key size and signature size compared with other lattice-based HIBS schemes. All of those schemes are defined on integer ring, and to the best of our knowledge, there is no lattice-based HIBS scheme constructed on polynomial ring yet.

Contributions Because of better performance, almost all the lattice-based schemes submitted to NIST's post-quantum project are defined in polynomial rings. In this paper, for the first time, we construct a lattice-based HIBS scheme in polynomial rings. The efficiency of multiplication in polynomial ring makes our new scheme much more practical than those defined in integer rings. Based on the $\boldsymbol{g}$-trapdoor introduced by Miaccianco *et al.* [20] and its ring version [21], we also propose two algorithms called $\mathsf{ExtLeft}$ and $\mathsf{ExtRight}$. Each algorithm generates a new $\boldsymbol{g}$-trapdoor for vector $\boldsymbol{f}_{\mathsf{id}}$ when it accesses to the $\boldsymbol{g}$-trapdoor of its sub-vector. We use the algorithm $\mathsf{ExtLeft}$ in the real system to obtain the user's secret key, while the other algorithm is only used by the simulator in the security proof to simulate the user's secret key for all queried identities.

To analyse the security of the new HIBS scheme, we rewrite the polynomial vector into matrix form and show that the Ring-SIS problems can be solved easily when one only accesses to a $\boldsymbol{g}$-trapdoor in integer ring. Finally, our scheme is proved to be secure under chosen message and chosen-identity attack in the ROM, assuming the hardness of Ring-$\mathrm{SIS}_{n,m,q,\beta}$ problem. As the underlying Ring-SIS problem is believed to be as hard as some worst-case approximation problems on lattices. The security of the signature scheme rests upon lattice problems in worst-case, which cannot be handled in polynomial time, even by quantum computers. That makes our HIBS scheme available and secure in the post-quantum era.

Future works Although, the underlying Ring-SIS problems are believed to be hard in quantum era, security proofs in this paper are only considered under the classical ROM. It will be ideal if our scheme can be proved secure when the adversary has the ability to query the hash function on a superposition of inputs (i.e. security in the quantum random oracle model). That is an interesting and challenge problem, and we will handle it in our future works. Besides, how to construct HIBS schemes based on the Module-SIS/LWE problems is another meaningful but not non-trivial task. To do that, one has to first propose new trapdoors for polynomial matrixes and then, construct new extension algorithms. We will leave the above research questions as our future works.

Organizations We organize the rest paper as follows. Section 2 describes some useful notations and definitions. Lattice and related hard problems will be introduced in Section 3. Section 4 will define some important sampling algorithms. A new ring-based HIBS scheme is proposed in Section 5, security proofs are also contained in it. Finally, conclusion is given in Section 6.

2. PRELIMINARIES

2.1. Notations

Let $\mathbb{Z}$ be the integer ring, we denote $\mathbb{Z}_q$ as the residue class ring $\mathbb{Z}/q\mathbb{Z}$. Bold letters are used to represent vectors in column notation. For a column vector $\boldsymbol{v}$, its i-th entry is denoted by $\boldsymbol{v}_i$. The capital letter represents matrix and $\boldsymbol{A}_i$ is the i-th column of matrix $\boldsymbol{A}$.

For any constant c, function $\epsilon(n)$ is called $\mathsf{negligible}$ if $\epsilon(n) = o\left(n^{-c}\right)$ and the probability $1-\epsilon(n)$ is said to be $\mathsf{overwhelming}$. The notion $z \leftarrow_{\$} D$ means that the variable z is sampled from the distribution D, and the probability of $z = x$ is denoted by $D(x)$. For two distributions D_1 and D_2 over the same discrete domain X, they are said to be statistically close with respect to n if $\frac{1}{2}\sum_{x\in X}|D_1(x) - D_2(x)|$ is negligible in n. $\log(\cdot)$ is the base 2 logarithms, respectively.

2.2. Hierarchical identity-based signature

Firstly, we give some introductions of HIBS scheme and the security model. In HIBS scheme, a user's identity at depth $l-1$ is regarded as a vector $\mathsf{id} = (\mathsf{id}_1, \cdots, \mathsf{id}_{l-1})$ of dimensions $l-1$ and a child identity $\mathsf{id}|\mathsf{id}_l$ is defined by $\mathsf{id}|\mathsf{id}_l = (\mathsf{id}_1, \cdots, \mathsf{id}_{l-1}, \mathsf{id}_l)$. The definition of HIBS scheme is described below.

DEFINITION 2.1. *An HIBS scheme* Π *consists of four PPT algorithms and a deterministic algorithm that work as follows:*

- $\mathsf{Setup}(1^n, d)$*:* Setup *will output a mask public key* mpk *and a mask secret key* msk *when it receives the security parameter* 1^n *and the maximum hierarchy depth is* d*.*
- $\mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$*: based on the mask key pair* $(\mathsf{mpk}, \mathsf{msk})$ *and an arbitrary identity* id*. It outputs the user's key pair* $\left(\mathsf{usk}_{\mathsf{id}}, \mathsf{upk}_{\mathsf{id}}\right)$ *associated with* id*.*

- $\mathsf{Derive}\left(\mathsf{upk}_{\mathsf{id}}, \mathsf{usk}_{\mathsf{id}}, \mathsf{id}|\mathsf{id}_l\right)$*: given a user key pair* $\left(\mathsf{upk}_{\mathsf{id}}, \mathsf{usk}_{\mathsf{id}}\right)$ *for identity* id*. The algorithm generates a user key pair* $\left(\mathsf{usk}_{\mathsf{id}|\mathsf{id}_l}, \mathsf{upk}_{\mathsf{id}|\mathsf{id}_l}\right)$ *for the child identity* $\mathsf{id}|\mathsf{id}_l$.
- $\mathsf{Sign}(\mu, \mathsf{id}, \mathsf{usk}_{\mathsf{id}})$*: the inputs of the algorithm* Sign *are a message* μ*, an identity* id *and its related user secret key* $\mathsf{usk}_{\mathsf{id}}$*. A signature* δ *is returned.*
- $\mathsf{Ver}(\mathsf{upk}_{\mathsf{id}}, \mathsf{id}, \mu, \delta)$*: take* $\mathsf{upk}_{\mathsf{id}}$, id, μ *and* δ *as input,* Ver *returns* 1 *if* δ *is valid and outputs* 0 *otherwise.*

For correctness, the scheme requires that the equation

$$\mathsf{Ver}(\mathsf{upk}_{\mathsf{id}}, \mathsf{id}, \mu, \mathsf{Sign}(\mu, \mathsf{id}, \mathsf{usk}_{id})) = 1,$$

will always hold for every n, id and $\mu \in \{0,1\}^*$.

Security As in the security model in identity-based cryptography [2], the adversary $\mathcal{A}$ can access to the signature oracle and the key extraction oracle adaptively to extract any identities except the parent identities of the challenge identity. In this work, we mainly focus on the selective-identity security, which requires $\mathcal{A}$ to choose the challenge identity before getting the master public key.

The following game defines the existential unforgeability against selective identity and chosen message attack, which is played between a challenge $\mathcal{C}$ and an adversary $\mathcal{A}$.

- $\mathsf{Setup}(1^n, d)$: $\mathcal{C}$ generates a fresh mask key pair by $(\mathsf{mpk}, \mathsf{msk}) \leftarrow_{\$} \mathsf{Setup}(1^n)$ and set d be the maximum hierarchy depth.
- $\mathcal{A}$ selects a challenge identity id^*, then issues the following types of queries adaptively.
  - $\mathsf{Extract}(\cdot)$: $\mathcal{A}$ chose an identity $\mathsf{id} \in \{0,1\}^*$, which is not the prefix of the id^*. Take id and master key pair as input, this oracle returns a secret signing key $\mathsf{usk}_{\mathsf{id}}$.
  - $\mathsf{Sign}(\cdot)$: when $\mathcal{A}$ issues a query on a message μ and an id, $\mathcal{C}$ returns a signature by $\delta \leftarrow_{\$} \mathsf{Sign}\left(\mu, \mathsf{id}, \mathsf{usk}_{\mathsf{id}}\right)$.
- $\mathsf{Forgery}$: the adversary $\mathcal{A}$ outputs a message μ^* and a forge signature δ^*. The adversary wins the game if $\mathsf{Ver}\left(\mathsf{upk}_{\mathsf{id}}, \mathsf{id}^*, \mu^*, \delta^*\right) = 1$ and Sign queries list never contains $\left(\mathsf{upk}_{\mathsf{id}}, \mathsf{id}^*, \mu^*, \delta^*\right)$.

The advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{UF\text{-}sID\text{-}CMA}}(n)$ of $\mathcal{A}$ is defined as the probability that the adversary success to forgery a valid signature. If $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{UF\text{-}sID\text{-}CMA}}(n)$ is negligible in n for any polynomial-time adversary $\mathcal{A}$, we regard that the HIBS scheme is $\mathsf{UF\text{-}sID\text{-}CMA}$ secure.

3. LATTICES

3.1. Basic notions

An integer lattice $\mathcal{L}$ is a discrete subgroup in $\mathbb{R}^m$ generated by several linear independent vectors $\boldsymbol{B} = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_k] \in \mathbb{R}^{m \times k}$, where m, and k are positive integers and $k \leq m$,

$$\mathcal{L}(\boldsymbol{B}) = \left\{ \sum_{i=0}^{k-1} \boldsymbol{b}_i x_i : x_i \in \mathbb{Z} \right\},$$

and $\boldsymbol{B}$ is called a basis matrix of this lattice.

For a polynomial $f(x) \in \mathbb{Z}[x]$ of degree n, $R = \mathbb{Z}[x]/f(x)$ denotes a polynomial ring. The definition of integer lattice can also be generalized to ring setting. In this paper, we also consider the following ideal lattices [21]. For an integer modulus q, $R_q = \mathbb{Z}_q[x]/f(x)$. A polynomial $u \in R_q$ and a vector $\boldsymbol{a} \in R_q^m$, lattices are defined as

$$\mathcal{L}_q(\boldsymbol{a}) = \{\boldsymbol{x} \in R^m : \exists s \in R_q, \text{s.t.} \boldsymbol{a}s = \boldsymbol{x} \mod q\},$$

$$\mathcal{L}_q^{\perp}(\boldsymbol{a}) = \{\boldsymbol{x} \in R^m : \boldsymbol{a}^t \boldsymbol{x} = 0 \mod q\},$$

$$\mathcal{L}_q^{u}(\boldsymbol{a}) = \{\boldsymbol{x} \in R^m : \boldsymbol{a}^t \boldsymbol{x} = u \mod q\}.$$

Matrix representation For any polynomial vector $\boldsymbol{e} \in R_q^m$, where $\boldsymbol{e}_i = \sum_{j=0}^{n-1} e_{ij} x^j$ and $i = 0, \ldots, m-1$, $\boldsymbol{e}$ can also be represented by a matrix $\boldsymbol{E} \in \mathbb{Z}_q^{m \times n}$,

$$\boldsymbol{E} = \begin{bmatrix} e_{0,0} & e_{0,1} & \cdots & e_{0,n-1} \\ e_{1,0} & e_{1,1} & \cdots & e_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m-1,0} & e_{m-1,1} & \cdots & e_{m-1,n-1} \end{bmatrix} = \left[\hat{\boldsymbol{e}}_0, \hat{\boldsymbol{e}}_1, \ldots, \hat{\boldsymbol{e}}_{n-1}\right], \tag{1}$$

where $\hat{\boldsymbol{e}}_j$ is the j-column vector of the integer matrix $\boldsymbol{E}$.

For a polynomial $a \in R_q$ and $a = \sum_{i=0}^{n-1} a_i x^i$, its infinite norm $||\cdot||_\infty$ is defined by

$$||a||_\infty = \max_{0 \leq i \leq n-1} \{|a_i|\},$$

$|a_i|$ is the absolute value of the integer a_i. Similarly, the infinite norm of a vector $\boldsymbol{a} = (\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{m-1}) \in R_q^m$ is given by

$$||\boldsymbol{a}||_\infty = \max_{0 \leq i \leq m-1} \{||\boldsymbol{a}_i||_\infty\}.$$

Gaussian measures $||\boldsymbol{x}||_2$ denotes the L_2 norm of vector $\boldsymbol{x} \in \mathbb{Z}^m$. ρ represents n-dimensional Gaussian function, which is defined by

$$\rho_s(\boldsymbol{x}) = \exp\left(-\pi \cdot ||\boldsymbol{x}||_2^2 / s^2\right).$$

$\mathcal{D}_{\mathbb{Z}^m, s}$ represents the discrete gaussian distribution over $\mathbb{Z}^m$ with deviation s. For an arbitrary vector $\boldsymbol{x} \in \mathbb{Z}^m$, the value of $\mathcal{D}_{\mathbb{Z}^m, s}(\boldsymbol{x})$ can be calculated by

$$\mathcal{D}_{\mathbb{Z}^m, s}(\boldsymbol{x}) = \rho_s(\boldsymbol{x}) / \rho_s\left(\mathbb{Z}^m\right),$$

where $\rho_s(\mathbb{Z}^m) = \sum_{\boldsymbol{y}\in\mathbb{Z}^m} \rho_s(\boldsymbol{y})$. As shown in [20, 22], the distribution $\mathcal{D}_{\mathbb{Z}^m,s}$ satisfies the following property.

LEMMA 3.1. ([20, 22]). *For any parameter $s > 0$, we have*

$$Pr_{\boldsymbol{x}\leftarrow D_{\mathbb{Z}^m,s}}[||\boldsymbol{x}||_2 > s\sqrt{m}] \leq 2^{-m}.$$

For $z \in R_q$, $z \leftarrow \mathcal{D}_s$ represents that it first generates a vector $\boldsymbol{z}$ from $\mathcal{D}_{\mathbb{Z}^n,s}$ and then constructs a polynomial in R_q as

$$z = \sum_{i=1}^{n} z_i x^{i-1} \mod q.$$

$\boldsymbol{B} \leftarrow \mathcal{D}_s^{m\times k}$ is to sample each element in matrix $\boldsymbol{B} \in R_q^{m\times k}$ from $\mathcal{D}_s$ independently.

3.2. Ring-SIS

In [11], Ajtai first proposed the *shortest integer solution* (SIS) problem, which is used in constructing one-way and collision-resistance hash functions. As a ring-based analogue of SIS problem, the Ring-SIS problem was introduced by Micciancio [23]. The Ring-SIS problem is parameterized by a ring R of degree n and a quotient ring $R_q = R/qR$ where q is a positive integer. The parameter β is a real number bound for 'short' solutions, and m is the number of samples.

DEFINITION 3.1. ([24–27]). *Sample m elements a_i from R_q uniformly random and construct a vector $\boldsymbol{a} \in R_q^m$. The Ring-SIS$_{n,m,q,\beta}$ problem is to find $\boldsymbol{z} \in R_q^m \setminus \{\boldsymbol{0}\}$ with norm $||\boldsymbol{z}||_\infty \leq \beta$ that satisfies*

$$\boldsymbol{a}^t\boldsymbol{z} = \sum_{i=1}^{m} a_i z_i = 0 \mod q.$$

If the polynomial ring R is changed into integer ring $\mathbb{Z}$, then the resulting problem will be the classical SIS problem. Compared with the SIS problem, the Ring-SIS problem is rather efficient and compact. In SIS problem, the parameter m is set to be $n\log q$ while it is only $\log q$ in the ring setting. In addition, each multiplication in R_q can be calculated in quasi-linear $\tilde{O}(n)$ time by using the FFT-like techniques. Thus, the total time costed in computing $\boldsymbol{a}^t\boldsymbol{z}$ is also quasi-linear. Moreover, the Ring-SIS$_{n,m,q,\beta}$ problem is at least as hard as SVP$_\gamma^\infty$ on ideal lattices in R [24], if $m > \frac{\log q}{\log(2\beta)}$, $\gamma = 16\beta mn\log^2 n$, and $q \geq \frac{\gamma\sqrt{n}}{4\log n}$.

3.3. Trapdoors for lattices

Primitive vector Let q, and k be positive parameters; the primitive vector $\boldsymbol{g} \in \mathbb{Z}_q^m$ is defined as a vector such that $\gcd(g_0, g_1, \ldots, g_{m-1}, q) = 1$. The lattice $\mathcal{L}^\perp(\boldsymbol{g})$ is defined as

$$\mathcal{L}^\perp(\boldsymbol{g}) = \{\boldsymbol{x} \in \mathbb{Z}^m : \boldsymbol{x}^t\boldsymbol{g} = 0 \mod q\} \subset \mathbb{Z}^m.$$

Let $\boldsymbol{S} \in \mathbb{Z}^{m\times m}$ be a basis of $\mathcal{L}^\perp$, then, $\boldsymbol{S}\boldsymbol{g} = \boldsymbol{0} \mod q$ and $|\det(\boldsymbol{S})| = q$ [20]. $\tilde{\boldsymbol{S}}$ denotes the Gram–Schmidt orthogonalization of the matrix $\boldsymbol{S}$, and $||\tilde{\boldsymbol{S}}||$ represents the Gram–Schmidt norm of $\boldsymbol{S}$. From the results in [20], we can conclude the theorem below.

THEOREM 3.1. *$q \geq 2$, $m = \lceil\log_2 q\rceil$ are positive integers; there is a primitive vector $\boldsymbol{g} \in \mathbb{Z}_q^m$ such that*

- *$\boldsymbol{S}_m \in \mathbb{Z}^{m\times m}$ with $||\boldsymbol{S}||_2 \leq \{\sqrt{5}, \sqrt{m}\}$ is a basis of the lattice $\mathcal{L}^\perp(\boldsymbol{g})$. Moreover, when $q = 2^m$, we have $||\boldsymbol{S}||_2 = \sqrt{5}$*
- *Both $\boldsymbol{g}$ and $\boldsymbol{S}$ require little storage. In particular, they are sparse and highly structured.*
- *Let gaussian parameter $s \geq ||\tilde{\boldsymbol{S}}||\omega(\sqrt{\log m})$. There is a PPT algorithms SampleG$(\boldsymbol{g}, u, s)$ that outputs $\boldsymbol{z} \in \mathcal{L}_q^u(\boldsymbol{g})$, which is drawn from a distribution statistically close to $\mathcal{D}_{\mathbb{Z}^m,s}$. For a constant c, all the operations can be performed in quasilinear $O(m\log^c m)$.*

Ideal Lattices For a matrix $\boldsymbol{R}$, denote the largest singular value of $\boldsymbol{R}$ as $s_1(\boldsymbol{R}) = \max_{\boldsymbol{x}} ||\boldsymbol{R}\boldsymbol{x}|| = \max_{\boldsymbol{x}} ||\boldsymbol{R}^t\boldsymbol{x}||$, where $\boldsymbol{x}$ is an arbitrary unit vector. Inspired by the $\boldsymbol{g}$-trapdoors defined in $\mathbb{Z}^m$, the notion can also be extended to the ring setting R_q [21].

DEFINITION 3.2. *Let $\boldsymbol{a}$ be a vector sampled from R_q^m uniformly and $\boldsymbol{g}$ be a primitive vector in $\mathbb{Z}_q^k$. A $\boldsymbol{g}$-trapdoor for $\boldsymbol{a}$ is defined as a matrix $\boldsymbol{R} \in R_q^{m\times k}$ such that $\boldsymbol{a}^t\boldsymbol{R} = h\boldsymbol{g}^t \mod q$. h is an inventible element in R_q and it is referred as the tag of $\boldsymbol{R}$.*

In this paper, $\boldsymbol{g}$ is taken as a public constant vector, and $\boldsymbol{g}$-trapdoors simply denote trapdoors. The algorithm below is to generate a (pseudo)random vector $\boldsymbol{a} \in R_q^m$ together with a $\boldsymbol{g}$-trapdoor.

Algorithm 1: GenTrap$(\boldsymbol{a}_0, h, m, k, s)$

Input: $\boldsymbol{a}_0 \in R_q^{m-k}$, invertible $h \in R_q, m, k > 0$.

Output: A vector $\boldsymbol{a} \in R_q^m$, $\boldsymbol{g}$-trapdoor $\boldsymbol{R} \in R_q^{m\times k}$.

1: Sample a matrix $\boldsymbol{R}'$ from $\mathcal{D}_s^{(m-k)\times k}$.

2: Output:

$\boldsymbol{a} = [\boldsymbol{a}_0, h\boldsymbol{g}^t - \boldsymbol{a}_0^t\boldsymbol{R}']^t \in R_q^m$.

$\boldsymbol{R} = [\boldsymbol{R}'^t, \boldsymbol{I}]^t \in R_q^{m\times k}$.

REMARK. In fact, the inputs $\boldsymbol{a}_0$ and h can be chosen by picking $\boldsymbol{a}_0 \in R_q^l$ uniformly at random, and setting $h = 1$. The

correctness of Algorithm 1 is obviously, and the distribution of $\mathbf{R}'$ ensures the randomness of vector $\boldsymbol{a}$.

4. SAMPLING ALGORITHMS

4.1. Preimage sampling in Ring Setting

In fact, a primitive vector $\boldsymbol{g} \in \mathbb{Z}_q^m$ can also be regarded as a polynomial vector in the ring setting R_q^m. Then, the pair $\left(\boldsymbol{g}, u = \boldsymbol{g}^t \boldsymbol{z}\right)$ will be a Ring-SIS instance for $\boldsymbol{z} \in R_q^m, u \in R_q$ and $\boldsymbol{z}_i = \sum_{j=0}^{n-1} z_{i,j} x^j$ and $u = \sum_{i=0}^{n-1} u_i x^i$. The Eq.(1) shows that each vector in R_q^m is equivalent to a matrix in $\mathbb{Z}_q^{m \times n}$. With this notion, the equation $u = \boldsymbol{g}^t \boldsymbol{z} \mod q$ can be represented as

$$\left[u_0, \cdots, u_{n-1}\right] = \left[g_0, \cdots, g_{n-1}\right] \begin{bmatrix} z_{0,0} & \cdots & z_{0,n-1} \\ \vdots & \ddots & \vdots \\ z_{m-1,0} & \cdots & z_{m-1,n-1} \end{bmatrix}. \tag{2}$$

Then for $j = 0, 1 \ldots n-1$

$$u_j = \boldsymbol{g}^t \hat{\boldsymbol{z}}_j \mod q, \tag{3}$$

where $\hat{\boldsymbol{z}}_j \in \mathbb{Z}_q^m$ and u_j is the j-th coefficient of polynomial u. The Theorem 3.1 implies that we can preimage sample $\hat{\boldsymbol{z}}_j$ for Eq.(3) efficiently with each $\hat{\boldsymbol{z}}_j$ distributes close to $\mathcal{D}_{\mathbb{Z}^m,s}$ when the gaussian parameter s is large enough. Therefore, we can get each vector $\hat{\boldsymbol{z}}_j$ for $j = 0, \ldots, n-1$ and construct a preimage $\boldsymbol{z} \in R_q^m$. We describe these results in Theorem 4.1.

THEOREM 4.1. *Let $\boldsymbol{S}$ be the matrix mentioned in Theorem 3.1. Then for any parameter $s \geq ||\tilde{\boldsymbol{S}}||\omega\left(\sqrt{\log m}\right)$ and polynomial $u \in R_q$, we can sample a polynomial vector $\boldsymbol{z}$ from the set $\mathcal{L}_q^u(\boldsymbol{g})$ in quasilinear time. Moreover, each integer vector $\hat{\boldsymbol{z}}_j$ follows the distribution $\mathcal{D}_{\mathbb{Z}^m,s}$ except* negl(m) *statical distance.*

Proof. By the above description, the equation $u = \boldsymbol{g}^t \boldsymbol{z} \mod q$ is equivalent to n equations in Equation (3). Based on the Theorem 3.1, each integer vector $\hat{\boldsymbol{z}}_j$ can be sampled through the algorithm SampleG($\boldsymbol{g}$,u_j,s). Then, a polynomial vector in $\mathcal{L}_q^u(\boldsymbol{g})$ can be obtained and all the operations are performed in quasilinear $O(nm \cdot \log^c m)$ time. Moreover, for a parameter $s \geq ||\tilde{\boldsymbol{S}}||\omega\left(\sqrt{\log m}\right)$, the distribution of each vector $\hat{\boldsymbol{z}}_j$ close to $\mathcal{D}_{\mathbb{Z}^m,s}$. ■

REMARK. As is analysed in [21], we come to know that the distribution of each entry in our constructed vector $\boldsymbol{z}$ is statistically close to $\mathcal{D}_s$. In this case, the vector $\boldsymbol{z} \in R_q^m$ will follows the distribution $\mathcal{D}_s^m$ within negligible distance.

4.2. Preimage sampling for Ring-SIS

In this section, Algorithm 2 below shows that we can solve Ring-SIS$_{n,m,q,\beta}$ problem relative to $\boldsymbol{a}$ by using a $\boldsymbol{g}$-trapdoor. $\boldsymbol{R} \in R_q^{m \times k}$ denotes a trapdoor for vector $\boldsymbol{a} \leftarrow_{\$} R_q^m$. Let $\left(\boldsymbol{a}, u = \boldsymbol{a}^t \boldsymbol{z}\right)$ be a Ring-SIS instance with $\boldsymbol{z} \in R_q^m$ and $u \in R_q$. This naturally yields a preimage sampling algorithm to get vectors from lattice $\mathcal{L}_q^u(\boldsymbol{a})$.

Algorithm 2: SamplePre($\boldsymbol{a}, \boldsymbol{R}, u, s$).

Input: An oracle $\mathcal{O}(u,s)$ that samples from $\mathcal{L}_q^u(\boldsymbol{g})$.
- Vector $\boldsymbol{a} \in R_q^m$, parameters s.
- $\boldsymbol{g}$-trapdoor $\boldsymbol{R} \in R_q^{m \times k}$ for $\boldsymbol{a}$.

An invertible tag $h \in R_q$.
- A polynomial u in R_q.

Output: The preimage vector $\boldsymbol{z}$.

1: Get $\boldsymbol{z}' \leftarrow \mathcal{O}(u,s)$.

2: **return** $\boldsymbol{z} = \boldsymbol{R} h^{-1} \boldsymbol{z}' \in R_q^m$.

LEMMA 4.1. *Suppose each entry in $\boldsymbol{R}$ is sampled from $\mathcal{D}_s$ and $h = 1$, the Algorithm 2 can solve the Ring-SIS$_{n,m,q,\beta}$ problem if $s^2 \leq \beta / \left(\sqrt{kn} \cdot \omega(\log n)\right)$.*

Proof. It is easy to verify that

$$\boldsymbol{a}^t \boldsymbol{z} = \boldsymbol{a}^t \boldsymbol{R} \boldsymbol{z}' = \boldsymbol{g}^t \boldsymbol{z}' = u.$$

Section 4.1 indicates that each entry of $\boldsymbol{z}'$ is sampled from $\mathcal{D}_s$. Based on the results in [21, 28], the distribution of polynomial $\boldsymbol{z}_i$ is statistically close to $\mathcal{D}_{s'}$ for $i = 1, 2, \ldots, m$, where $s' = s^2 \sqrt{k} \cdot \omega(\log n)$. From Lemma 3.1, we can conclude that $||\boldsymbol{z}_i||_2 \leq s' \sqrt{n}$ with overwhelm probability. Hence,

$$||\boldsymbol{z}||_\infty \leq s^2 \sqrt{k} \cdot \omega(\log n) \cdot \sqrt{n},$$

which means that $||\boldsymbol{z}||_\infty \leq \beta$ will hold except negligible probability. $\boldsymbol{z}$ is indeed a solution of the Ring-SIS$_{n,m,q,\beta}$ problem.■

4.3. Two special extension algorithms

In [29], Agrawal *et al.* proposed a family of lattices and constructed two distinct trapdoors for them. Inspired by their ideal, we will introduce two new algorithms that can generate different $\boldsymbol{g}$-trapdoors for two kinds of lattices.

Algorithm ExtLeft Let $\boldsymbol{a}$ be a vector in R_q^m and $\boldsymbol{R} \in R_q^{m \times k}$ is the related $\boldsymbol{g}$-trapdoor. $\boldsymbol{b}$ denotes a random vector in R_q^l, then we define $20.\boldsymbol{f} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \in R_q^{m+l}$. Taking $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{R}$ and a

gauss parameter s as input, the algorithm $\mathsf{ExtLeft}(\boldsymbol{f}, \boldsymbol{R}, s)$ is to generate the trapdoor $\boldsymbol{R}_f \in R_q^{(m+l)\times k}$ for $\boldsymbol{f}$ such that

$$\boldsymbol{f}^t \boldsymbol{R}_f = h\boldsymbol{g}^t \mod q,$$

where h is an invertible polynomial and $\boldsymbol{g}$ is a primitive vector. More details can be found in Algorithm 3.

Algorithm 3: $\mathtt{ExtLeft}(\boldsymbol{f}, \boldsymbol{R}, s)$.

Input: A vector $\boldsymbol{f}$ in R_q^{m+l} defined as above.
- A $\boldsymbol{g}$-trapdoor of $\boldsymbol{a}$.
- A gauss parameter s.

Output: A $\boldsymbol{g}$-trapdoor $\boldsymbol{R}_f$ for vector $\boldsymbol{f}$.

1: Sample a random matrix $\boldsymbol{S} \leftarrow_{\$} \mathcal{D}_s^{l\times k}$.
2: Compute $\boldsymbol{y} = -\boldsymbol{b}^t\boldsymbol{S} + h\boldsymbol{g}^t \in R_q^k$.
3: Run $\boldsymbol{r}'_j \leftarrow \mathtt{SamplePre}(\boldsymbol{a}, \boldsymbol{R}, \boldsymbol{y}_j, s)$ for $j = 1, \ldots, k$.
4: Let $\boldsymbol{R}' = [\boldsymbol{r}'_1, \ldots, \boldsymbol{r}'_k]$.
5: Output $\boldsymbol{R}_f = [\boldsymbol{R}'^t, \boldsymbol{S}^t]^t \in R_q^{(m+l)\times k}$.

Clearly $\boldsymbol{f}^t\boldsymbol{R}_f = \boldsymbol{a}^t\boldsymbol{R}' + \boldsymbol{b}^t\boldsymbol{S}$, the algorithm SamplePre ensures that the equation $\boldsymbol{a}^t\boldsymbol{r}'_j = \boldsymbol{y}_j \mod q$ will hold for $j = 1, \ldots, k$ and $\boldsymbol{r}'_j \in \mathcal{L}_q^{\boldsymbol{y}_j}(\boldsymbol{a})$. We have

$$\boldsymbol{f}^t\boldsymbol{R}_f = \boldsymbol{y}^t + \boldsymbol{b}^t\boldsymbol{S} = -\boldsymbol{b}^t\boldsymbol{S} + h\boldsymbol{g}^t + \boldsymbol{b}^t\boldsymbol{S} = h\boldsymbol{g}^t.$$

Hence, $\boldsymbol{R}_f$ is indeed a $\boldsymbol{g}$-trapdoor for vector $\boldsymbol{f}$. Moreover, the distributions of all elements in $\boldsymbol{R}$ are statistically close to $\mathcal{D}_s$.

Algorithm ExtRight To introduce this algorithm, we first randomly sample some vectors $\boldsymbol{a} \leftarrow_{\$} R_q^m, \boldsymbol{c} \leftarrow_{\$} R_q^l, \boldsymbol{d} \leftarrow_{\$} R_q^r$ and run the algorithm GenTrap to get a vector $\boldsymbol{b} \in R_q^m$ and its trapdoor $\boldsymbol{R} \in R_q^k$. Define vector $\boldsymbol{f} \in R_q^{2m+l+r}$ as

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{c} \\ \boldsymbol{a}^t\boldsymbol{T} + \boldsymbol{b}^t\boldsymbol{H} \\ \boldsymbol{d} \end{bmatrix}, \tag{4}$$

where $\boldsymbol{T}$ is sampled from $\mathcal{D}_s^{m\times m}$ and $\boldsymbol{H}$ is an invertible matrix in $\mathbb{Z}_q^{m\times m}$. Take $\boldsymbol{f}$ and $\boldsymbol{R}$ as input, the algorithm ExtRight defined in Algorithm 4 will output a new $\boldsymbol{g}$-trapdoor $\boldsymbol{R}_f$ for $\boldsymbol{f}$.

Algorithm 4: $\mathtt{ExtRight}(\boldsymbol{f}, \boldsymbol{R}, s)$.

Input: A vector $\boldsymbol{f}$ mentioned in 4.
- A $\boldsymbol{g}$-trapdoor for $\boldsymbol{b}$.
- A gauss parameter s.

Output: A $\boldsymbol{g}$-trapdoor $\boldsymbol{R}_f$ for vector f.

1: $\boldsymbol{A} \leftarrow_{\$} \mathcal{D}_s^{m\times k}, \boldsymbol{C} \leftarrow_{\$} \mathcal{D}_s^{l\times k}$ and $\boldsymbol{D} \leftarrow_{\$} \mathcal{D}_s^{r\times k}$.
2: Compute $\boldsymbol{y} = -\boldsymbol{c}^t\boldsymbol{C} - \boldsymbol{d}^t\boldsymbol{D} + h\boldsymbol{g}^t \in R_q^k$.
3: Run $\boldsymbol{r}'_j \leftarrow \mathtt{SamplePre}(\boldsymbol{a}, \boldsymbol{R}, \boldsymbol{y}_j, s)$ for $j = 1, \ldots, k$.
4: Let $\boldsymbol{R}' = [\boldsymbol{r}'_1, \ldots, \boldsymbol{r}'_k]$.
5: Output $\boldsymbol{R}_f = [-\left(\boldsymbol{T}\boldsymbol{H}^{-1}\boldsymbol{R}\right)^t, \boldsymbol{C}^t, (\boldsymbol{H}^{-1}\boldsymbol{R})^t, \boldsymbol{D}^t]^t$.

We can compute that $\boldsymbol{f}^t\boldsymbol{R}_f = \boldsymbol{b}^t\boldsymbol{R}' + \boldsymbol{c}^t\boldsymbol{C} + \boldsymbol{d}^t\boldsymbol{D}$. Based on the output of the algorithm SamplePre, it is easy to verify that

$$\boldsymbol{f}^t\boldsymbol{R}_f = \boldsymbol{y}^t + \boldsymbol{c}^t\boldsymbol{C} + \boldsymbol{d}^t\boldsymbol{D} = h\boldsymbol{g}^t.$$

In this case, ExtRight succeed in delegating a trapdoor for $\boldsymbol{b} \in R_q^m$ to a trapdoor for $\boldsymbol{f} \in R_q^{2m+l+r}$.

5. THE MAIN CONSTRUCTION: A RING-BASED HIBS

In this part, we will describe our HIBS scheme and prove its security in the ROM. There are some parameters involved in our construction, which are defined below:

- $k, d, l, n, \eta, \lambda$ are all positive integers.
- To enable the NTT, positive prime q is chosen such that $q \equiv 1 \mod 2n$.
- Gauss parameter satisfies $s^2 \leq \eta / \left(\sqrt{kn} \cdot \omega(\log n)\right)$.
- Bound $\beta \geq (4\lambda + 2)\eta$, with dimension parameter $m > \log q / \log(2\beta)$.

5.1. Tool functions

The identity we considered here is represented by a matrix $\mathsf{id} = [\mathsf{id}_1, \ldots, \mathsf{id}_d] \in \mathbb{Z}_q^{m\times d}$ where each component id_i belongs to $\mathbb{Z}_q^m \backslash \{\boldsymbol{0}\}$. In [29], Agrawal *et al.* introduced an encoding function

$$H : \mathbb{Z}_q^m \to \mathbb{Z}_q^{m\times m},$$

which is an encoding with full-rank differences. That is, the matrix $H(\boldsymbol{x}) - H(\boldsymbol{y}) \in \mathbb{Z}_q^{m\times m}$ is invertible for all distinct $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_q^m$, and function H can be computed in polynomial time.

Let $\mathcal{M}$ be the message space and λ be a positive integer. $\mathcal{V}$ represents a subset of R, which is defined as

$$\mathcal{V} = \{\boldsymbol{v} \in R, \boldsymbol{v}_i \in \{-1, 0, 1\}, ||\boldsymbol{v}||_2 = \sqrt{\lambda}\}.$$

We define two hash functions as follows:

$$H_1 : R_q \times \mathcal{M} \to \mathcal{V},$$

and H_2 maps elements from $\mathbb{Z}_q^m$ to R_q.

5.2. Construction

For the sake of simplicity, the tag h of $\boldsymbol{g}$-trapdoor is set to be 1 in our HIBS construction. Suppose $\mathsf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_{l-1})$ and a child identity $\mathsf{id}|\mathsf{id}_l = [\mathsf{id}, \mathsf{id}_l]$. Let $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Sign}, \mathsf{Ver})$ be the new ring-based HIBS scheme, the definition of each algorithm is given below. $\mathsf{Setup}\ (1^n, d)$: take the security parameter 1^n and a parameter d as input. Do

1 . $\mathsf{GenTrap}$ generates a vector $\boldsymbol{a}_0 \in R_q^m$ and its trapdoor $\boldsymbol{R}_0 \in R_q^{m \times k}$.
2 . Sample $d+1$ uniformly random vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_d$ and $\boldsymbol{b}$ from R_q^m.
3 . Return the master key pair $(\mathsf{mpk}, \mathsf{msk})$

$$\mathsf{mpk} = \{\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_d, \boldsymbol{b}\}, \qquad \mathsf{msk} = \{\boldsymbol{R}_0\}.$$

$\mathsf{Derive}\left(\mathsf{upk}_{\mathsf{id}}, \mathsf{usk}_{\mathsf{id}}, (\mathsf{id}|\mathsf{id}_l)\right)$: inputs include the user key pair $\left(\mathsf{upk}_{\mathsf{id}}, \mathsf{usk}_{\mathsf{id}}\right)$ of an identity id at depth $l-1$. It will output a user key pair for identity $\mathsf{id}|\mathsf{id}_l$.

In fact, $\mathsf{upk}_{\mathsf{id}}$ is defined as vector

$$\boldsymbol{f}_{\mathsf{id}} = \begin{bmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_1 + \boldsymbol{b}^t H(\mathsf{id}_1) \\ \vdots \\ \boldsymbol{a}_{l-1} + \boldsymbol{b}^t H(\mathsf{id}_{l-1}) \end{bmatrix} \in R_q^{l \times m}.$$

$\mathsf{usk}_{\mathsf{id}}$ is the $\boldsymbol{g}$-trapdoor of $\boldsymbol{f}_{\mathsf{id}}$. We can calculate the public key for user $\mathsf{id}|\mathsf{id}_l$ as

$$\boldsymbol{f}_{\mathsf{id}|\mathsf{id}_l} = \begin{bmatrix} \boldsymbol{f}_{\mathsf{id}} \\ \boldsymbol{a}_l + \boldsymbol{b}^t H(\mathsf{id}_l) \end{bmatrix} \in R_q^{(l+1) \times m}$$

and construct the $\boldsymbol{g}$-trapdoor for $\boldsymbol{f}_{\mathsf{id}|id_l}$ by running

$$\boldsymbol{R}_{\mathsf{id}|\mathsf{id}_l} \leftarrow \mathsf{ExtLeft}\left(\boldsymbol{f}_{\mathsf{id}|\mathsf{id}_l}, \mathsf{usk}_{\mathsf{id}}, s\right),$$

Output $\mathsf{upk}_{\mathsf{id}|\mathsf{id}_l} \leftarrow \boldsymbol{f}_{\mathsf{id}|\mathsf{id}_l}$ and $\mathsf{usk}_{\mathsf{id}|\mathsf{id}_l} \leftarrow \boldsymbol{R}_{\mathsf{id}|\mathsf{id}_l}$.

Algorithm $\mathsf{Extract}$ behaves the same as Derive by changing the user keys into master keys and setting $\boldsymbol{f}_0 = \boldsymbol{a}_0$.

$\mathsf{Sign}\ (\mu, \mathsf{id}, \mathsf{usk}_{id})$: On input a message $\mu \in \mathcal{M}$, an identity and the user secret key, do

1. Compute $y_{\mathsf{id}} = \prod_{i=1}^{l-1} H_2(\mathsf{id}_i) \in R_q$.
2. Sample $\boldsymbol{x}_{\mathsf{id}} \in R_q^{l \times m}$ as

$$\boldsymbol{x}_{\mathsf{id}} \leftarrow \mathsf{SamplePre}\left(\boldsymbol{f}_{\mathsf{id}}, R_{\mathsf{id}}, y_{\mathsf{id}}, s\right)$$

where $\mathsf{upk}_{\mathsf{id}} = \boldsymbol{f}_{\mathsf{id}}$ and $\mathsf{usk}_{\mathsf{id}} = \boldsymbol{R}_{\mathsf{id}}$. If $||\boldsymbol{x}_{\mathsf{id}}||_\infty > \eta$, rerun Step 2.

3. Select a vector $\boldsymbol{e}$ from $R_\eta^{l \times m}$ and compute

$$z = v\boldsymbol{x}_{\mathsf{id}} + \boldsymbol{e} \mod q,$$

where $v = H_1\left(\boldsymbol{f}_{\mathsf{id}}^t \boldsymbol{e}, \mu\right) \in R_q$. Output the signature as $\sigma = (z, v)$. $\mathsf{Ver}\left(\mathsf{upk}_{id}, \mathsf{id}, \mu, \sigma\right)$: based on a user public key $\mathsf{upk}_{\mathsf{id}}$, an identity id, a signature σ and a message μ, the algorithm will decide to accept or reject.

1. Compute $y_{\mathsf{id}} = \prod_{i=1}^{l-1} H_2(\mathsf{id}_i)$ and obtain a vector by

$$\boldsymbol{w} = \boldsymbol{f}_{\mathsf{id}}^t z_{\mathsf{id}} - y_{\mathsf{id}} v.$$

2. If $\boldsymbol{v} = H_1(\boldsymbol{w}, \mu)$, output 1. Otherwise, output 0 and reject.

5.3. Correctness and security

The algorithm $\mathsf{SamplePre}$ confirms that $\boldsymbol{f}_{\mathsf{id}}^t \boldsymbol{x}_{\mathsf{id}} = y_{\mathsf{id}} \mod q$. When the scheme is behaved as specified, we know that

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{f}_{\mathsf{id}}^t z_{\mathsf{id}} - y_{\mathsf{id}} v \mod q \\ &= \boldsymbol{f}_{\mathsf{id}}^t \boldsymbol{x}_{\mathsf{id}} v + \boldsymbol{f}_{\mathsf{id}}^t \boldsymbol{e} - y_{\mathsf{id}} v \mod q \\ &= y_{\mathsf{id}} v + \boldsymbol{f}_{\mathsf{id}}^t \boldsymbol{e} - y_{\mathsf{id}} v \mod q \\ &= \boldsymbol{f}_{\mathsf{id}}^t \boldsymbol{e} \mod q. \end{aligned}$$

Hence, the equation $H_1(\boldsymbol{w}, \mu) = H_1\left(\boldsymbol{f}_{\mathsf{id}}^t \boldsymbol{e}, \mu\right) = \boldsymbol{v}$ will always hold. The tag polynomial h is set to be 1 in this case and the trapdoor $\boldsymbol{R}_{\mathsf{id}|\mathsf{id}_l}$ outputted by $\mathsf{ExtLeft}$ follows the distribution $\mathcal{D}_s^{(l+1)m \times k}$, Lemma 4.1 shows that $||\boldsymbol{x}_{\mathsf{id}}||_\infty \leq \eta$ will hold in most cases. Thus, the number of iterations of Step 2 in Sign algorithm will be small.

5.4. Security analysis

We will show that the HIBS scheme is $\mathsf{UF\text{-}sID\text{-}CMA}$ secure in the ROM, under the Ring-$\mathrm{SIS}_{n,m,q,\beta}$ assumption.

THEOREM 5.1. *The HIBS scheme we constructed is $\mathsf{UF\text{-}sID\text{-}CMA}$ secure provided that the Ring-$\mathrm{SIS}_{n,m,q,\beta}$ assumption holds.*

Proof. A sequence of games is constructed to prove the security. The $\mathsf{UF\text{-}sID\text{-}CMA}$ game defined in Section 2 is set to be Game 0. Based on the hardness of Ring-$\mathrm{SIS}_{n,m,q,\beta}$ problem, we will show that the adversary has negligible advantage to win in Game 2. Finally, we can conclude that there is no

PPT adversary can win the original UF-sID-CMA game with non-negligible probability if those games are indistinguishable. Game 0. This game behaves the same as the original UF-sID-CMA game, which is between an adversary $\mathcal{A}$ against our HIBS scheme and a UF-sID-CMA challenger $\mathcal{C}$. Game 1. The master public key $\{\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_d, \boldsymbol{b}\}$ in Game 0 is generated by the challenger, where each vector is randomly distributed in R_q^m and a $\boldsymbol{g}$-trapdoor is known for $\boldsymbol{a}_0$.

Game 0 and Game 1 proceed identically except those vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_d$ generated by the challenger. The target vector id^* that the adversary intends to attack is denoted by $\mathsf{id}^* = [\mathsf{id}_1^*, \ldots, \mathsf{id}_l^*]$. In Game 1, for $i = 1, \ldots, d$, the challenger samples matrixes $\boldsymbol{T}_i$ from $R_q^{m \times m}$ randomly and construct $\boldsymbol{a}_i$ as

$$\boldsymbol{a}_i \leftarrow \boldsymbol{a}_0^t \boldsymbol{T}_i - \boldsymbol{b}^t H\left(\mathsf{id}_i^*\right).$$

The rest of the game is the same. Since each entry in $\boldsymbol{T}$ is a random polynomial in R_q, the vector $\boldsymbol{a}_0^t \boldsymbol{T}_i$ is statistically close to unform distribution. $\boldsymbol{a}_i$ in Game 0 and Game 1 are indistinguishable and adversary has negligible probability to distinguish Game 0 and Game 1.

Game 2. We change the rest vectors in master public key: $\boldsymbol{a}_0$ and $\boldsymbol{b}$. In Game 2, we select $\boldsymbol{a}_0$ from R_q^m randomly but run algorithm GenTrap to get vector $\boldsymbol{b} \in R_q^{m \times k}$ and its $\boldsymbol{g}$-trapdoor $\boldsymbol{R} \in R_q^{m \times m}$. The construction of $\boldsymbol{a}_i$ remains unchanged.

$\mathcal{C}$ answers Extract queries by using the $\boldsymbol{g}$-trapdoor of vector $\boldsymbol{b}$. To generate a user key pair for $\mathsf{id} \in \mathbb{Z}_q^{m \times (k-1)}$, which is not a parent identity of id^*, $\mathcal{C}$ should construct a $\boldsymbol{g}$-trapdoor for vector $\boldsymbol{f}_{id}$ where

$$\boldsymbol{f}_{\mathsf{id}} = \begin{bmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_0^t \boldsymbol{T}_1 + \boldsymbol{b}^t \left(H\left(\mathsf{id}_1\right) - H\left(\mathsf{id}_1^*\right)\right) \\ \vdots \\ \boldsymbol{a}_0^t \boldsymbol{T}_i + \boldsymbol{b}^t \left(H\left(\mathsf{id}_i\right) - H\left(\mathsf{id}_i^*\right)\right) \\ \vdots \end{bmatrix}.$$

We suppose that i is the first index such that $\mathsf{id}_i \neq \mathsf{id}_i^*$. By construction, $\left(H\left(\mathsf{id}_i\right) - H\left(\mathsf{id}_i^*\right)\right)$ is an invertible matrix. In this case, the vector $\boldsymbol{f}_{\mathsf{id}}$ can be written as

$$\boldsymbol{f}_{\mathsf{id}} = \begin{bmatrix} \boldsymbol{a}_0 \\ \boldsymbol{c} \\ \boldsymbol{a}_0^t \boldsymbol{T}_i + \boldsymbol{b}^t \left(H\left(\mathsf{id}_i\right) - H\left(\mathsf{id}_i^*\right)\right) \\ \boldsymbol{d} \end{bmatrix},$$

where $\boldsymbol{c} = \begin{bmatrix} \boldsymbol{a}_0^t \boldsymbol{T}_1 \\ \vdots \\ \boldsymbol{a}_{i-1}^t \boldsymbol{T}_{i-1} \end{bmatrix}$ and $\boldsymbol{d}$ denotes the rest part of vector $\boldsymbol{f}_{\mathsf{id}}$. $\mathcal{C}$ can output the user private key by running

$$\boldsymbol{R}_{\mathsf{id}} \leftarrow \mathsf{ExtRight}\left(\boldsymbol{f}_{\mathsf{id}}, \boldsymbol{R}, s\right).$$

Set $\mathsf{upk}_{\mathsf{id}} = \boldsymbol{f}_{\mathsf{id}}$, $\mathsf{usk}_{\mathsf{id}} = \boldsymbol{R}_{\mathsf{id}}$ and output $\left(\mathsf{upk}_{\mathsf{id}}, \mathsf{usk}_{\mathsf{id}}\right)$.

Given the user secret key $\mathsf{usk}_{\mathsf{id}} = \boldsymbol{R}$, the Derive quires about the user $\mathsf{id}|\mathsf{id}_k$ can be answered easily by running the algorithm

$$\boldsymbol{R}_{\mathsf{id}|\mathsf{id}_k} \leftarrow \mathsf{ExtLeft}\left(\boldsymbol{f}_{\mathsf{id}|\mathsf{id}_k}, \boldsymbol{R}_{\mathsf{id}}, s\right).$$

Reduction from Ring-SIS. It remains to show that a PPT adversary $\mathcal{A}$ fails to output a valid but non-trivial forgery $\sigma^* = (z^*, v^*)$ of identity id^* on message μ^*.

If $\mathcal{A}$ successes with non-negligible probability, then the challenger $\mathcal{C}$ reruns the adversary $\mathcal{A}$ and takes the same random tape but different hash function H_1 as inputs. The General Forking Lemma [30] indicates that $\mathcal{A}$ will output a new forgery $\sigma' = \left(z', v'\right)$ of id^* on the same message μ^* with non-negligible probability and $v^* \neq v'$. Then, we come to know that

$$\boldsymbol{f}_{\mathsf{id}^*}^t z^* - y_{\mathsf{id}^*} v^* = \boldsymbol{f}_{\mathsf{id}^*}^t z' - y_{\mathsf{id}^*} v'.$$

Taking the equation $y_{\mathsf{id}^*} = \boldsymbol{f}_{\mathsf{id}^*}^t \boldsymbol{x}_{\mathsf{id}^*}$ into consideration, we have

$$\begin{aligned} &\boldsymbol{f}_{\mathsf{id}^*}^t z^* - y_{\mathsf{id}^*} v^* - \boldsymbol{f}_{\mathsf{id}^*}^t z' + y_{\mathsf{id}^*} v' \\ &= \boldsymbol{f}_{\mathsf{id}^*} \left(z^* - z' - \boldsymbol{x}_{\mathsf{id}^*} v^* + \boldsymbol{x}_{\mathsf{id}^*} v'\right) \\ &= 0. \end{aligned}$$

With $||\boldsymbol{x}_{\mathsf{id}^*}||_\infty \leq \eta$, by designing, we have $||\boldsymbol{x}_{\mathsf{id}^*} v^*||_\infty$, $||\boldsymbol{x}_{\mathsf{id}^*} v'||_\infty \leq \lambda\eta$ and $||z^*||_\infty, ||z'||_\infty \leq (\lambda + 1)\eta$. The solution $z^* - z' - \boldsymbol{x}_{\mathsf{id}^*} v^* + \boldsymbol{x}_{\mathsf{id}^*} v'$ is bounded by

$$||z^* - z' - \boldsymbol{x}_{\mathsf{id}^*} v^* + \boldsymbol{x}_{\mathsf{id}^*} v'||_\infty \leq (4\lambda + 2)\eta \leq \beta.$$

In this case, the Ring-SIS$_{n,m,q,\beta}$ problem can be solved as long as

$$z^* - z' - \boldsymbol{x}'_{\mathsf{id}^*} \left(v^* - v'\right) \neq 0$$

with non-negligible probability. Based on the preimage min-entropy property [12], one can also get another vector $\boldsymbol{x}'_{\mathsf{id}^*}$ with probability greater than $1 - 2^{\omega(\log m)}$. Moreover, vectors $\boldsymbol{x}_{\mathsf{id}^*}$ and $\boldsymbol{x}'_{\mathsf{id}^*}$ will be identity expect the i-th entry and

$$\boldsymbol{f}_{\mathsf{id}^*}^t \boldsymbol{x}_{\mathsf{id}^*} = \boldsymbol{f}_{\mathsf{id}^*}^t \boldsymbol{x}'_{\mathsf{id}^*} = y_{\mathsf{id}^*}.$$

If

$$z^* - z' - \boldsymbol{x}_{\mathsf{id}^*} \left(v^* - v'\right) = 0,$$

then we can conclude that

$$z^* - z' - \boldsymbol{x}'_{\mathsf{id}^*} \left(v^* - v'\right) \neq 0.$$

As the vectors $\boldsymbol{x}_{\mathsf{id}^*}$ and $\boldsymbol{x}'_{\mathsf{id}^*}$ play the same role in our scheme and the adversary $\mathcal{A}$ has no ideal about which vector is used in this simulation. Hence,

$$z^* - z' - \boldsymbol{x}'_{\mathsf{id}^*} v^* + \boldsymbol{x}'_{\mathsf{id}^*} v' \neq 0$$

will hold with probability at least 1/2. Hence, with non-negligible probability the Ring-SIS$_{n,m,q,\beta}$ problem can be solved in polynomial time, which contradicts the assumption. ■

6. CONCLUSION

In this paper, we proposed two new trapdoor delegation algorithms in the ring setting and then constructed the first ring-based HIBS scheme. We also proved its UF-sID-CMA security in ROM assuming the hardness of the Ring-SIS$_{n,m,q,\beta}$ problem. Compared with the former integer-based scheme, the new one is more efficient. Because the operations in polynomial ring is much faster.

Funding

This work is supported by the National Natural Science Foundation of China (Nos. 61702541, 11531002, 61872087, 61822202, 61872089), the Young Elite Scientists Sponsorship Program by China Association for Science and Technology (No. YESS20170128), the Science and Technology Research Plan Program by National University of Defence Technology (No. ZK17-03-46). .

REFERENCES

[1] Shamir, A. (1984) Identity-Based Cryptosystems and Signature Schemes. In *Proc. CRYPTO 84*, Santa Barbara, CA, USA, 19–22 August, pp. 47–53. Springer, Berlin.

[2] Boneh, D. and Franklin, M. (2001) Identity-Based Encryption from the Weil Pairing. In *Proc. CRYPTO 01*, Santa Barbara, CA, USA, 19–23 August, pp. 213–229. Springer, Berlin.

[3] Hess, F. (2002) Efficient Identity Based Signature Schemes Based on Pairings. In *Proc. SAC 02*, St. John's, NF, Canada, 15–16 August, pp. 310–324. Springer, Berlin.

[4] Choon, J.C. and Cheon, J.H. (2003) An Identity-Based Signature from Gap Diffie-Hellman Groups. In *Proc. PKC 03*, Miami, FL, USA, 6–8 January, pp. 18–30. Springer, Berlin.

[5] Barreto, P.S., Libert, B., McCullagh, N. and Quisquater, J.-J. (2005) Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps. In *Proc. ASIACRYPT 05*, Chennai, India, 4–8 December, pp. 515–532. Springer, Berlin.

[6] Paterson, K.G. and Schuldt, J.C. (2006) Efficient Identity-Based Signatures Secure in the Standard Model. In *Proc. ACISP 06*, Melbourne, VIC, Australia, 3–5 July, pp. 207–222. Springer, Berlin.

[7] Gentry, C. and Silverberg, A. (2002) Hierarchical ID-Based Cryptography. In *Proc. ASIACRYPT 02*, Queenstown, New Zealand, 1–5 December, pp. 548–566. Springer, Berlin.

[8] Guo, F., Chen, R., Susilo, W., Lai, J., Yang, G. and Mu, Y. (2017) Optimal Security Reductions for Unique Signatures: Bypassing Impossibilities with a Counterexample. In *Proc. CRYPTO 17*, Santa Barbara, USA, 20–24 August, pp. 517–547. Springer, Berlin.

[9] Shor, P.W. (1999) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41, 303–332.

[10] Buchmann, J. and Ding, J. (2008) Post-Quantum Cryptography: Second International Workshop. In *PQCrypto 2008*, Cincinnati, OH, USA, 17–19 October. Springer Science & Business Media, Berlin.

[11] Ajtai, M. (1996) Generating Hard Instances of Lattice Problems. In *Proc. CSCW 96*, Boston, Massachusetts, USA, November, pp. 99–108. ACM, New York.

[12] Gentry, C., Peikert, C. and Vaikuntanathan, V. (2008) Trapdoors for Hard Lattices and New Cryptographic Constructions. In *Proc. STOC 08*, Victoria, British Columbia, Canada, May, pp. 197–206. ACM, New York.

[13] Lyubashevsky, V. (2009) Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *Proc. ASIACRYPT 09* Tokyo, Japan, 6–10 December, pp. 598–616. Springer, Berlin.

[14] Cash, D., Hofheinz, D., Kiltz, E. and Peikert, C. (2010) Bonsai Trees, or How to Delegate a Lattice Basis. In *Proc. EUROCRYPT 10*, Monaco and Nice, France, 30 May–3 June, pp. 523–552. Springer, Berlin.

[15] Lyubashevsky, V. and Micciancio, D. (2008) Asymptotically Efficient Lattice-Based Digital Signatures. In *Proc. TCC 08*, New York, NY, USA, 19–21 March, pp. 37–54. Springer, Berlin.

[16] Rückert, M. (2010) Strongly Unforgeable Signatures and Hierarchical Identity-Based Signatures from Lattices Without Random Oracles. In *Proc. PQCRYPTO 10*, Darmstadt, Germany, 25–28 May, pp. 182–200. Springer, Berlin.

[17] Liu, Z., Hu, Y., Zhang, X. and Li, F. (2013) Efficient and strongly unforgeable identity-based signature scheme from lattices in the standard model. *Secur. Commun. Netw.*, 6, 69–77.

[18] Tian, M., Huang, L. and Yang, W. (2012) A new hierarchical identity-based signature scheme from lattices in the standard model. *IJ Netw. Secur.*, 14, 310–315.

[19] Tian, M., Huang, L. and Yang, W. (2013) Efficient hierarchical identity-based signatures from lattices. *Int. J. Electr. Secur. Digital Forensics*, 5, 1–10.

[20] Micciancio, D. and Peikert, C. (2012) Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *Proc. EUROCRYPT 12*, Cambridge, United Kingdom, 15–19 April, pp. 700–718. Springer, Berlin.

[21] Lai, R.W., Cheung, H.K. and Chow, S.S. (2014) Trapdoors for Ideal Lattices with Applications. In *Proc. INSCRYPT 14*, Beijing, China, 13–15 December, pp. 239, Springer–256, Berlin.

[22] Banaszczyk, W. (1993) New bounds in some transference theorems in the geometry of numbers. *Math. Ann.*, 296, 625–635.

[23] Micciancio, D. (2002) Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions from Worst-Case Complexity Assumptions. In *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, Vancouver, BC, Canada, 19–19 November, pp. 356–365. IEEE, New York.

[24] Lyubashevsky, V. and Micciancio, D. (2006) Generalized Compact Knapsacks are Collision Resistant. In *Proc. ICALP 06*, Venice, Italy, 10–14 July, pp. 144–155. Springer, Berlin.

[25] Peikert, C. and Rosen, A. (2006) Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In

Proc. TCC 06, New York, NY, USA, 4–7 March, pp. 145–166. Springer, Berlin.

[26] Lyubashevsky, V., Micciancio, D., Peikert, C. and Rosen, A. (2008) SWIFFT: A Modest Proposal for FFT Hashing. In *Proc. FSE 08*, Lausanne, Switzerland, 10–13 February, pp. 54–72. Springer, Berlin.

[27] Ling, S., Nguyen, K. and Wang, H. (2015) Group Signatures from Lattices: Simpler, Tighter, Shorter, Ring-Based. In *Proc. PKC 15*, Gaithersburg, MD, USA, 30 March–1 April, pp. 427–449. Springer, Berlin.

[28] Lyubashevsky, V., Peikert, C. and Regev, O. (2013) A Toolkit for Ring-LWE Cryptography. In *Proc. EUROCRYPT 13*, Athens, Greece, 26–30 May, pp. 35–54. Springer, Berlin.

[29] Agrawal, S., Boneh, D. and Boyen, X. (2010) Efficient lattice (H) IBE in the standard model. In *Proc. EUROCRYPT 10*, Monaco and Nice, France, 30 May–3 June, pp. 553–572. Springer, Berlin.

[30] Bellare, M. and Neven, G. (2006) Multi-Signatures in the Plain Public-Key Model and A General Forking Lemma. In *Proc. CCS 06*, Alexandria Virginia, USA, October, pp. 390–399. ACM, New York.