Trapdoors for Ideal Lattices with Applications

Russell W.F. Lai[1(✉)], Henry K.F. Cheung[2], and Sherman S.M. Chow[1]

[1] Department of Information Engineering, The Chinese University of Hong Kong, Sha Tin, New Territories, Hong Kong
{wflai,sherman}@ie.cuhk.edu.hk
[2] Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Sha Tin, New Territories, Hong Kong
kfcheung@se.cuhk.edu.hk

Abstract. There is a lack of more complicated ideal-lattice-based cryptosystems which require the use of lattice trapdoors, for the reason that currently known trapdoors are either only applicable to general lattices or not well-studied in the ring setting. To facilitate the development of such cryptosystems, we extend the notion of lattice trapdoors of Micciancio and Peikert (Eurocrypt '12) into the ring setting with careful justification. As a demonstration, we use the new trapdoor to construct a new hierarchical identity-based encryption scheme, which allows us to construct public-key encryption with chosen-ciphertext security, signatures, and public-key searchable encryption.

Keywords: Ideal lattices · Trapdoors · Identity-based encryption

1 Introduction

Lattice-based cryptography is a promising alternative to create cryptosystems that are secure even against quantum adversaries. Many powerful primitives including fully-homomorphic encryption [1–4], homomorphic signatures [5,6], multilinear map [7], (hierarchical) identity-based encryption [8,9] (which is also useful for achieving other cryptographic goals like public-key encryption with chosen-ciphertext security, signatures, and public-key searchable encryption), and much more can be realized by lattices. Security reductions of some of these constructions are directly based on the now well-studied (ring-)LWE (learning with errors) or (ring-)SIS (short integer solutions) problems, which are both as hard as the corresponding worst-case (ideal) lattice problems.

Hard Lattice Problems. An instance of the LWE problem is defined by a random n by m integer matrix A and a vector $\boldsymbol{b}$, where $\boldsymbol{b} = A^T\boldsymbol{s} + \boldsymbol{e} \bmod q$ for some

This work is supported by grants 439713, 14201914 from Research Grants Council (RGC), and grants 4055018, 4930034 from The Chinese University of Hong Kong. Sherman Chow is supported by the Early Career Award from RGC. Part of the work was done while the second author is with Department of Information Engineering.

D. Lin et al. (Eds.): Inscrypt 2014, LNCS 8957, pp. 239–256, 2015.
DOI: 10.1007/978-3-319-16745-9_14

secret vector $\boldsymbol{s}$ and small noise vector $\boldsymbol{e}$. The problem is to find the vector $\boldsymbol{s}$. As a "dual" problem to LWE, an instance of the SIS problem is defined by the same random matrix A, where one is asked to find a short vector $\boldsymbol{x}$ so that $A\boldsymbol{x} = 0 \bmod q$.

The "ring" versions of LWE and SIS, named ring-LWE and ring-SIS respectively, are specific instances of LWE and SIS respectively defined for some structured matrix A to be explained below.

Ideal Lattices. In ideal lattices, or the so called "ring setting", the matrix A above is required to have some additional algebraic structures. One commonly used example is to interpret each column of A as coefficients of a degree-$(n-1)$ polynomial $p(x)$, and require that $xp(x) \bmod (x^n + 1)$ is also contained in some column of A. In such case, the matrix multiplications by A are equivalent to polynomial multiplications. We can therefore view each vector $\boldsymbol{v}$ as an element $\mathbf{v}$ in the ring $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$, and each n-by-n sub-matrix A_i in A a ring element $\mathbf{a}_i$ in R_q. As the (ring-)LWE and (ring-)SIS problems have such simple forms, the operations performed in the corresponding cryptosystems are rather efficient.

Due to the algebraic structure of ideal lattices, cryptosystems based on ideal lattices (with security based on the ring-LWE or ring-SIS assumptions) are more efficient than their counterparts in general lattices: (1) The size of some parameters, which are originally matrices, is reduced by a factor of n, as each n-by-n sub-matrix is now represented as a ring element; (2) The multiplications of ring elements in R_q can be implemented on hardware by a variant of Fourier transform [3].

Lattice Trapdoors. For more complicated primitives, a "trapdoor" is generated together with a random lattice so that, while it is still hard for the adversary to solve the (ring-)LWE or (ring-)SIS problems, the problems become easily solvable with the help of the trapdoor.

Initiated by the work of Gentry *et al.* [10], a (old-type) trapdoor [10,11] of (the lattice defined by) a matrix A is a short basis of the lattice $\Lambda_q^{\perp}(A)$, which contains all the vectors $\boldsymbol{x}$ such that $A\boldsymbol{x} = \mathbf{0} \bmod q$. Using the trapdoor, one can sample short vectors $\boldsymbol{x}$ so that $A\boldsymbol{x} = \boldsymbol{u} \bmod q$ for any target vector $\boldsymbol{u}$. Moreover, the owner of the trapdoor of A can "delegate" the trapdoor of an extended matrix (A, B) for any matrix B.

Micciancio and Peikert [12] developed a new type of trapdoors for general lattices which is simpler and more efficient to use when compared to the old trapdoors. A new-type trapdoor of A is a matrix T with small norm so that $A\begin{bmatrix} T \\ I \end{bmatrix} = HG$ for some invertible matrix H, which is referred as the tag of the trapdoor, and a nicely structured matrix G called the primitive matrix, where the inversions of SIS (also known as "Gaussian sampling") and LWE involving G are easy and efficient.[1] At the high-level sense, T can be considered as a secret

[1] We switch the notation from the original R in [12] to T to avoid clashing of notations in the later sections.

transformation from A to G which reduces the originally difficult inversions of SIS and LWE involving A to the much easier inversions involving G. Notice that the new-type trapdoors have the additional ability to invert LWE, which is not the case for the old-type trapdoors. In addition, the size of the new-type trapdoors is much (at least 4 times) smaller than that of the old-type.

However, despite the increase of efficiency, there is a lack of cryptosystems in ideal lattices that require the use of trapdoors. One possible reason for this is that, the trapdoors introduced by Gentry *et al.* [10] and improved by Alwen and Peikert [11] are based on general lattices. Stehlé [13] attempted to extend the trapdoor algorithms to ideal lattices, but the result is based on a non-standard ideal-LWE assumption which, unlike the ring-LWE assumption, does not have search-to-decision reduction. Later, Micciancio and Peikert [12] introduced a new notion of lattice trapdoors which have even greater functionality, namely, to invert not only SIS but also LWE. More importantly, the new trapdoors can be translated to the ring setting, as mentioned in [12] but unfortunately without much details.

Our Contributions. In this work, we extend the trapdoors from Micciancio and Peikert [12] to the ring setting. As a result, the sizes of the "primitive vectors", the public vectors and trapdoors are reduced by a factor of n. As in other recent cryptosystems [2,3,14] that are based on ring-LWE, we work with the "preferred" choice of ring $R := \mathbb{Z}[x]/\langle x^n + 1\rangle$ where n is a power of 2. For such choice of ring R, the general strategy of transforming the trapdoors to the ring setting is to interpret each n by n submatrix in the construction of [12] as a ring element in R. By breaking down elements in R in terms of the "power basis" $1, x, x^2, \ldots, x^{n-1}$, we show that some of the algorithms in [12] can be reused. We also justify the correctness of such transformation carefully by replacing certain theorems and lemmas by those proven in the ring setting.

Finally, we demonstrate the power of the new trapdoors by constructing a new identity-based encryption (IBE) scheme which improves the IBE scheme constructed by Agrawal *et al.* [8] in three aspects, namely, being ideal-lattice-based, having reduced trapdoor size, and being secure against chosen-ciphertext attack.

2 Preliminary

Notations. Let λ be the security parameter. Let $f(x) = f_\lambda(x) \in \mathbb{Z}[x]$ be a polynomial of degree $n = n(\lambda)$. Let $q = q(\lambda) \in \mathbb{Z}$ be a prime integer, $p = p(\lambda) \in \mathbb{Z}_q^*$ be relatively prime to q. Let $R := \mathbb{Z}[x]/\langle f(x)\rangle$ and $R_q := R/qR$. Let χ be a distribution over the ring R. "|" denotes row concatenation of vectors or matrices. If S is a set, then $x \leftarrow S$ denotes the sampling of a uniformly random element x from S. If X is a distribution, then $x \leftarrow X$ denotes the sampling of a random element x according to the distribution X. If $\mathcal{A}$ is an algorithm, then $x \leftarrow \mathcal{A}$ means that x is the output of the algorithm $\mathcal{A}$. To distinguish between elements, vectors and matrices of $\mathbb{Z}$ and R, we follow the notations listed in

Table 1. Notations of elements, vectors and matrices of $\mathbb{Z}$ and R

	Element	Vector	Matrix
Integers $\mathbb{Z}$	a	$\boldsymbol{a}$	A
Ring R	$\mathbf{a}$	$\mathbf{a}$	$\mathbf{A}$

Table 1. We denote the k-by-k identity matrix over R by $\mathbf{I}_k$ and the k-by-l zero matrix over R by $\mathbf{0}_{k\times l}$. Without further specifications, $\|\boldsymbol{x}\|$ denotes the L_2 norm of the vector $\boldsymbol{x}$ and is extended naturally to $\|\mathbf{x}\|$ via the coefficient embedding.

2.1 Lattice Background

Statistical Distance. Let X and Y be two random variables taking values in some finite set Ω. The statistical distance $\Delta(X;Y)$ is defined as

$$\Delta(X;Y) := \frac{1}{2}\sum_{s\in\Omega} |\Pr[X=s] - \Pr[Y=s]|.$$

We say that the ensembles of random variables $X(\lambda)$ and $Y(\lambda)$ are statistically close if $\Delta(X;Y)$ is a negligible function in λ.

Integer Lattices. We consider three types of integer lattices. For an integer modulus q, $A \in \mathbb{Z}_q^{n\times m}$ and $u \in \mathbb{Z}_q^n$, define:

$$\begin{aligned}
\Lambda_q(A^T) &:= \{\boldsymbol{x} \in \mathbb{Z}^m : \exists\ \boldsymbol{s} \in \mathbb{Z}_q^n\ s.t.\ A^T\boldsymbol{s} = \boldsymbol{x} \bmod q\}\\
\Lambda_q^\perp(A) &:= \{\boldsymbol{x} \in \mathbb{Z}^m : A\boldsymbol{x} = \mathbf{0} \bmod q\}\\
\Lambda_q^{\boldsymbol{u}}(A) &:= \{\boldsymbol{x} \in \mathbb{Z}^m : A\boldsymbol{x} = \boldsymbol{u} \bmod q\}
\end{aligned}$$

Note that for any $t \in \Lambda_q^{\boldsymbol{u}}(A)$, $\Lambda_q^{\boldsymbol{u}}(A) = \Lambda_q^\perp(A) + \boldsymbol{t}$ is a shift of $\Lambda_q^\perp(A)$.

Ideal Lattices. Correspondingly, we consider three types of ideal lattices. For an integer modulus q, $\mathbf{a} \in R_q^k$ and $\mathbf{u} \in R_q$, define:

$$\begin{aligned}
\Lambda_q(\mathbf{a}) &:= \{\mathbf{x} \in R^k : \exists\ \mathbf{s} \in R_q\ s.t.\ \mathbf{as} = \mathbf{x} \bmod q\}\\
\Lambda_q^\perp(\mathbf{a}^T) &:= \{\mathbf{x} \in R^k : \mathbf{a}^T\mathbf{x} = \mathbf{0} \bmod q\}\\
\Lambda_q^{\mathbf{u}}(\mathbf{a}^T) &:= \{\mathbf{x} \in R^k : \mathbf{a}^T\mathbf{x} = \mathbf{u} \bmod q\}
\end{aligned}$$

Note that for any $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{a}^T)$, $\Lambda_q^{\mathbf{u}}(\mathbf{a}^T) = \Lambda_q^\perp(\mathbf{a}^T) + \mathbf{t}$ is a shift of $\Lambda_q^\perp(\mathbf{a}^T)$.

Discrete Gaussian. Let $L \subset \mathbb{Z}^n$, $c \in \mathbb{R}^n$, $\sigma \in \mathbb{R}^+$. Define:

$$\rho_{\sigma,c}(x) = \exp(-\pi\frac{\|x-c\|^2}{\sigma^2}) \text{ and } \rho_{\sigma,c}(L) = \sum_{x\in L}\rho_{\sigma,c}(x).$$

The discrete Gaussian distribution over L with center c and parameter σ is defined as

$$\forall x \in L, \mathcal{D}_{L,\sigma,c}(x) = \frac{\rho_{\sigma,c}(x)}{\rho_{\sigma,c}(L)}.$$

For $c = 0$, denote $\rho_{\sigma,0}$ as ρ_σ and $\mathcal{D}_{L,\sigma,0}$ as $\mathcal{D}_{L,\sigma}$.

Gram-Schmidt Norm. Let $T = \{t_1, \ldots, t_k\} \subset \mathbb{R}^m$ be a set of real vectors, and $\|T\|$ denotes the L_2-norm of the longest vector in T, *i.e.*, $\|T\| := \max_{j=1}^{k} \|t_j\|$, $\widetilde{T} := \{\widetilde{t_1}, \ldots, \widetilde{t_k}\}$ denotes the Gram-Schmidt orthogonalization of the vectors $\widetilde{t_1}, \ldots, \widetilde{t_k}$ taken in that order. $\|\widetilde{T}\|$ is called the Gram-Schmidt norm of T.

2.2 Assumptions

The learning with errors (LWE) problem defined by Regev [15] is now a well-studied hard problem that is as hard as some worst-case lattice hard problems such as the shortest vector problem (SVP), via either quantum or classical reductions [15,16]. An LWE instance is defined by a matrix $A \in \mathbb{Z}_q^{n \times m}$ and a vector $\boldsymbol{b} \in \mathbb{Z}_q^m$. The search version of LWE is to find a secret vector $\boldsymbol{s} \in \mathbb{Z}^n$ so that $A^T\boldsymbol{s} + \boldsymbol{e} = \boldsymbol{b}$ for some short error vector $\boldsymbol{e} \in \mathbb{Z}^m$. The decision version is to decide whether such a pair of $(A, \boldsymbol{b})$ comes from the uniform distribution or the LWE distribution, *i.e.* $A^T\boldsymbol{s} + \boldsymbol{e} = \boldsymbol{b}$ for some short error vector $\boldsymbol{e} \in \mathbb{Z}^m$.

To define LWE in the ring setting, namely ring-LWE, A is restricted to have a certain algebraic structure. We interpret the entries in a column of A as the coefficients of a degree-$(n-1)$ polynomial $p(x)$, and require that the vector given by the coefficients of $xp(x)$ mod $f(x)$, for some degree-n polynomial $f(x)$, is also contained in some column of A. Assuming that $m = nk$ for some integer k, we can then interpret the i-th n-by-n sub-matrix of A as a ring element $\mathbf{a}_i$ in R_q, the vector $\boldsymbol{s}$ as $\mathbf{s}$ in R, $\boldsymbol{e}$ as $(\mathbf{e}_1, \ldots, \mathbf{e}_k)^T$ in R^k and $\boldsymbol{b}$ as $(\mathbf{b}_1, \ldots, \mathbf{b}_k)^T$ in R_q^k. Multiplications between a sub-matrix of A and the vector $\boldsymbol{s}$ correspond to the multiplications of the ring elements $\mathbf{a}_i$ and $\mathbf{s}$. Apparently, the search version of ring-LWE is then to find $\mathbf{s}$ given $\{\mathbf{a}_i, \mathbf{b}_i = \mathbf{a}_i\mathbf{s} + \mathbf{e}_i\}_{i=1}^k$. The decision version of ring-LWE is to distinguish the distribution of the given samples from the uniform distribution. For the detailed definitions and reductions related to ring-LWE, we refer to the comprehensive work of Lyubashevsky *et al.* [3,14].

In this work, we further restrict the polynomial $f(x)$ such that $f(x) = x^n + 1$, which is the preferred choice in recent ring-LWE-based cryptosystems [2,3,14] due to its simplicity. In this case, the ring-LWE assumption has a much simpler form. This special case is named as polynomial LWE (PLWE) by Brakerski and Vaikuntanathan [2].

Definition 1 *(PLWE assumption [2]). For all $\lambda \in \mathbb{N}$, $l = \mathsf{poly}(\lambda)$, $\mathbf{a}_i, \mathbf{u} \leftarrow R_q$, $\mathbf{e}_i, \mathbf{s} \leftarrow \chi$, the $\mathsf{PLWE}_{f,q,\chi}^{(l)}$ assumption states that the distribution of $\{(\mathbf{a}_i, \mathbf{a}_i\mathbf{s} + p\mathbf{e}_i)\}_{i=1}^l$ is computationally indistinguishable from the distribution of $\{(\mathbf{a}_i, \mathbf{u}_i)\}_{i=1}^l$.*

Theorem 1 *[2, Theorem 1]. Let λ be the security parameter. Let $k \in \mathbb{N}$ and let $m = 2^{\lfloor \log \lambda \rfloor}$ be a power of two. Let $\Phi_m(x) = x^n + 1$ be the m-th cyclotomic polynomial of degree $n = \varphi(m) = m/2$. Let $\sigma \geq \omega(\sqrt{\log n})$ be a real number, and let $q \equiv 1 \pmod{m}$ be a prime integer. Let $R = \mathbb{Z}[x] = \langle \Phi_m(x) \rangle$. Then there is a randomized reduction from $(n^2 q/r) \cdot (n(l+1)/\log(n(l+1)))^{1/4}$-approximate R-SVP to $\mathsf{PLWE}^{(l)}_{\Phi_m,q,\chi}$ where $\chi = \mathcal{D}_{\mathbb{Z}^n,\sigma}$ is the discrete Gaussian distribution. The reduction runs in time $\mathsf{poly}(n, q, l)$.*

3 Primitive Vectors in Ideal Lattices

Although Micciancio and Peikert [12] mentioned that their trapdoors can be extended to ideal lattices, they did not explain it in details. In this section, we first extend their notion of primitive matrices for general lattices to the notion of primitive vectors (of ring elements) for ideal lattices. We will then show in Sect. 4 how to use the primitive vectors to generate trapdoors for ideal lattices. The general strategy used in these sections is to interpret each n-by-n submatrices in the notion of trapdoors for general lattices as ring elements in R_q.

3.1 Construction of Primitive Vectors

Recall from [12] that a matrix $G \in \mathbb{Z}_q^{n \times m}$ is primitive if its columns generate all of $\mathbb{Z}_q^n$, *i.e.*, $G \cdot \mathbb{Z}^m = \mathbb{Z}_q^n$. For some nicely structured primitive matrices, LWE inversion and Gaussian sampling can be done efficiently. Given such a primitive matrix, the crux of the trapdoor generation algorithm is to perform a random transform on the primitive matrix.

As mentioned in [12, Sect. 4.3], the primitive vector $\mathbf{g} = (1, 2, \ldots, 2^{k-1})^T$ in R_q^k can be used in the ring setting to replace the previous primitive matrix G by interpreting the values in the ring R_q instead of $\mathbb{Z}_q$. Furthermore, the inversion and Gaussian sampling algorithms can be obtained in the ring setting as well.

Intuitively, to obtain a primitive vector in the ring setting, we need to find a primitive matrix (in the general lattices setting) in which each n-by-n submatrix is rotational, *i.e.*, a column is obtained by shifting the previous column by one entry and adding a negative sign to the first entry. One way is to permute the columns of the previous primitive matrix G to obtain such a structure. An example of G is as follows [12]:

$$G := \begin{bmatrix} \ldots \boldsymbol{g}^T \ldots & & & \\ & \ldots \boldsymbol{g}^T \ldots & & \\ & & \ddots & \\ & & & \ldots \boldsymbol{g}^T \ldots \end{bmatrix}, \boldsymbol{g}^T = \begin{bmatrix} 1 & 2 & 4 & \ldots & 2^{k-1} \end{bmatrix} \in \mathbb{Z}_q^{1 \times k}$$

We permute columns of G so that identical terms forms n-by-n diagonal block matrices. As a result, we obtain:

$$G' := [I_n | 2I_n | \ldots | 2^{k-1} I_n].$$

Since G' is obtained by permutation of columns of G, G' is still primitive. By the same permutation on the basis S of $\Lambda_q^\perp(G)$, we obtain the basis S' of $\Lambda_q^\perp(G')$, where

$$S' := \begin{bmatrix} 2I_n & & & q_0 I_n \\ -I_n & 2I_n & & q_1 I_n \\ & -I_n & & q_2 I_n \\ & & \ddots & \\ & & 2I_n & q_{k-2} I_n \\ & & -I_n & q_{k-1} I_n \end{bmatrix} \in \mathbb{Z}^{nk \times nk}.$$

The matrices G' and S' correspond to the collections of vectors of ring elements $\mathbf{g} = (1, 2, \ldots, 2^{k-1})^T \in R_q^k$ and $(\mathbf{s}_1, \ldots, \mathbf{s}_k) \in R_q^{k \times k}$ respectively, where $\mathbf{s}_i = (0, \ldots, 0, 2, -1, 0, \ldots, 0)^T$ for $i < k$, and $\mathbf{s}_k = (q_0, q_1, \ldots, q_{k-1})^T$, where $q = \sum_{i=0}^{k-1} 2^i q_i$ and $q_i \in \{0, 1\}$.

Theorem 2 summarizes the result of primitive vectors in the ring setting, with explanation deferred to the later subsections.

Theorem 2. *For any $q = \sum_{i=0}^{k-1} 2^i q_i < 2^k$ where $q_i \in \{0, 1\}$ and $k \geq 1$, there exists $\mathbf{g} = (1, 2, \ldots, 2^{k-1})^T \in R_q^k$ and $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_k) \in R_q^{k \times k}$ (thus $\mathbf{g}^T \mathbf{S} = \mathbf{0}_{1 \times k} \in R_q^k$), such that:*

- *We have $\|\tilde{\mathbf{s}_\mathbf{i}}\| < \sqrt{5}$ in the coefficient embedding.*
- *The storage requirement of $\mathbf{g}$ and $\mathbf{S}$ are further reduced by a factor of n compared to their counterparts in general lattices.*
- *Inverting $\alpha_\mathbf{g}(\mathbf{z}, \mathbf{e}) := \mathbf{g}\mathbf{z} + \mathbf{e} \bmod q$ can be performed in quasilinear $O(n \cdot \log^c n)$ time for any $\mathbf{z} \in R$ and any $\mathbf{e} \in q \cdot \mathbf{B}^{-T} \cdot [-\frac{1}{2}, \frac{1}{2})^{nk}$, where $\mathbf{B}$ can denote either $\mathbf{S}$ or $\tilde{\mathbf{S}}$. Moreover, the algorithm is perfectly parallelizable, running in polylogarithmic $O(\log^c n)$ time using n processors.*
- *Preimage sampling for $\beta_\mathbf{g}(\mathbf{x}) = \mathbf{g}^T \mathbf{x} \bmod q$ with Gaussian parameter $\sigma \geq \|\tilde{\mathbf{S}}\| \cdot w(\sqrt{\log n})$ can be performed in quasilinear $O(n \cdot \log^c n)$ time, or parallel polylogarithmic $O(\log^c n)$ time using n processors.*

3.2 Inversion for Primitive Vectors

Given a PLWE instance $\mathbf{b} = \mathbf{g}\mathbf{z} + \mathbf{e}$, which is equivalent to $\mathbf{b}_i = 2^i \mathbf{z} + \mathbf{e}_i$ for $i = 0, \ldots, k-1$, we can expand $\mathbf{b}_i$, $\mathbf{z}$ and $\mathbf{e}_i$ in terms of the power basis $1, x, x^2, \ldots, x^{n-1}$ so that the problem is equivalent to solving $b_{ij} = 2^i z_j + e_{ij}$ independently for $j = 0, \ldots, n-1$, where $\mathbf{b}_i = \sum_{j=0}^{n-1} b_{ij} x^j$, $\mathbf{z} = \sum_{j=0}^{n-1} z_j x^j$ and $\mathbf{e}_i = \sum_{j=0}^{n-1} e_{ij} x^j$. Recombining the terms according to i, the problem becomes solving $\boldsymbol{b}_j = \boldsymbol{g} z + \boldsymbol{e}_j$ where $\boldsymbol{g} = (1, 2, \ldots, 2^{k-1})^T \in \mathbb{Z}_q^k$. Let $S = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k)$ be a basis of $\Lambda_q^\perp(\boldsymbol{g}^T)$. Then $V = qS^{-T} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k)$ is a basis of $\Lambda_q(\boldsymbol{g})$. We can then use Babai's nearest plane algorithm to recover $z \in \mathbb{Z}_q$ from $\boldsymbol{b} = \boldsymbol{g} z + \boldsymbol{e}$:

Algorithm 3. *[17]* **Babai's Nearest Plane Algorithm**
Input: $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ *a basis of* $\Lambda_q(\boldsymbol{g})$, $\boldsymbol{b}$.
Output: z *and* $\boldsymbol{e}$.

1. *Compute Gram-Schmidt basis* $\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*$.
2. *For* $j = k \rightarrow 1$*:*
   (a) *Compute* $l_j = < \boldsymbol{b}_j, \boldsymbol{v}_j^* > / < \boldsymbol{v}_j^*, \boldsymbol{v}_j^* >$.
   (b) *Set* $\boldsymbol{b}_{j-1} = \boldsymbol{b}_j - (l_j - \lfloor l_j \rceil)\boldsymbol{v}_j^* - \lfloor l_j \rceil \boldsymbol{v}_j$.
3. *Return* $z = \sum_{j=1}^{k} \lfloor l_j \rceil c_j \bmod q$, $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{g}z$, *where* $\boldsymbol{v}_j = c_j \boldsymbol{g} \bmod q$.

3.3 Gaussian Sampling for Primitive Vectors

We first recall that the goal of Gaussian sampling in [12] is to sample a vector from $\Lambda_q^{\boldsymbol{u}}(G)$. This can be done by repeating n times of the sampling from $\Lambda_q^{u_j}(\boldsymbol{g}^T)$ for a desired syndrome $u_j \in \mathbb{Z}_q$, where $j = 0, \ldots, n-1$.

For the later task, there are two extreme approaches and one hybrid approach. In the one extreme, we first pre-compute a large set of samples from $\mathcal{D}_{\mathbb{Z}^k,\sigma}$ and bucket them according to the different values of u. The sampling algorithm simply draws one sample from the appropriate bucket. This approach requires large storage so that each bucket can be filled with sufficient number of samples. The other extreme exploits the fact that if q is a power of 2, then we have the orthogonalized basis $\tilde{S}_k = 2I_k$. In this case, there is a simple and efficient way to perform Babai's nearest plane algorithm [17]. In this algorithm, we first pre-compute two large sets of samples from $\mathcal{D}_{2\mathbb{Z},\sigma}$ and $\mathcal{D}_{2\mathbb{Z}+1,\sigma}$. The sampling algorithm draws each coefficient of x one by one from the appropriate set. This approach requires less storage space but takes k steps to complete. Naturally, there is a hybrid approach that pre-computes samples from $\mathcal{D}_{\mathbb{Z}^l,\sigma}$ for some $l < k$ and fills in the coefficients of x in blocks of l.

To perform Gaussian sampling in the ring setting, we can of course use the sampling algorithm for general lattices and perform the permutation mentioned above to the preimage. More formally, recall that our task is to sample a vector of ring elements $\mathbf{x}$ from $\Lambda_q^{\mathbf{u}}(\mathbf{g}^T) = \{\mathbf{x} \in R^k : \mathbf{g}^T\mathbf{x} = \mathbf{u} \bmod q\}$ where $\mathbf{u} \in R_q$. That is, $\sum_{i=0}^{k-1} 2^i \mathbf{x}_i = \mathbf{u}$. By expanding $\mathbf{x}_i$ in the power basis $1, x, x^2, \ldots, x^{n-1}$, this is equivalent to $\sum_{i=0}^{k-1} 2^i x_{ij} = u_j$ for $j = 0, 1, \ldots, n-1$, where $\mathbf{x}_i = \sum_{j=0}^{n-1} x_{ij} x^j$ and $u = \sum_{j=0}^{n-1} u_j x^j$. Thus, we can use the same sampling algorithms for each equation $\sum_{i=0}^{k-1} 2^i x_{ij} = u_j$ in the ring setting, since x_{ij} and u_j are integers modulo q. However, notice that the reduction of ring-LWE in [2,3,14] requires that $q = 1 \bmod 2n$, which means that q cannot be a power of 2. Therefore, practically we can only use the first approach for Gaussian sampling in the ring setting.

4 Trapdoors in Ideal Lattices

Analogous to the trapdoors for general lattices defined in [12], we extend the notion to the ring setting. This includes the derivation of old-type trapdoors from

Gentry *et al.* [10] (Sect. 4.1), the generation of (new-type) trapdoors (Sect. 4.2), ring-LWE inversion (Sect. 4.3), Gaussian sampling (Sect. 4.4) and trapdoors delegation (Sect. 4.5).

Definition 2. *Let* $\mathbf{a} \in R_q^{l+k}$ *and* $\mathbf{g} \in R_q^k$. *A* $\mathbf{g}$*-trapdoor for* $\mathbf{a}$ *is a collection of linearly independent vectors of ring elements* $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_k) \in R_q^{l \times k}$ *such that* $\mathbf{a}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} = \mathbf{h}\mathbf{g}^T$, *for some non-zero ring element* $\mathbf{h} \in R_q$. $\mathbf{h}$ *is referred as the tag or label of the trapdoor. The qulity of the trapdoor is measured by its largest singular value* $s_1(\mathbf{R})$, *which is computed as the largest singular value of the matrix obtained by interpreting* $\mathbf{R}$ *as a matrix in* $\mathbb{Z}_q^{ln \times kn}$.

4.1 Derivation of Old Trapdoors

Lemma 1. *Let* $\mathbf{g} \in R_q^k$ *and* $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_k) \in R^{k \times k}$ *be linearly independent with* $\mathbf{g}^T \mathbf{s}_i = \mathbf{0} \in R_q$ *for* $i = 1, \ldots, k$. *Let* $\mathbf{a} \in R_q^{l+k}$ *have trapdoor* $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_k) \in R^{k \times k}$ *with tag* $\mathbf{h} \in R_q$. *Then the lattice* $\Lambda_q^{\perp}(\mathbf{a}^T)$ *is generated by*

$$\mathbf{S}_{\mathbf{a}} = \begin{bmatrix} \mathbf{I}_l & \mathbf{R} \\ \mathbf{0}_{k \times l} & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{I}_l & \mathbf{0}_{l \times k} \\ \mathbf{W} & \mathbf{S} \end{bmatrix},$$

where $\mathbf{W} \in R^{k \times l}$ *is an arbitrary solution to* $\mathbf{g}^T \mathbf{W} = -\mathbf{h}^{-1} \mathbf{a}^T \left[\mathbf{I}_l | \mathbf{0}_{l \times k}\right]^T \mod q$.

Moreover, the basis $\mathbf{S}_{\mathbf{a}}$ *satisfies* $\|\widetilde{\mathbf{S}_{\mathbf{a}}}\| \leq s_1 \left(\begin{bmatrix} \mathbf{I}_l & \mathbf{R} \\ \mathbf{0}_{k \times l} & \mathbf{I}_k \end{bmatrix} \right) \cdot \|\tilde{\mathbf{S}}\| \leq (s_1(\mathbf{R}) + 1) \cdot \|\tilde{\mathbf{S}}\|$, *when* $\mathbf{S}_{\mathbf{a}}$ *is orthogonalized in suitable order and interpreted as a matrix in* $\mathbb{Z}^{[(l+k)n] \times [(l+k)n]}$ *by the coefficient embedding.*

Proof. Compared to the derivation in general lattices, the non-trivial part is to construct a matrix $\mathbf{W}$ of ring elements, or equivalently, a matrix W consisting of $k \times l$ blocks of $n \times n$ rotational matrices. Otherwise, the rest of the proof follows the proof of [12, Lemma 5.3]. To construct such a matrix $\mathbf{W}$, let $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{l+k})^T \in R_q^{l+k}$ and let

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_{1,1} & \cdots & \mathbf{w}_{1,l} \\ \vdots & \ddots & \vdots \\ \mathbf{w}_{k,1} & \cdots & \mathbf{w}_{k,l} \end{bmatrix}$$

where $\mathbf{w}_{i,j} \in R_q$.

Now, $\mathbf{g}^T \mathbf{W} = -\mathbf{h}^{-1} \mathbf{a}^T \left[\mathbf{I}_l | \mathbf{0}_{l \times k}\right]^T \mod q$ implies

$$[1|2|\ldots|2^{k-1}]\mathbf{W} = [1|2|\ldots|2^{k-1}] \begin{bmatrix} \mathbf{w}_{1,1} & \cdots & \mathbf{w}_{1,l} \\ \vdots & \ddots & \vdots \\ \mathbf{w}_{k,1} & \cdots & \mathbf{w}_{k,l} \end{bmatrix} = -\mathbf{h}^{-1} \left[\mathbf{a}_1 \ \mathbf{a}_2 \ \ldots \ \mathbf{a}_l.\right]$$

This equation implies that for each $j = 1, \ldots, l$, we need to independently solve

$$[1|2|\ldots|2^{k-1}] \begin{bmatrix} \mathbf{w}_{1,j} \\ \vdots \\ \mathbf{w}_{k,j} \end{bmatrix} = -\mathbf{h}^{-1}\mathbf{a}_i \in R_q.$$

By expanding $\mathbf{w}_{i,j}$ and $\mathbf{a}_j$ with respect to the power basis $1, x, x^2, \ldots, x^{n-1}$, the problem is equivalent to solving the system for each coefficient independently. □

Although the derivation of the old-type trapdoors for ideal lattices is merely theoretical, it solves an open problem in [10] which asked how trapdoors can be generated together with random looking ideal lattices.

4.2 Generation of New Trapdoor

As in [12], the derivation of old trapdoors from the new trapdoors is just a proof of concept and will not be used in the rest of this work. In this subsection, we extend their trapdoors for general lattices to our ring version in Algorithm 4.

Algorithm 4. $\mathsf{ringGenTrap}^{\mathcal{D}}(\mathbf{a}_0, \mathbf{h})$
Input:

- *a vector of ring elements* $\mathbf{a}_0 = (\mathbf{a}_1, \ldots, \mathbf{a}_l)^T \in R_q^l$*;*
- *a non-zero ring element* $\mathbf{h} \in R_q$*;*
- *a distribution* $\chi^{l \times k}$ *over* $R^{l \times k}$*. (If no particular* $\mathbf{a}_0$*,* $\mathbf{h}$ *are given as input, then the algorithm may choose them itself, e.g. picking* $\mathbf{a}_0 \leftarrow R_q^l$ *uniformly, and setting* $\mathbf{h} = 1$*.)*

Output:

- *a vector of ring elements* $\mathbf{a} = (\mathbf{a}_0^T, \mathbf{a}_1^T)^T \in R_q^{l+k}$*;*
- *a trapdoor* $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_k) \in R^{l \times k}$ *with tag* $\mathbf{h} \in R_q$*.*

1. *Choose a collection of linearly independent vectors of ring elements* $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_k) \in R^{l \times k}$ *from distribution* $\chi^{l \times k}$*,*
2. *Output* $\mathbf{a} = (\mathbf{a}_0^T, \mathbf{h}\mathbf{g}^T - \mathbf{a}_0^T\mathbf{R})^T \in R_q^{l+k}$ *and trapdoor* $\mathbf{R} \in R^{l \times k}$*.*

Moreover, the distribution of $\mathbf{a}$ *is close to uniform (either statistically or computationally) as long as the distribution of* $(\mathbf{a}_0^T, -\mathbf{a}_0^T\mathbf{R})$ *is.*

The correctness of Algorithm 4 is immediate. To show that the distribution of $(\mathbf{a}_0^T, -\mathbf{a}_0^T\mathbf{R})$ is close to uniform, we need to show that the distribution of $\mathbf{a}_0^T\mathbf{R}$ is close to uniform and hence is independent to that of $\mathbf{a}_0$, or equivalently the distribution of $\mathbf{a}_0^T\mathbf{r}_i$ is close to uniform and independent to that of $\mathbf{a}_0$ for all i. As for the trapdoors for general lattices, the uniformity of $\mathbf{a}$ can be instantiated to be either statistical by using a regularity lemma or computational by the ring-LWE assumption.

Lemma 2. *[3, Sect. 7] (Regularity Lemma) Let* $\mathbf{a}_i \leftarrow R_q$ *and* $\mathbf{r}_i \leftarrow \chi$ *for* $i = 1, \ldots, l$. *Then* $\mathbf{b} = \sum_{i=1}^{l} \mathbf{a}_i \mathbf{r}_i$ *is within* $2^{-\Omega(n)}$ *statistical distance to the uniform distribution over* R_q. *Moreover, the case where* $l = 2$ *corresponds to the normal form of ring-*LWE.

4.3 ring-LWE Inversion from New Trapdoors

Given a trapdoor $\mathbf{R}$ for $\mathbf{a} \in R_q^{l+k}$ and a $\mathsf{PLWE}_{f,q,\chi}^{(l)}$ instance $\mathbf{b} = \mathbf{as} + \mathbf{e} \bmod q$, the ring-LWE inversion algorithm given in Algorithm 5 is to find the solution $\mathbf{s}$ to the instance.

Algorithm 5. $\mathsf{ringInvert}^{\mathcal{O}}(\mathbf{R}, \mathbf{a}, \mathbf{b})$
Input:

- *an oracle* $\mathcal{O}$ *for inverting the function* $\alpha_{\mathbf{g}}(\mathbf{s}', \mathbf{e}')$ *when* $\mathbf{e}' \in R^k$ *is suitably small;*
- *a vector of ring element* $\mathbf{a} \in R_q^{l+k}$*;*
- $\mathbf{g}$*-trapdoor* $\mathbf{R} \in R^{l \times k}$ *for* $\mathbf{a}$ *with tag* $\mathbf{h}$*;*
- *vector* $\mathbf{b} = \mathbf{as} + \mathbf{e}$ *for any* $\mathbf{s} \in R_q$ *and suitably small* $\mathbf{e} \in R^{l+k}$.

Output: $\mathbf{s}$ *and* $\mathbf{e}$.

1. *Get* $(\mathbf{s}', \mathbf{e}') \leftarrow \mathcal{O}([\mathbf{R}^T | \mathbf{I}_k]\mathbf{b})$.
2. *return* $\mathbf{s} = \mathbf{h}^{-1}\mathbf{s}'$ *and* $\mathbf{e} = \mathbf{b} - \mathbf{as}$ *(interpreted as a vector in* R^{l+k} *with where each entry has coefficients in* $[-\frac{q}{2}, \frac{q}{2})$*).*

The correctness of Algorithm 5 is indicated by Theorem 6 stated below.

Theorem 6. *Suppose that* $\mathcal{O}$ *in Algorithm 5 correctly inverts* $\alpha_{\mathbf{g}}(\mathbf{s}', \mathbf{e}')$ *for any small error vector* $\mathbf{e}' \in \mathcal{D}^k_{\mathbb{Z}^n, \sigma\sqrt{l\sigma^2 \cdot \omega(\log n) + k}}$. *Then for any* $\mathbf{s} \in R_q$ *and* $\mathbf{e} \leftarrow \chi^{l+k}$, *Algorithm 5 correctly inverts* $\alpha_{\mathbf{a}}(\mathbf{s}, \mathbf{e})$ *with overwhelming probability over the choice of* $\mathbf{e}$.

Proof. We first show that $\mathbf{b}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix}$ gives a correct input to the oracle $\mathcal{O}$.

$$\begin{aligned}
\mathbf{b}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} &= (\mathbf{a}^T\mathbf{s} + \mathbf{e}^T) \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} \\
&= \mathbf{a}^T\mathbf{s} \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} + \mathbf{e}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} \\
&= \mathbf{s}[\mathbf{a}_0^T | \mathbf{h}\mathbf{g}^T - \mathbf{a}_0^T\mathbf{R}] \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} + \mathbf{e}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} \\
&= \mathbf{s}(\mathbf{a}_0^T\mathbf{R} + \mathbf{h}\mathbf{g}^T - \mathbf{a}_0^T\mathbf{R}) + \mathbf{e}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} \\
&= \mathbf{g}^T\mathbf{h}\mathbf{s} + \mathbf{e}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} \\
&= \mathbf{g}^T\mathbf{s}' + \mathbf{e}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix}
\end{aligned}$$

Now we need to show that $\mathbf{e}' = \mathbf{e}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix}$ has the appropriate distribution. Consider

$$\mathbf{e}'_j = \sum_{i=1}^{l} \mathbf{e}_i \mathbf{r}_{ij} + \sum_{i=l+1}^{l+k} \mathbf{e}_i \quad \forall j = 1, \ldots, k$$

where $\mathbf{e}'_j = j$-th component of $\mathbf{e}'$, $\mathbf{e}_i = i$-th component of $\mathbf{e}$, $\mathbf{r}_{ij} = ij$-th component of $\mathbf{R}$. Since each entry of $\mathbf{e}$ and $\mathbf{R}$ are sampled from $\chi = \mathcal{D}_{\mathbb{Z}^n,\sigma}$, then the distribution of $\mathbf{e}_i \mathbf{r}_{ij}$ is statistically close to $\mathcal{D}_{\mathbb{Z}^n,\sigma^2 \cdot \omega(\sqrt{\log n})}$ [3, Lemma 8.7]. Hence, the distribution of $\mathbf{e}'_j$ is statistically close to $\mathcal{D}_{\mathbb{Z}^n,\sigma\sqrt{l\sigma^2 \cdot \omega(\log n)+k}}$ [3, Lemma 8.6]. Therefore, the distribution of $\mathbf{e}'$ has the correct distribution. □

4.4 Gaussian Sampling from New Trapdoors

Given a trapdoor $\mathbf{R}$ for $\mathbf{a} \in R_q^{l+k}$ and $\mathbf{u} = \beta_{\mathbf{a}}(\mathbf{x}) = \mathbf{a}^T\mathbf{x} \bmod q$, the Gaussian Sampling algorithm given in Algorithm 7 is to find the solution $\mathbf{x}$ to the instance.

Algorithm 7. $\mathsf{ringSampleD}^{\mathcal{O}}(\mathbf{R}, \mathbf{a}_0, \mathbf{h}, \mathbf{u}, \sigma)$
Input:
Offline phase:

- *an oracle* $\mathcal{O}(\mathbf{v})$ *for Gaussian sampling over a desired coset* $\Lambda_q^{\mathbf{v}}(\mathbf{g}^T)$ *with parameter* σ*, where* $\mathbf{v} \in R_q$*;*
- *a vector of ring elements* $\mathbf{a}_0 \in R_q^l$*;*
- *a trapdoor* $\mathbf{R} \in R^{l \times k}$*;*
- *a Gaussian parameter* σ*.*

Online phase:

- *a non-zero tag* $\mathbf{h} \in R_q$ *defining* $\mathbf{a} = (\mathbf{a}_0^T, \mathbf{h}\mathbf{g}^T - \mathbf{a}_0^T\mathbf{R})^T \in R_q^{l+k}$ *(*$\mathbf{h}$ *may instead be provided in the offline phase, if it is known);*
- *syndrome* $\mathbf{u} \in R_q$*.*

Output: *A vector* $\mathbf{x}$ *drawn from a distribution statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{v}}(\mathbf{a}_0^T),\sigma'}$ *for some Gaussian parameter* σ'*.*
Offline phase:

1. *Choose fresh perturbations* $\mathbf{p}_1 \leftarrow \chi_1^l$ *and* $\mathbf{p}_2 \leftarrow \chi_2^k$ *for some distributions* χ_1 *and* χ_2 *over* R*.*
2. *Compute* $\mathbf{w}_0 = \mathbf{a}_0^T(\mathbf{p}_1 - \mathbf{R}\mathbf{p}_2) \in R_q$ *and* $\mathbf{w}_1 = \mathbf{g}^T\mathbf{p}_2 \in R_q$*.*

Online phase:

1. *Let* $\mathbf{v} \leftarrow \mathbf{h}^{-1}(\mathbf{u} - \mathbf{w}_0) - \mathbf{w}_1 = \mathbf{h}^{-1}(\mathbf{u} - \mathbf{a}^T\mathbf{p}) \in R_q$*, and choose* $\mathbf{z} \leftarrow \mathcal{D}_{\Lambda_q^{\mathbf{v}}(\mathbf{g}^T),\sigma}$ *by calling* $\mathcal{O}(\mathbf{v})$*.*
2. *Return* $\mathbf{x} \leftarrow \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_k \end{bmatrix} \mathbf{z}$*.*

Theorem 8. *Algorithm 7 is correct.*

Proof. Let $\mathbf{x} \leftarrow \mathsf{ringSampleD}^{\mathcal{O}}(\mathbf{R}, \mathbf{a}_0, \mathbf{h}, \mathbf{u}, \sigma)$. Then

$$\begin{aligned}\mathbf{a}^T\mathbf{x} &= [\mathbf{a}_0^T | \mathbf{h}\mathbf{g}^T - \mathbf{a}_0^T\mathbf{R}] \left(\begin{bmatrix}\mathbf{p}_1 \\ \mathbf{p}_2\end{bmatrix} + \begin{bmatrix}\mathbf{R} \\ \mathbf{I}_k\end{bmatrix} \mathbf{z} \right) \\ &= \mathbf{a}_0^T\mathbf{p}_1 + \mathbf{a}_0^T\mathbf{R}\mathbf{z} + \mathbf{h}\mathbf{g}^T\mathbf{p}_2 - \mathbf{a}_0^T\mathbf{R}\mathbf{p}_2 + \mathbf{h}\mathbf{g}^T\mathbf{z} - \mathbf{a}_0^T\mathbf{R}\mathbf{z} \\ &= \mathbf{a}_0^T(\mathbf{p}_1 - \mathbf{R}\mathbf{p}_2) + \mathbf{h}\mathbf{g}^T\mathbf{p}_2 + \mathbf{h}\mathbf{v} \\ &= \mathbf{w}_0 + \mathbf{h}\mathbf{w}_1 + \mathbf{u} - \mathbf{w}_0 - \mathbf{h}\mathbf{w}_1 \\ &= \mathbf{u}\end{aligned}$$

Now, consider $\mathbf{x} = \begin{bmatrix}\mathbf{p}_1 \\ \mathbf{p}_2\end{bmatrix} + \begin{bmatrix}\mathbf{R} \\ \mathbf{I}_k\end{bmatrix} \mathbf{z}$. Since each entry of $\mathbf{R}$ and $\mathbf{z}$ are sampled from $\chi = \mathcal{D}_{\mathbb{Z}^n,\sigma}$, and each entry of $\mathbf{p}_1$ and $\mathbf{p}_2$ are sampled from $\chi_1 = \mathcal{D}_{\mathbb{Z}^n,\sigma^2\cdot\omega\sqrt{\log n}}$ and $\chi_2 = \mathcal{D}_{\mathbb{Z}^n,\sigma\sqrt{\sigma^2(k+1)\cdot\omega(\log n)-1}}$, respectively, then the distributions of all entries of $\mathbf{x}$ are statistically close to $\chi' = \mathcal{D}_{\mathbb{Z}^n,\sigma'}$, where $\sigma' = \sigma^2\sqrt{k+1}\cdot\omega(\sqrt{\log n})$. [3, Lemma 8.6 & 8.7]. □

4.5 Trapdoors Delegation

Using the trapdoor of $\mathbf{a}$, there is an efficient trapdoor delegation algorithm given in Algorithm 9 that generates a trapdoor for the vector $(\mathbf{a}^T, \mathbf{a}_1^T)^T$.

Algorithm 9. $\mathsf{ringDelTrap}^{\mathcal{O}}(\mathbf{a}' = (\mathbf{a}^T, \mathbf{a}_1^T)^T, \mathbf{h}', \sigma)$
Input:

- *an oracle $\mathcal{O}$ for discrete Gaussian sampling over cosets of $\Lambda_q^{\perp}(\mathbf{a}^T)$ with parameter σ';*
- *a vector of ring elements $\mathbf{a}' = (\mathbf{a}^T, \mathbf{a}_1^T)^T \in R_q^{m+k}$;*
- *a non-zero ring element $\mathbf{h}' \in R_q$.*

Output: *a trapdoor $\mathbf{R} \in R^{(m+k)\times k}$ for $\mathbf{a}'$ with tag $\mathbf{h}'$.*

- *Using $\mathcal{O}$, sample each column of $\mathbf{R}$ independently from a discrete Gaussian with parameter σ' over the appropriate cosets of $\Lambda_q^{\perp}(\mathbf{a}^T)$, so that $\mathbf{a}^T\mathbf{R} = \mathbf{h}'\mathbf{g}^T - \mathbf{a}_1^T$.*

5 INDr-ID-CCA-Secure (H)IBE in Ideal Lattices

5.1 Identity-Based Encryption

Identity-based encryption (IBE) is a generalization of public-key encryption [18]. In an IBE, to encrypt a message to an identity *id*, the encrypter does not need to lookup the public key for the intended identity *id*. Instead, the encryption algorithm simply takes the public parameters, the identity *id* and the message as input and outputs a ciphertext encrypting the message to *id*. The identity *id* obtains its secret key derived from the master secret key through the key generation algorithm for decrypting all ciphertexts encrypted to *id*. Formally, the syntax of IBE is defined as follows.

Definition. An identity-based encryption (IBE) scheme consists of four PPT algorithms (Setup, Extract, Encrypt, Decrypt). The Setup algorithm outputs a public parameter PP and a master key MK. Using MK, the Extract algorithm extracts a secret key SK_{id} for an identity id. Unlike public-key encryption, the Encrypt algorithm in IBE can encrypt messages directly to an identity id. The user with identity id uses her secret key SK_{id} to Decrypt a ciphertext.

Hierarchical identity-based encryption (HIBE) is an extension of IBE such that an identity $ID = [id_1|\ldots|id_d]$ is a hierarchy of identities with depth d. There is an additional algorithm Derive that inputs a secret key $SK_{[id_1|\ldots|id_{j-1}]}$ in the $(j-1)$-th level and an identity $ID = [id_1|\ldots|id_j]$ in the j-th level and outputs the secret key SK_{ID} for identity ID.

Security. In addition to the (adaptive) chosen-plaintext-attack (CPA(2)) security or (adaptive) chosen-ciphertext-attack (CCA(2)) security as in public-key encryption, an (H)IBE scheme should also be secure against chosen-identity-attack (ID). A weaker security model called selective-identity-attack (sID) is also considered, where the adversary must choose the identity she is going to perform CPA(2) or CCA(2) before receiving the public parameter. The indistinguishability (IND) of ciphertexts under the combinations of attacks in the two sets of variants give us eight security model, namely, IND-ID-CCA(2), IND-sID-CCA(2), IND-ID-CPA(2) and IND-sID-CPA(2). In [8], a stronger security guarantee in which the ciphertexts are indistinguishable from random (INDr) elements in the ciphertext space is considered. This implies both semantic security (of the plaintext) and recipient anonymity.

From now on, we will focus on the INDr-ID-CCA security, which is modeled as a security game between a PPT simulator and a PPT adversary. In this game, the simulator first generates the public parameters PP and passes them to the adversary. The adversary is then granted the rights to query the secret keys SK_{id} for polynomially many identities id of its choice, and the rights to query the decryption of any ciphertext of its choice. After that, the adversary issues a challenge message to be encrypted to the identity id^*, whose secret key has never been queried before. The simulator replies by either encrypting the challenge message to id^* or generating a uniformly random ciphertext. The adversary wins the game if it can guess which among two ways the ciphertext is generated.

An INDr-ID-CCA2-secure HIBE of depth d can be obtained by combining an INDr-ID-CPA-secure HIBE of depth $d+1$ and a strong one-time signature scheme [19]. The rough idea is to encrypt the message to the "identity" $[id_1|id_2|\ldots|id_d|vk]$ where vk is the verification key of the one-time signature, and sign the ciphertext using the one-time signature. In particular, from IBE, we obtain a CCA2-secure public-key encryption scheme. We will omit the details here.

Applications. Most earlier (H)IBE schemes are realized by pairings. Readers can refer to [20,21] for reviews of those. Agrawal *et al.* proposed a lattice-based (H)IBE scheme in the standard model [8]. The ciphertext of their scheme can

be proven to be a random element in the ciphertext space, which implies receipt anonymity against user attacks. An anonymous IBE can be used to obtain a public-key searchable encryption scheme [22]. By applying Naor's transformation [18], we can also obtain a signature scheme from an IBE scheme.

5.2 Construction

Using the trapdoors for ideal lattices developed above, the CCA-secure public-key encryption provided in [12] and the INDr-ID-CPA-secure (H)IBE scheme in [8], we construct an INDr-ID-CCA-secure (H)IBE in ideal lattices. We only present the basic IBE scheme below, because the HIBE scheme can be obtained trivially by defining the Derive algorithm of HIBE to be the same as the $\mathsf{Extract}$ algorithm of the basic IBE in our case.

The basic IBE scheme is constructed as follows:

- $\mathsf{Setup}(1^\lambda)$
  - Sample $\mathbf{a}_{-1} \leftarrow R_q^l$, $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_t \leftarrow R_q^k$ and $\mathbf{h} \leftarrow R_q \setminus \{0\}$.
  - Sample $(\mathbf{a}, \mathbf{R}_{MK}) \leftarrow \mathsf{ringGenTrap}^{\mathcal{D}}(\mathbf{a}_{-1}, \mathbf{h})$.
  - Output $PP = (\mathbf{a}, \mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_t)$ and $MK = (\mathbf{h}, \mathbf{R}_{MK})$.
- $\mathsf{Extract}(PP, MK, id)$
  - Sample $\mathbf{h}_{id} \leftarrow R_q$.
  - Set $\mathbf{a}_{id} = \mathbf{a}_0 + \sum_{i=1}^{t} id_i \mathbf{a}_i$ and $\mathbf{f}_{id,\mathbf{h}_{id}} = (\mathbf{a}^T, \mathbf{a}_{id}^T)^T$.
  - Sample $\mathbf{R}_{id} \leftarrow \mathsf{ringDelTrap}^{\mathcal{O}}(\mathbf{f}_{id,\mathbf{h}_{id}}, \mathbf{h}_{id}, \sigma)$.
  - Output $SK_{id} = (\mathbf{h}_{id}, \mathbf{R}_{id})$.
- $\mathsf{Encrypt}(PP, id, \mathbf{m} \in R_p^k)$
  - Sample $\mathbf{h}' \leftarrow R_q$.
  - Set $\mathbf{a}_{id,\mathbf{h}'} = \mathbf{a}_{id} + \mathbf{h}'\mathbf{g}$.
  - Sample $\mathbf{s} \leftarrow \chi$, $\mathbf{e}_0 \leftarrow \chi^{l+k}$ and $\mathbf{R}_i \leftarrow \chi^{l \times k}$ for $i = 0, 1, \ldots, t$.
  - Set $\mathbf{R} = \mathbf{R}_0 + \sum_{i=1}^{t} id_i \mathbf{R}_i$.
  - Set $\mathbf{e}_1^T = -\mathbf{e}_0^T \mathbf{R}$.
  - Compute $\mathbf{u} = \mathbf{a}_0 \mathbf{s} + p\mathbf{e}_0$ and $\mathbf{v} = \mathbf{a}_{id,\mathbf{h}'} \mathbf{s} + p\mathbf{e}_1 + \mathbf{m}$.
  - Output $CT = (\mathbf{h}', \mathbf{u}, \mathbf{v})$.
- $\mathsf{Decrypt}(PP, SK_{id}, CT)$
  - Output $\perp$ if $\mathbf{h}' = -\mathbf{h}_{id}$.
  - Set $\mathbf{f}_{id,(\mathbf{h}_{id}+\mathbf{h}')} = (\mathbf{a}^T, \mathbf{a}_{id}^T + \mathbf{h}'\mathbf{g}^T)^T = (\mathbf{a}^T, (\mathbf{h}_{id} + \mathbf{h}')\mathbf{g}^T - \mathbf{a}^T \mathbf{R}_{id})^T$.
  - Compute $(\mathbf{s}, \mathbf{e}) \leftarrow \mathsf{ringInvert}^{\mathcal{O}}(\mathbf{R}_{id}, \mathbf{f}_{id,(\mathbf{h}_{id}+\mathbf{h}')}, (\mathbf{u}^T, \mathbf{v}^T)^T)$.
  - Compute $(\mathbf{0}_{1\times k}, \mathbf{m}^T)^T = \mathbf{e} \bmod p$.
  - Output $\mathbf{m}$.

Theorem 10. *By the* $\mathsf{PLWE}_{f,q,\chi}^{(l)}$ *assumption, the (H)IBE scheme stated above is INDr-ID-CCA-secure.*

Proof. The simulation strategy is a result of combining those from [8,12]. The simulator is given a PLWE instance $(\mathbf{a}, \mathbf{b})$. It chooses the rest of the public parameters $\mathbf{a}_0, \ldots, \mathbf{a}_t$ as follows:

– It samples $\mathbf{h}^* \leftarrow R_q$.
– It samples $\mathbf{R}_i \leftarrow \chi^{l \times k}$ and let $\mathbf{a}_i^T = \mathbf{h}_i \mathbf{g}^T - \mathbf{a}^T \mathbf{R}_i$, where $\mathbf{h}_0 = 1 - \mathbf{h}^*$ and $\mathbf{h}_i \leftarrow R_q$, for $i = 0, \ldots, t$.

Since the distribution of $\mathbf{a}$ is uniform and each entry of $\mathbf{R}_i$ is sampled from the distribution χ, by the regularity lemma (Lemma 2), the distribution of $\mathbf{a}_i$ is also uniform for all i.

To answer the queries for secret key of id, it simply returns $(\mathbf{h}_{id}, \mathbf{R}_{id})$ where $\mathbf{h}_{id} = 1 + \sum_{i=1}^{t} id_i \mathbf{h}_i - \mathbf{h}^*$ and $\mathbf{R}_{id} = \mathbf{R}_0 + \sum_{i=1}^{t} id_i \mathbf{R}_i$. Note that $\mathbf{a}_{id}^T = \mathbf{a}_0^T + \sum_{i=1}^{t} id_i \mathbf{a}_i^T = \mathbf{h}_{id} \mathbf{g}^T - \mathbf{a}^T \mathbf{R}_{id}$.

It answers the decryption queries $(id, CT = (\mathbf{h}, \mathbf{u}, \mathbf{v}))$ using the tag $\mathbf{h}_{id}$, the trapdoor $\mathbf{R}_{id}$ and the Decrypt algorithm. As long as $\mathbf{h} \neq \mathbf{h}^*$ or $-\mathbf{h}_{id} \neq \mathbf{h}^*$, the simulator can still simulate faithfully. Since $\mathbf{a}_i$ is uniformly random in the view of the adversary, $\mathbf{h}^*$ and $\mathbf{h}_i$ are all hidden from the adversary for all $i = 0, 1, \ldots, t$. Therefore the event $-\mathbf{h}_{id} = \mathbf{h} = \mathbf{h}^*$ only happens with negligible probability.

Finally, the challenge ciphertext for $(id^*, \mathbf{m}^*)$ is generated as $(\mathbf{h}^*, \mathbf{u}^*, \mathbf{v}^*)$ where $\mathbf{u}^* = \mathbf{b}$ and $\mathbf{v}^* = (-\mathbf{b}^T \mathbf{R}_{id^*} + \mathbf{m}^{*T})^T$.

Suppose that the $(\mathbf{a}, \mathbf{b})$ given in the PLWE instance is uniform, then the distribution of the challenge ciphertext $(\mathbf{h}^*, \mathbf{u}^*, \mathbf{v}^*)$ is also uniform. Otherwise, suppose $\mathbf{b} = \mathbf{a}\mathbf{s} + p\mathbf{e}$ for some $\mathbf{s}$ and $\mathbf{e}$ sampled from the appropriate distributions, then

$$\begin{aligned} \mathbf{v}^{*T} &= -\mathbf{b}^T \mathbf{R}_{id^*} + \mathbf{m}^{*T} \\ &= -\mathbf{s}\mathbf{a}^T \mathbf{R}_{id^*} - p\mathbf{e}^T \mathbf{R}_{id^*} + \mathbf{m}^{*T} \\ &= \mathbf{s}(\mathbf{a}_{id^*}^T - \mathbf{h}_{id^*} \mathbf{g}^T) - p\mathbf{e}^T \mathbf{R}_{id^*} + \mathbf{m}^{*T} \end{aligned}$$

By [8, Lemma 24], we have $\mathbf{h}_{id^*} = -\mathbf{h}^*$ with non-negligible probability. In such case, we have

$$\mathbf{v}^{*T} = \mathbf{s}(\mathbf{a}_{id^*}^T + \mathbf{h}^* \mathbf{g}^T) - p\mathbf{e}^T \mathbf{R}_{id^*} + \mathbf{m}^{*T}$$

which is distributed identically as valid ciphertexts do. □

6 Concluding Remarks

We detailed how to generate trapdoors for ideal lattices. We then use it to construct a new (H)IBE scheme. Our scheme has several improvement over that constructed by Agrawal *et al.* [8]:

– Our scheme is based on ideal lattices, therefore the size of the public parameters, master key and the identity secret key are reduced by a factor of n.
– Using the new trapdoor delegation algorithm, the size of the identity secret key grows linearly, rather than quadratically, in the depth of the hierarchy.
– Our scheme is secure against chosen-chiphertext-attack.

References

1. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, pp. 169–178 (2009)
2. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)
3. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 35–54. Springer, Heidelberg (2013)
4. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013)
5. Boneh, D., Freeman, D.M.: Homomorphic signatures for polynomial functions. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 149–168. Springer, Heidelberg (2011)
6. Boneh, D., Freeman, D.M.: Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 1–16. Springer, Heidelberg (2011)
7. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)
8. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
9. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)
11. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory Comput. Syst. **48**(3), 535–553 (2011)
12. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
13. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 617–635. Springer, Heidelberg (2009)
14. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
15. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34:1–34:40 (2009)
16. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC, pp. 575–584 (2013)
17. Babai, L.: On Lovász' Lattice Reduction and the Nearest Lattice Point Problem (Shortened Version). In: Mehlhorn, K. (ed.) STACS 1985. LNCS, vol. 182, pp. 13–20. Springer, Heidelberg (1985)

18. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003)
19. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. **36**(5), 1301–1328 (2007)
20. Chow, S.S.M.: Removing escrow from identity-based encryption. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 256–276. Springer, Heidelberg (2009)
21. Chow, S.S.M.: New privacy-preserving architectures for identity-/attribute-based encryption. Ph.D. thesis, New York University (2010)
22. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. J. Cryptol. **21**(3), 350–391 (2008)