12-2017

Compact hierarchical IBE from lattices in the standard model

Daode ZHANG

Fuyang FANG

Bao LI

Haiyang XUE
Singapore Management University, haiyangxue@smu.edu.sg

Bei LIANG

Citation

ZHANG, Daode; FANG, Fuyang; LI, Bao; XUE, Haiyang; and LIANG, Bei. Compact hierarchical IBE from lattices in the standard model. (2017). *Proceedings of the19th International Conference, ICICS 2017, Beijing, China, December 6-8*. 210-221.

Available at: https://ink.library.smu.edu.sg/sis_research/9183

Compact Hierarchical IBE from Lattices in the Standard Model

Daode Zhang[1,2,3], Fuyang Fang[4(✉)], Bao Li[1,2,3], Haiyang Xue[1], and Bei Liang[5]

[1] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing, China
{zhangdaode,lb}@is.ac.cn, xuehaiyang@iie.ac.cn
[2] State Key Laboratory of Information Security,
Institute of Information Engineering, Beijing, China
[3] Science and Technology on Communication Security Laboratory,
Chengdu, China
[4] Information Science Academy, China Electronics Technology Group Corporation,
Beijing, China
fuyang_fang@163.com
[5] Chalmers University of Technology, Gothenburg, Sweden
lbei@chalmers.se

Abstract. At Crypto'10, Agrawal *et al.* proposed a lattice-based selectively secure Hierarchical Identity-based Encryption (HIBE) scheme (ABB10b) with small ciphertext on the condition that λ (the length of identity at each level) is small in the standard model. In this paper, we present another lattice-based selectively secure HIBE scheme with depth d, using a gadget matrix $\mathbf{G}' \in \mathbb{Z}_q^{n\times n\lceil \log_b q\rceil}$ with enough large $b = 2^d$ to replace the matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ in the HIBE scheme proposed by Agrawal *et al.* at Eurocrypt'10. In our HIBE scheme, not only the size of ciphertext at level ℓ is $O(\frac{d+\ell}{\lambda d})$ larger than the size in ABB10b and at least $O(\ell)$ smaller than the sizes in the previous HIBE schemes except ABB10b, but also the size of the master public key is at least $O(d)$ times smaller than the previous schemes.

Keywords: Lattices · Hierarchical identity-based encryption
Selectively secure · Compact public parameters

1 Introduction

Hierarchical identity-based encryption (HIBE) proposed by Horwitz *et al.* [7,8] is an extension of identity-based encryption (IBE)[12], in which arbitrary string can be as the public key. In a HIBE scheme, an identity at level k of the hierarchy tree is provided with a private key from its parent identity and also can delegate private keys to its descendant identities, but cannot decrypt the message intended for other identities.

HIBE from Lattices: The first lattice-based HIBE scheme based on the Learning with Errors (LWE) problem [11] proposed by Cash *et al.* [5], using the basis

S. Qing et al. (Eds.): ICICS 2017, LNCS 10631, pp. 210–221, 2018.
https://doi.org/10.1007/978-3-319-89500-0_19

delegation technique for lattices. Agrawal *et al.* [1] proposed SampleLeft and SampleRight algorithms, then extended them and obtained another basis delegation technique, with which they constructed an efficient HIBE scheme with selective security in the standard model. However, the above basis delegation techniques will increase the dimension of lattice involved, as well as the size of ciphertext. Later, Agrawal *et al.* [2] proposed a different delegation mechanism, called "in place" delegation technique, which preserves the dimension of lattices. With this technique, they constructed two HIBE schemes with and without random oracles, and the dimension of lattices involved for all nodes in the hierarchy remained unchanged. Nevertheless, as they said in [2], the construction in the standard model was competitive with previous schemes in [1,5] only when the bits of identity ($|\boldsymbol{id}_i| = \lambda$) at level i in the hierarchy is small, e.g., $\lambda = 1$ at each level. Furthermore, as the length of identity increases, e.g., $\lambda = n$, the sizes of ciphertext, private key and master public key will be worse than the parameters in [1]. With the "in place" delegation technique, Fang *et al.* also utilized the Learning with Rounding (LWR) assumption [3,4] over small modulus to construct HIBE schemes. Thus, they possess the same restrictions as [2]. Micciancio and Peikert [10] introduced the notion of $\mathbf{G}$-trapdoor for lattices and proposed an efficient trapdoor delegation for lattices. With this technique, they can decrease the public key and ciphertext by 4 factors and the size of the delegated trapdoor grows only linearly with the dimension of lattices in the hierarchy, rather than quadratically in [1], but the ciphertext will be increased by $nk \log q$ bits node by node.

1.1 Our Contributions and Techniques

We apply a gadget matrix $\mathbf{G}' \in \mathbb{Z}_q^{n \times nk}$ defined in [10] into the basis delegation technique in [1] to construct a selectively secure HIBE scheme with small parameter based on the LWE problem in the standard model, where $k = \lceil \log_b q \rceil$, $b = 2^d$ and d is the maximum depth of the HIBE scheme.

The public parameter in our HIBE scheme needs to contain one matrix of the same dimension as $\mathbf{G}'$ (i.e., about $n \log_b q$) and the size of ciphertext is $n \log_b q \log q \approx \frac{1}{d} \cdot n \log^2 q$ for each level of the hierarchy. However, we obtain this improvement at the cost of increasing the size of private key. Thus, the parameters in our HIBE are the trade-off of the sizes of the public parameter and private keys. Next, we compare our scheme with the previous schemes in following Table 1.

From Table 1, the advantages of our HIBE scheme are:

1. The size of the master public key in [1,10] is reduced by a factor of $O(d)$;
2. The sizes of the ciphertext and lattice dimension at level ℓ are $\frac{d}{d+\ell} \cdot \ell = O(\ell)$ times smaller than the sizes in [1,10] and $\frac{d+\ell}{d} < 2$ times larger than the sizes in [2] on the condition that $\lambda = 1$. In particular, the parameters in ABB10b except the private key are competitive with our HIBE scheme only when $\lambda = 1$.

Table 1. Comparison of Lattice-based selective-id secure HIBE schemes in the standard model. In this table, d is the maximum depth of HIBE schemes and ℓ be the depth of the identity in query. $|ct|$ denotes the size of ciphertext at level ℓ. $|mpk|$ denotes the the size of the master public key in scheme. $|\mathsf{SK}_{\boldsymbol{id}}|$ denotes the size of the private key ar level ℓ. Error rate $(1/\alpha)$ denotes the security of LWE problem. The last columns denotes the lattice dimension involved at level ℓ. In order to compare the HIBE schemes, we let λ be the number of bits in each component of the identity.

Schemes	$\lvert ct\rvert$	$\lvert mpk\rvert$	$\lvert\mathsf{SK}_{\boldsymbol{id}}\rvert$	Error rate $1/\alpha$	Lattice dimension
[5]	$\tilde{O}(\lambda\ell nd^2)$	$\tilde{O}(\lambda n^2d^3)$	$\tilde{O}(\lambda^2\ell^3n^2d^2)$	$\tilde{O}(d^d(\lambda n)^{d/2})$	$\tilde{O}(\lambda\ell nd)$
[1]	$\tilde{O}(\ell nd^2)$	$\tilde{O}(n^2d^3)$	$\tilde{O}(\ell^3n^2d^2)$	$\tilde{O}(d^dn^{d/2})$	$\tilde{O}(\ell nd)$
[2]	$\tilde{O}(\lambda^2nd^2)$	$\tilde{O}(\lambda^3n^2d^3)$	$\tilde{O}(\lambda^3\ell n^2d^2)$	$\tilde{O}((\lambda dn)^{\lambda d+d/2})$	$\tilde{O}(\lambda nd)$
[10]	$\tilde{O}(\ell nd^2)$	$\tilde{O}(n^2d^3)$	$\tilde{O}(\ell n^2d^2)$	$\tilde{O}(d^dn^{d/2})$	$\tilde{O}(\ell nd)$
Our HIBE	$\tilde{O}(nd(d+\ell))$	$\tilde{O}(n^2d^2)$	$\tilde{O}((\frac{d}{\log n}+\ell)n^2(d+\ell)^2)$	$\tilde{O}((4d)^{d/2}n^{d/2})$	$\tilde{O}(n(d+\ell))$

And the disadvantages of our scheme are

1. The size of the private key at level ℓ is $\frac{d+\ell\log n}{\ell\log n}\cdot(\frac{d+\ell}{d})^2 = O(\frac{d}{\ell\log n}+1)$ times larger than [10] and the maximum ratio can reach to $O(\frac{d}{\log n}+1)$when $\ell = 1$.
2. The error rate $1/\alpha$ is lightly smaller than the sizes in [1,10] when $d > 4$.

Analysis: Before explaining why this modification works, let us firstly describe the reason that the sizes of ciphertexts in [1,10] increase as mentioned above. In [1], the identity-based encryption matrix for identity $\boldsymbol{id} = (\boldsymbol{id}_1,\cdots,\boldsymbol{id}_\ell) \in (\{0,1\}^\lambda)^\ell$ is

$$\mathbf{F}_{\boldsymbol{id}} = [\mathbf{A}|\mathbf{A}_1 + \mathsf{H}(\boldsymbol{id}_1)\mathbf{B}|\cdots|\mathbf{A}_\ell + \mathsf{H}(\boldsymbol{id}_\ell)\mathbf{B}] \in \mathbb{Z}_q^{n\times(\ell+1)m}$$

where $\mathbf{A},\mathbf{A}_1,\cdots,\mathbf{A}_\ell,\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and $m = O(n\log q)$. The difference in [10] is that the matrix $\mathbf{B}$ is replaced by a gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times nk}$, that is,

$$\mathbf{F}_{\boldsymbol{id}} = [\mathbf{A}|\mathbf{A}_1 + \mathsf{H}(\boldsymbol{id}_1)\mathbf{G}|\cdots|\mathbf{A}_\ell + \mathsf{H}(\boldsymbol{id}_\ell)\mathbf{G}|\mathbf{A}_{\ell+1}] \in \mathbb{Z}_q^{n\times(m+(\ell+1)nk)}$$

where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{A}_1,\cdots,\mathbf{A}_\ell \in \mathbb{Z}_q^{n\times\ell k}$ and $k = \lceil\log q\rceil$. Obviously, the ciphertext in [1,10] will increase $m = O(n\log q)$ and $k = n\lceil\log q\rceil$ elements in $\mathbb{Z}_q$ to each level in the hierarchy, respectively.

The size of the public parameters of the HIBE scheme in [10] is

$$(m + dnk)n\log q = (O(n\log q) + dnk)n\log q = (O(1) + d)\cdot n^2\log^2 q$$

where d is the maximum depth of the HIBE scheme and $O(1)$ here satisfies $O(1) \geq 2$ is a small constant. That is, the parameter d plays the important role on the size of the public parameters.

The straight modification is to replace $\mathbf{B}$ and $\mathbf{G}$ with another matrix, which has a short basis as trapdoor but with smaller columns. We know that the gadget matrix $\mathbf{G}$ has special structure that can be simply modified. The widely used version of $\mathbf{G}$ is defined as

$$\mathbf{G} = \mathbf{g}^t \otimes \mathbf{I}_n \in \mathbb{Z}_q^{n \times nk}$$

where $\mathbf{g}^t = (1, 2, \cdots, 2^{k-1})$ and $k = \lceil \log q \rceil$. Lattice $\Lambda^\perp(\mathbf{G})$ has a short basis $\mathbf{S}$ and $\|\tilde{\mathbf{S}}\| \le \sqrt{5}$. In fact, a generalized notion of gadget $\mathbf{G}$ provided in [10] is defined as

$$\mathbf{G} = \mathbf{g}^t \otimes \mathbf{I}_n \in \mathbb{Z}_q^{n \times nk}$$

where $\mathbf{g}^t = (1, b, \cdots, b^{k-1})$ and $k = \lceil \log_b q \rceil$. Then lattice $\Lambda^\perp(\mathbf{G}')$ has a short basis $\mathbf{S}'$ and $\|\tilde{\mathbf{S}}'\| \le \sqrt{b^2+1}$.

If we let b be large enough, then k can be small enough. How small should be k to choose in the HIBE scheme? What we want is that the item dbk is approximate $n \log q$. If we set $b = 2^d$, then we have

$$dnk = dn\lceil \log_b q \rceil = dn(\frac{1}{d}\log q + e) = n \log q + dne$$

where $e \in [-1/2, 1/2)$ and the modulus q in [1,10] is at least $\tilde{O}(n^{d/2})$ and $\log q = O(d \cdot \log n) \gg de$. Therefore, we have $dnk \approx n \log q$ and we can imply that

$$(m + dnk)n \log q = O(n^2 \log^2 q)$$

When using the gadget matrix $\mathbf{G}'$, the identity-based encryption matrix is similar with [10]. However, we do not adopt the DelTrap algorithm to delegate the private key for identities. Because the Gaussian parameter σ_ℓ in DelTrap algorithm requires that $\sigma_\ell \ge s_1(\mathbf{R}_{\boldsymbol{id}_{\ell-1}}) \cdot \|\widetilde{\mathbf{S}'}\|\omega(\sqrt{\log n})$ and then the output $s_1(\mathbf{R}_{\boldsymbol{id}_\ell}) \le \sigma_\ell \cdot \sqrt{m}$ will be proportion to $\|\widetilde{\mathbf{S}'}\|^\ell = 2^{\ell d}$ which could be larger than q. Therefore, we still utilize the SampleBasisRight algorithm in the security proof.

The cost of this modification is that the norm of basis increases from $\sqrt{5}$ to $\sqrt{b^2+1}$, which will affect the bound of Gaussian parameter of SampleBasisRight algorithm in the security of proof. The Gaussian parameter σ_ℓ of SampleBasisRight algorithm in level ℓ should satisfy

$$\sigma_\ell \ge s_1(\mathbf{R}_{\boldsymbol{id}}) \cdot \|\tilde{\mathbf{S}}'\| \cdot \omega(\sqrt{\log n}) \ for \ \ell = 1, \cdots, d$$

It seems that $\sigma_\ell \gg s_1(\mathbf{R}_{\boldsymbol{id}}) \cdot \sqrt{5} \cdot \omega(\sqrt{\log n}) = s_1(\mathbf{R}) \cdot \|\tilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$, which maybe deteriorate the parameters of our HIBE scheme. Fortunately, this intuition is not true for our scheme.

In the Subsect. 3.2 for the correctness of our scheme, we give the bound that the Gaussian parameter $\sigma_{\ell+1}$ at level $\ell + 1$ should satisfy

$$\sigma_{\ell+1} \ge s_1(\mathbf{R}_{\boldsymbol{id}}) \cdot \|\tilde{\mathbf{S}}'\| \cdot (m + \ell nk)^{\frac{\ell}{2}} \cdot \omega(\log^{\frac{\ell}{2}}(m + \ell nk))$$

to meet the conditions of SampleBasisLeft and SampleBasisRight algorithms, where $s_1(\mathbf{R}_{\boldsymbol{id}}) \le O(\sqrt{m + \ell nk})$. Meanwhile, the correctness requires that

$$\alpha_\ell q \omega(\sqrt{\log n}) + \alpha_\ell q \sigma_\ell (m + \ell nk)^{3/2} \cdot \omega(\sqrt{\log(m + \ell nk)}) \le q/5$$

and the hardness of LWE requires that $\alpha_\ell q \geq 2\sqrt{n}$.

Without loss of generality, we can set $m = 2n \log q$. Hence, the modulus q should satisfy

$$\begin{aligned} & q \geq \sqrt{n} \cdot (m + knd)^{(d+3)/2} \cdot b \cdot \omega(\log^{\frac{d}{2}}(m + knd)) \\ \Rightarrow\ & q \geq \sqrt{n} \cdot (2m)^{d/2} \cdot 2^d \cdot \omega(\log^{\frac{d}{2}}(2m)) \\ \Rightarrow\ & q \geq \sqrt{n} \cdot (dn \log n)^{d/2} \cdot 2^d \cdot \omega(\log^{\frac{d}{2}}(2m)) \\ \Rightarrow\ & q \geq \tilde{O}((4d)^{d/2} \cdot n^{d/2}) \end{aligned}$$

which is sufficient for our HIBE scheme and lightly smaller than the sizes of q in [1,10] if $d > 4$.

Furthermore, we decrease the columns of $\mathbf{G}$ from $n\lceil \log q \rceil$ to $n\lceil \log_b q \rceil$ so that the sizes of ciphertext and the master key increase linearly with $n\lceil \log_b q \rceil$, rather than m in [1] or $n\lceil \log q \rceil$ in [10] for each hierarchy and $m + \ell nk < m + dnk < 2m = O(dn \log n)$. This is why the sizes of ciphertext and the master key decrease by about ℓ and d factors, respectively.

2 Preliminaries

Let n be the security parameter and we use $negl(n)$ to denote an arbitrary negligible function $f(n)$ where $f(n) = o(n^{-c})$ for every fixed constant c. We say that a probability is *overwhelming* if it is $1 - negl(n)$. We use $poly(n)$ and $\widetilde{O}(n)$ to denote an unspecified function $f(n) = O(n^c)$ and $f(n) = O(n \cdot log^c n)$ respectively for some constant c. We use A $\approx_{c(s)}$ B to denote a distribution A is computationally (statistically) indistinguishable from a distribution B. Let $\mathbb{Z}_q$ be a q-ary finite field for a prime $q \geq 2$. The $s_1(\mathbf{R})$ are called the singular values of $\mathbf{R}$ and $s_1(\mathbf{R}) = \max_{\boldsymbol{u}} \|\mathbf{R}\boldsymbol{u}\| = \max_{\boldsymbol{u}} \|\mathbf{R}^t\boldsymbol{u}\| \leq \|\mathbf{R}\|, \|\mathbf{R}^t\|$, where the maximum are taken over all unit vectors $\boldsymbol{u}$. Let $a \overset{\$}{\leftarrow} \mathbb{Z}_q$ denote that a is randomly chosen from $\mathbb{Z}_q$.

2.1 Hierarchical IBE

An identity-based encryption (IBE) scheme with the message space $\mathcal{M}$ can be defined by a tuple of PPT algorithms (KeyGen, $\mathsf{Extract}$, Enc, Dec) as below:

- $\mathsf{KeyGen}(1^n) \rightarrow (mpk, msk)$: The probabilistic algorithm $\mathsf{KeyGen}(1^n)$ generates (mpk, msk), which denotes public key and master key respectively.
- $\mathsf{Extract}(mpk, msk, \boldsymbol{id}) \rightarrow SK_{\boldsymbol{id}}$: The $\mathsf{Extract}$ algorithm uses the master key to extract a private key $SK_{\boldsymbol{id}}$ corresponding to a given identity $\boldsymbol{id}$.
- $\mathsf{Enc}(mpk, \boldsymbol{id}, \boldsymbol{m}) \rightarrow \boldsymbol{c}$: Given a message $\boldsymbol{m} \in \mathcal{M}$ and an identity $\boldsymbol{id}$, the probabilistic algorithm Enc uses the public key mpk to encrypt the message with respect to the identity $\boldsymbol{id}$ and outputs a ciphertext $\boldsymbol{c}$.
- $\mathsf{Dec}(SK_{\boldsymbol{id}}, \boldsymbol{id}, \boldsymbol{c}) \rightarrow \boldsymbol{m}\ or \perp$: Given a ciphertext $\boldsymbol{c}$ with respect to an identity $\boldsymbol{id}$, the deterministic algorithm Dec uses the private key $SK_{\boldsymbol{id}}$ to recover the message $\boldsymbol{m}$. When the ciphertext $\boldsymbol{c}$ is invalid, the algorithm outputs $\perp$.

In a HIBE scheme of depth d, there is a fifth algorithm Derive, which takes as input an identity $\boldsymbol{id} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_\ell\}$ at depth $\ell \leq d$ and the private key $\mathsf{SK}_{\boldsymbol{id}_{\ell-1}}$ of the parent identity $\boldsymbol{id}_{\ell-1} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_{\ell-1}\}$ at depth $\ell - 1 > 0$ and outputs the private key $\mathsf{Sk}_{\boldsymbol{id}}$ for identity $\boldsymbol{id}$. In such an HIBE scheme, identities are vectors.

For an (H)IBE system described above, the correctness is that: for any message $\boldsymbol{m} \in \mathcal{M}$, $\boldsymbol{id}$ and (mpk, msk) generated by $\mathsf{KeyGen}(1^n)$, $\boldsymbol{c}$ is the ciphertext output by the $\mathsf{Enc}(mpk, \boldsymbol{id}, \boldsymbol{m})$ algorithm, then the $\mathsf{Dec}(\mathsf{SK}_{\boldsymbol{id}}, \boldsymbol{id}, \boldsymbol{c})$ will output $\boldsymbol{m}$ with overwhelming probability.

2.2 Security Definition

HIBE Security. For a HIBE system, besides the requirement of correctness, it also needs to achieve other security requirements. In the following, we will simply define selective security and adaptive security for a HIBE system. Let $\mathcal{A}$ be any non-uniform probability polynomial time adversary, the security experiment of selective security (INDr-sID-CPA) is defined as follows:

- Init: The adversary $\mathcal{A}$ is given the maximum hierarchy depth d and announces a target identity $\boldsymbol{id}^* = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_t\}$ of depth $t < d$.
- KeyGen: The simulator $\mathcal{S}$ generates the KeyGen algorithm to generate the public parameter mpk and master key msk and sends mpk to adversary $\mathcal{A}$.
- Query1: The adversary $\mathcal{A}$ makes queries on identity $\boldsymbol{id}_1, ..., \boldsymbol{id}_k$, where no one is a prefix of $\boldsymbol{id}^*$. The simulator returns the private key $\mathsf{Sk}_{\boldsymbol{id}_i}$ responding to each query on identity $\boldsymbol{id}_i$ by calling the Extract algorithm.
- Challenge Ciphertext: When the phase of Query1 is over and the adversary $\mathcal{A}$ sends a challenge message $\boldsymbol{m} \in \mathcal{M}$ to $\mathcal{S}$. The simulator $\mathcal{S}$ chooses a random bit $b \in \{0, 1\}$ and a random $\boldsymbol{c}'$ from the ciphertext space. If $b = 0$, then $\mathcal{S}$ generates the challenge ciphertext $\boldsymbol{c}^*$ by calling $\mathsf{Enc}(mpk, \boldsymbol{id}^*, \boldsymbol{m})$ with message $\boldsymbol{m}$; Otherwise, $\mathcal{S}$ sends $\boldsymbol{c}'$ as the challenge ciphertext $\boldsymbol{c}^*$ to $\mathcal{A}$.
- Query2: The adversary makes additional adaptive private key queries as in the phase of Query1 and the simulator proceeds as before.
- Guess: Finally, the adversary outputs a guess $b' \in \{0, 1\}$ and wins if $b' = b$.

Definition 1. *Let $\mathcal{A}$ be a PPT adversary in above INDr-sID-CPA experiment attacking the* HIBE *scheme, the advantage of adversary $\mathcal{A}$ is defined as*

$$\mathbf{Adv}_{\mathrm{HIBE},\mathcal{A}}^{indr\text{-}sid\text{-}cpa} \triangleq \left|\mathbf{Pr}[b' = b] - \frac{1}{2}\right|$$

We say an HIBE *scheme of depth d is selective secure if for any INDr-sID-CPA adversaries $\mathcal{A}$ there is*

$$\mathbf{Adv}_{\mathrm{HIBE},\mathcal{A}}^{indr\text{-}sid\text{-}cpa} \leq negl(n)$$

2.3 The Gadget Matrix G

In this section, we will recall a parity-check matrix $\mathbf{G}$ used in [10], where $\mathbf{G}$ is defined as:

$$\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^t \in \mathbb{Z}^{n \times nk}, k = \lceil \log_b q \rceil$$

where $\mathbf{g}^t = (1, b, b^2, ..., b^{k-1}) \in \mathbb{Z}^k$ is a special vector, $\mathbf{I}_n \in \mathbb{Z}^{n \times n}$ is the identity matrix and $\otimes$ denotes the tensor product.

For the lattice $\Lambda^{\perp}(\mathbf{g}^t)$, we have a good basis $\mathbf{T}$ as follows, and then $\mathbf{S} = \mathbf{I}_n \otimes \mathbf{T} \in \mathbb{Z}^{nk \times nk}$ is the basis of $\Lambda^{\perp}(\mathbf{G})$, that is,

$$\mathbf{T} := \begin{bmatrix} b & & & & q_0 \\ -1 & b & & & q_1 \\ & & \ddots & & \vdots \\ & & & b & q_{k-2} \\ & & & -1 & q_{k-1} \end{bmatrix} \in \mathbb{Z}^{k \times k}, \mathbf{S} := \mathbf{I} \otimes \mathbf{T} = \begin{bmatrix} \mathbf{T} & & & & \\ & \mathbf{T} & & & \\ & & \ddots & & \\ & & & \mathbf{T} & \\ & & & & \mathbf{T} \end{bmatrix} \in \mathbb{Z}^{nk \times nk}$$

where $q_0, ..., q_{k-1} \in [0, b)^k$ is decomposition of $q = \Sigma_i (b^i \cdot q_i)$ with base b.

There are some properties of this gadget matrix $\mathbf{G}$ proposed in [10]:

- **Short Basis:** $\mathbf{S}$ is the basis of lattice $\Lambda^{\perp}(\mathbf{G})$ with $\|\tilde{\mathbf{S}}\| \leq \sqrt{b^2+1}$.
- **Inverting simply:** The function $\mathbf{g}_{\mathbf{G}}(\boldsymbol{s}, \boldsymbol{e}) = \mathbf{G}^t \boldsymbol{s} + \boldsymbol{e}$ mod q can be inverted in quasi-linear time $O(n \cdot log^c n)$ for any $\boldsymbol{s} \in \mathbb{Z}_q^n$ and $\boldsymbol{e} \leftarrow \chi^m$ such that $\boldsymbol{e} \in \mathcal{P}_{1/2}(q \cdot \mathbf{S}^{-t})$ or $\boldsymbol{e} \in \mathcal{P}_{1/2}(q \cdot \tilde{\mathbf{S}}^{-t})$.

2.4 Some Algorithms

In this subsection, we will recall some algorithms: the trapdoor generation algorithm GenTrap in [10], the preimage sampling algorithms SamplePre with a short basis in [6] and SampleD with a trapdoor $\mathbf{R}$ in [10] and the extensions of preimage sampling algorithms SampleLeft and SampleRight in [1]. The concrete algorithms were described as follows.

Lemma 1 ([10]). *Let $n, q > 2$, $m = O(n \log q)$ be integers, then there exists a polynomial time algorithm* GenTrap(n, m, q) *outputs a vector $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a matrix $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$, where $\mathbf{T_A}$ is a basis for $\Lambda^{\perp}(\mathbf{A})$ such that $\mathbf{A}$ is statistically close to uniform and $\|\widetilde{\mathbf{T_A}}\| = O(\sqrt{n \log q})$.*

Lemma 2 ([6]). *Let n, $q > 2$, $w > n$, $m = O(n \log q)$ be integers. Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a matrix $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ is a basis for $\Lambda^{\perp}(\mathbf{A})$, a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$ and a Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m+w)})$, then there exists a polynomial time algorithm* SamplePre$(\mathbf{A}, \boldsymbol{u}, \mathbf{T_A}, \sigma)$ *outputs a vector $\boldsymbol{e} \in \mathbb{Z}^m$ sampled from a distribution which is statistically close to $D_{\Lambda^{\perp}(\mathbf{A}), \sigma, \boldsymbol{u}}$ and satisfies $\mathbf{A} * \boldsymbol{e} = \boldsymbol{u}$.*

Lemma 3 ([10]). *Let $n, q, w > n$ be integers, $m = O(n \log q)$ and $k = \lceil \log_b q \rceil$ for $2 < b < q$. Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \in \mathbb{Z}^{m \times \ell}$, $\mathbf{A}' = \mathbf{AR} + \mathsf{H}\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ and a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, a matrix $\mathbf{S} \in \mathbb{Z}^{nk \times nk}$ such that $\mathbf{S}$ is a basis for $\Lambda^{\perp}(\mathbf{G})$ and a*

Gaussian parameter $\sigma \geq \sqrt{s_1(\mathbf{R})^2+1} \cdot \|\widetilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log(m+nk)})$*, then there exists a polynomial time algorithm* $\mathsf{SampleD}(\mathbf{A}, \mathbf{R}, \mathbf{A}', \boldsymbol{u}, \mathbf{S}, \sigma)$ *outputs a vector* $\boldsymbol{e} \in \mathbb{Z}^{m+kn}$ *sampled from a distribution which is statistically close to* $D_{\Lambda^{\perp}([\mathbf{A}|\mathbf{A}']),\sigma,\boldsymbol{u}}$ *and satisfies* $[\mathbf{A}|\mathbf{A}'] * \boldsymbol{e} = \boldsymbol{u}$*.*

Lemma 4 **([1,5]).** *Let* n*,* $q > 2$*,* $w > n$*,* $m = O(n \log q)$ *be integers. Given* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $\mathbf{A}' \in \mathbb{Z}_q^{n \times w}$ *and a vector* $\boldsymbol{u} \in \mathbb{Z}_q^n$*, a matrix* $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ *is a basis for* $\Lambda^{\perp}(\mathbf{A})$ *and a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m+w)})$*, then there exists a polynomial time algorithm* $\mathsf{SampleLeft}(\mathbf{A}, \mathbf{A}', \boldsymbol{u}, \mathbf{T_A}, \sigma)$ *outputs a vector* $\boldsymbol{e} \in \mathbb{Z}^{m+w}$ *sampled from a distribution which is statistically close to* $D_{\Lambda^{\perp}([\mathbf{A}|\mathbf{A}']),\sigma,\boldsymbol{u}}$ *and satisfies* $[\mathbf{A}|\mathbf{A}'] * \boldsymbol{e} = \boldsymbol{u}$*.*

Lemma 5 **([1,9]).** *Let* n*,* $q > 2$*,* $w > n$*,* $m = O(n \log q)$ *be integers. Given* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $\mathbf{R} \in \mathbb{Z}^{m \times w}$*,* $\mathbf{B} \in \mathbb{Z}_q^{n \times w}$*,* $\mathbf{A}' = \mathbf{AR} + \mathbf{B} \in \mathbb{Z}_q^{n \times w}$*, a vector* $\boldsymbol{u} \in \mathbb{Z}_q^n$ *and a matrix* $\mathbf{T_B} \in \mathbb{Z}^{w \times w}$ *is a basis for* $\Lambda^{\perp}(\mathbf{B})$ *and a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{T_B}}\| \cdot s_1(\mathbf{R}) \cdot \omega(\sqrt{\log w})$*, then there exists a polynomial time algorithm* $\mathsf{SampleRight}(\mathbf{A}, \mathbf{R}, \mathbf{A}', \boldsymbol{u}, \mathbf{T_B}, \sigma)$ *outputs a vector* $\boldsymbol{e} \in \mathbb{Z}^{m+w}$ *sampled from a distribution which is statistically close to* $D_{\Lambda^{\perp}([\mathbf{A}|\mathbf{A}']),\sigma,\boldsymbol{u}}$ *and satisfies* $[\mathbf{A}|\mathbf{A}'] * \boldsymbol{e} = \boldsymbol{u}$*.*

2.5 Trapdoor Delegation Algorithms

In this subsection, we will recall several trapdoor delegation algorithms. The SampleBasisLeft and SampleBasisRight algorithms were extensions of SampleLeft and SampleRight in [1]. Micciancio and Peikert [10] introduced another trapdoor delegation algorithm DelTrap with a trapdoor $\mathbf{R}$. The concrete algorithms are described as follows.

Lemma 6 **([1]).** *Let* n*,* $q > 2$*,* $w > n$*,* $m = O(n \log q)$ *be integers. Given* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $\mathbf{A}' \in \mathbb{Z}_q^{n \times w}$*, a matrix* $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ *is a basis for* $\Lambda^{\perp}(\mathbf{A})$ *and a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m+w)})$*, then there exists a polynomial time algorithm* $\mathsf{SampleBasisLeft}(\mathbf{A}, \mathbf{A}', \mathbf{T_A}, \sigma)$ *outputs a basis* $\mathbf{T}$ *for lattice* $\Lambda^{\perp}([\mathbf{A}|\mathbf{A}'])$ *and satisfies* $\|\widetilde{\mathbf{T}}\| \leq \sigma\sqrt{(m+w)}$*.*

Lemma 7 **([1]).** *Let* n*,* $q > 2$*,* $m = O(n \log q)$ *be integers. Given* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and the identity-based encryption matrix for* $\boldsymbol{id} = (\boldsymbol{id}_1, \cdots, \boldsymbol{id}_\ell)$ *is*

$$\mathbf{F}_{\boldsymbol{id}} = [\mathbf{A}|\mathbf{A}\mathsf{R}_1 + \mathsf{H}(\boldsymbol{id}_1)\mathbf{B}|\cdots|\mathbf{A}\mathsf{R}_\ell + \mathsf{H}(\boldsymbol{id}_\ell)\mathbf{B}] \in \mathbb{Z}_q^{n \times (\ell+1)m}$$

Let $\mathbf{R}_\ell = (\mathsf{R}_1|\cdots|\mathsf{R}_\ell)$ *and* $h_{\boldsymbol{id}} = [\mathsf{H}(\boldsymbol{id}_1 - \boldsymbol{id}_1^*)\mathbf{B}|\cdots|\mathsf{H}(\boldsymbol{id}_\ell - \boldsymbol{id}_\ell^*)\mathbf{B}] \in \mathbb{Z}_q^{n \times \ell m}$*. The matrix* $\mathbf{T_B} \in \mathbb{Z}^{m \times m}$ *is a basis for* $\Lambda^{\perp}(\mathbf{B})$ *and a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{T_B}}\| \cdot s_1(\mathbf{R}_\ell) \cdot \omega(\sqrt{\log m})$*, then there exists a polynomial time algorithm* $\mathsf{SampleBasisRight}(\mathbf{A}, \mathbf{R}_\ell, \mathbf{F}_{\boldsymbol{id}}, \mathbf{T_B}, \sigma)$ *outputs a basis* $\mathbf{T}$ *for lattice* $\Lambda^{\perp}(\mathbf{F}_{\boldsymbol{id}})$ *and satisfies* $\|\widetilde{\mathbf{T}}\| \leq \sigma\sqrt{(\ell+1)m}$*.*

In our work, we will use the gadget matrix $\mathbf{G}'$ instead of the matrix $\mathbf{B}$ in the SampleBasisRight algorithm and use the algorithm SampleD instead of SampleRight in the SampleBasisRight algorithm. Because the $\mathbf{G}'$ is rank n and $\mathbf{S}'$ is a short basis for $\Lambda^{\perp}(\mathbf{G}')$, we can obtain the following corollary.

Corollary 1. *Let n, $q > 2$, $w > n$, $m = O(n \log q)$ be integers. Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and the identity-based encryption matrix for $\boldsymbol{id} = (\boldsymbol{id}_1, \cdots, \boldsymbol{id}_\ell)$ is*

$$\mathbf{F}_{\boldsymbol{id}} = [\mathbf{A}|\mathbf{A}\mathsf{R}_1 + \mathsf{H}(\boldsymbol{id}_1 - \boldsymbol{id}_1^*)\mathbf{G}'|\cdots|\mathbf{A}\mathsf{R}_\ell + \mathsf{H}(\boldsymbol{id}_\ell - \boldsymbol{id}_\ell^*)\mathbf{G}'] \in \mathbb{Z}_q^{m+\ell nk}$$

Let $\mathbf{R}_\ell = (\mathsf{R}_1|\cdots|\mathsf{R}_\ell)$ and $h_{\boldsymbol{id}} = [\mathsf{H}(\boldsymbol{id}_1 - \boldsymbol{id}_1^*)\mathbf{G}'|\cdots|\mathsf{H}(\boldsymbol{id}_\ell - \boldsymbol{id}_\ell^*)\mathbf{G}'] \in \mathbb{Z}_q^{n\times \ell nk}$. The matrix $\mathbf{S}' \in \mathbb{Z}^{nk\times nk}$ is a basis for $\Lambda^\perp(\mathbf{G}')$ and a Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{S}'}\| \cdot s_1(\mathbf{R}_\ell) \cdot \omega(\sqrt{\log nk})$, then there exists a polynomial time algorithm $\mathsf{SampleBasisRight}(\mathbf{A}, \mathbf{R}_\ell, \mathbf{F}_{\boldsymbol{id}}, \mathbf{S}', \sigma)$ outputs a basis $\mathbf{T}$ for lattice $\Lambda^\perp([\mathbf{A}|\mathbf{A}'])$ and satisfies $\|\widetilde{\mathbf{T}}\| \leq \sigma\sqrt{m + \ell nk}$.

Proof. When we replace the matrix $\mathbf{B}$ with the gadget matrix $\mathbf{G}'$ and set the Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{S}'}\| \cdot s_1(\mathbf{R}_\ell) \cdot \omega(\sqrt{\log nk})$, then output of the algorithms $\mathsf{SampleRight}$ and $\mathsf{SampleD}$ are the same. Therefore, the $\mathsf{SampleBasisRight}$ algorithm will output the short basis for $\mathbf{F}_{\boldsymbol{id}}$. □

Lemma 8 ([10])**.** *Let n, $q > 2$, $m = O(n \log q)$ be integers. Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{R} \in \mathbb{Z}^{m\times nk}$ is a $\mathbf{G}$-trapdoor for $\mathbf{A}$, let $\mathbf{A}' = [\mathbf{A}|\mathbf{A}_1] \in \mathbb{Z}_q^{n\times(m+k')}$, a tag $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ and a Gaussian parameter $\sigma \geq \sqrt{s_1(\mathbf{R})^2 + 1} \cdot s_1(\sqrt{\Sigma_{\mathbf{G}}})$, then there exists a polynomial time algorithm $\mathsf{DelTrap}(\mathbf{A}, \mathbf{R}, \mathbf{A}', \sigma)$ outputs a trapdoor $\mathbf{R}' \in \mathbb{Z}_q^{m\times k'}$ for $\mathbf{A}'$ with tag $\mathbf{H}'$ such that $\mathbf{A}\mathbf{R}' = \mathbf{H}'\mathbf{G}' - \mathbf{A}_1$ and satisfies $s_1(\mathbf{R}') \leq \sigma \cdot O(\sqrt{m} + \sqrt{k'})$, where $\mathbf{G}'$ is set by base b and $k' = \lceil \log_b q \rceil$.*

3 Hierarchical IBE with Compact Ciphertext from LWE

In this section, we will introduce our HIBE scheme based on the LWE problem. In the construction, we will utilize the function defined in [1,10] that encodes the identities into matrices and satisfies the "unit differences" property: for any $\boldsymbol{id}_i \neq \boldsymbol{id}_j$, $\mathsf{H}(\boldsymbol{id}_i) - \mathsf{H}(\boldsymbol{id}_j) = \mathsf{H}(\boldsymbol{id}_i - \boldsymbol{id}_j)$ is invertible and $\mathsf{H}(\mathbf{0}) = \mathbf{0}$. Moreover, the parity-check matrix $\mathbf{G}'$ in our construction is defined as $\mathbf{G}' = \mathbf{g}^t \otimes \mathbf{I}_n \in \mathbb{Z}^{n\times nk}$, where $\mathbf{g}^t = (1, b, b^2, ..., b^{k-1}) \in \mathbb{Z}^k$, $b = 2^d$ and $k = \lceil \log_b q \rceil$. And $\mathbf{S}'$ is a short basis of lattice $\Lambda^\perp(\mathbf{G}')$ with $\|\tilde{\mathbf{S}}'\| \leq \sqrt{b^2 + 1}$.

3.1 The HIBE Construction

- $\mathsf{KeyGen}(1^n, q) \to (pk, sk)$: The algorithm calls $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \xleftarrow{\$} \mathsf{GenTrap}(n, m, q)$ to generate $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a matrix $\mathbf{T}_\mathbf{A}$ is a short basis for lattice $\Lambda^\perp(\mathbf{A})$, then randomly samples d matrices $\mathbf{A}_1, \cdots, \mathbf{A}_d \xleftarrow{\$} \mathbb{Z}_q^{n\times nk}$ and a vector $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_q^n$. Therefore, the master public key mpk and the master secret key msk are

$$mpk = (\mathbf{A}, \mathbf{A}_1, \cdots, \mathbf{A}_d, \boldsymbol{u}) \in \mathbb{Z}_q^{n\times(m+dnk+1)}; \ msk = \mathbf{T}_\mathbf{A}.$$

- $\mathsf{Derive}(mpk, \boldsymbol{id}|\boldsymbol{id}_\ell, \mathsf{SK}_{\boldsymbol{id}}) \to \mathsf{SK}_{\boldsymbol{id}|\boldsymbol{id}_\ell}$: Given the master public key mpk, a private key $\mathsf{SK}_{\boldsymbol{id}}$ for identity $\boldsymbol{id} = \{\boldsymbol{id}_1, \cdots, \boldsymbol{id}_{\ell-1}\}$ at depth $\ell - 1$ as inputs, the algorithm works as follows:

1. Let $\mathbf{A}_{\boldsymbol{id}} = [\mathbf{A}_1 + \mathsf{H}(\boldsymbol{id}_1)\mathbf{G}'|\cdots|\mathbf{A}_{\ell-1} + \mathsf{H}(\boldsymbol{id}_{\ell-1})\mathbf{G}']$ and $\mathbf{F}_{\boldsymbol{id}} = [\mathbf{A}|\mathbf{A}_{\boldsymbol{id}}]$, then $\mathsf{SK}_{\boldsymbol{id}}$ is a short basis for lattice $\Lambda_q^{\perp}(\mathbf{F}_{\boldsymbol{id}})$;
2. Let $\mathbf{F}_{\boldsymbol{id}|\boldsymbol{id}_\ell} = [\mathbf{F}_{\boldsymbol{id}}|\mathbf{A}_\ell + \mathsf{H}(\boldsymbol{id}_\ell)\mathbf{G}']$;
3. Construct short basis for lattice $\Lambda_q^{\perp}(\mathbf{F}_{\boldsymbol{id}|\boldsymbol{id}_\ell})$ by invoking

$$\mathbf{S} \leftarrow \mathsf{SampleBasisLeft}(\mathbf{A}_{\boldsymbol{id}}, \mathbf{A}_\ell + \mathsf{H}(\boldsymbol{id}_\ell)\mathbf{G}', \mathsf{SK}_{\boldsymbol{id}}, \sigma_\ell)$$

4. Output $\mathsf{SK}_{\boldsymbol{id}|\boldsymbol{id}_\ell} = \mathbf{S}$ as the private key for identity $\boldsymbol{id}|\boldsymbol{id}_\ell$.

- $\mathsf{Encrypt}(mpk, \boldsymbol{id}, \boldsymbol{m}) \to \boldsymbol{c}$: Given the master public key mpk, the identity $\boldsymbol{id}$ and message $\boldsymbol{m} \in \{0,1\}$ as inputs, the algorithm works as follows:
  1. For identity $\boldsymbol{id} = \{\boldsymbol{id}_1, \cdots, \boldsymbol{id}_\ell\}$, compute

$$\mathbf{A}_{\boldsymbol{id}} = [\mathbf{A}_1 + \mathsf{H}(\boldsymbol{id}_1)\mathbf{G}'|\cdots|\mathbf{A}_\ell + \mathsf{H}(\boldsymbol{id}_\ell)\mathbf{G}'] \in \mathbb{Z}_q^{n\times \ell nk}$$

  and $\mathbf{F}_{\boldsymbol{id}} = [\mathbf{A}|\mathbf{A}_{\boldsymbol{id}}] \in \mathbb{Z}_q^{n\times(m+\ell nk)}$;
  2. Choose a uniformly randomness $\boldsymbol{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$;
  3. Choose a uniformly random matrix $\mathsf{R} \overset{\$}{\leftarrow} \{-1,1\}^{m\times \ell nk}$;
  4. Choose $(x_0, \boldsymbol{x}_1) \leftarrow D_{\mathbb{Z},\alpha_\ell q} \times D^m_{\mathbb{Z},\alpha_\ell q}$, then set $\boldsymbol{x}_2^t = \boldsymbol{x}_1^t \mathsf{R} \in \mathbb{Z}_q^{\ell nk}$ and compute

$$\begin{cases} c_0 = \boldsymbol{s}^t\boldsymbol{u} + x_0 + \lfloor q/2 \rfloor \cdot \boldsymbol{m} \\ \boldsymbol{c}_1 = \boldsymbol{s}^t\mathbf{F}_{\boldsymbol{id}} + [\boldsymbol{x}_1^t|\boldsymbol{x}_2^t] = \boldsymbol{s}^t[\mathbf{A}|\mathbf{A}_{\boldsymbol{id}}] + [\boldsymbol{x}_1^t|\boldsymbol{x}_2^t] \end{cases}$$

  5. Output the ciphertext$\boldsymbol{c}^t = (c_0, \boldsymbol{c}_1^t) \in \mathbb{Z}_q \times \mathbb{Z}_q^{m+\ell nk}$.
- $\mathsf{Decrypt}(\boldsymbol{c}, \boldsymbol{id}, \mathsf{SK}_{\boldsymbol{id}}) \to \boldsymbol{m}$ or $\perp$: Given the ciphertext $\boldsymbol{c}$ and the private key $\mathsf{SK}_{\boldsymbol{id}}$ for identity $\boldsymbol{id}$ as input, the algorithm works as follows:
  1. Parse $\boldsymbol{c}^t = (c_0, \boldsymbol{c}_1^t)$, if $\boldsymbol{c}$ cannot parse in this way, output $\perp$;
  2. Compute $\mathbf{A}_{\boldsymbol{id}}$, $\mathbf{F}_{\boldsymbol{id}}$ as before;
  3. Let $\tau_\ell = \sigma_\ell \cdot \sqrt{m + \ell nk} \cdot \omega(\sqrt{\log(m+\ell nk)}) \geq \|\widetilde{\mathsf{SK}_{\boldsymbol{id}}}\| \cdot \omega(\sqrt{\log(m+\ell nk)})$ and sample $\boldsymbol{e}_{\boldsymbol{id}} \in \mathbb{Z}^{m+\ell nk}$ as

$$\boldsymbol{e}_{\boldsymbol{id}} \leftarrow \mathsf{SamplePre}(\mathbf{F}_{\boldsymbol{id}}, \mathsf{SK}_{\boldsymbol{id}}, \tau_\ell, \boldsymbol{u})$$

  s.t. $\mathbf{F}_{\boldsymbol{id}}\boldsymbol{e}_{\boldsymbol{id}} = \boldsymbol{u}$;
  4. Compute $w = c_0 - \boldsymbol{c}_1\boldsymbol{e}_{\boldsymbol{id}}$;
  5. Compute and output the message $\boldsymbol{m} = \lfloor \frac{w}{q/2} \rceil \pmod 2$.

3.2 Parameters and Correctness

In this subsection, we will describe the requirement of parameters which meets the correctness and security of the above HIBE scheme, then we will propose a set of parameters.

Lemma 9 (Correctness). *Assume the parameters $n, m, q, \ell, \alpha_\ell, \sigma_\ell, \tau_\ell$ satisfy the condition $\alpha_\ell q\omega(\sqrt{\log n}) + \alpha_\ell q\sigma_\ell(m+\ell nk)^{3/2} \cdot \omega(\log(m+\ell nk)) \leq q/5$ with all but negligible probability, then the* Dec *algorithm of the above HIBE will have negligible decryption error.*

Proof. For a valid ciphertext $\boldsymbol{c}$ of message $\boldsymbol{m}$, the Dec algorithm computes

$$c_0 - \boldsymbol{c}_1^t \boldsymbol{e_{id}} = \lfloor \frac{q}{2} \rfloor \boldsymbol{m} + x_0 - [\boldsymbol{x}_1^t | \boldsymbol{x}_2^t] \boldsymbol{e_{id}}.$$

Thus, on the condition that $\|x_0 - [\boldsymbol{x}_1^t|\boldsymbol{x}_2^t]\boldsymbol{e_{id}}\| \leq q/5$, the Dec algorithm can recover the message $\boldsymbol{m}$ correctly. Since $x_0 \leftarrow D_{\mathbb{Z},\alpha_\ell q}$, we have $\|x_0\| \leq \alpha_\ell q \omega(\sqrt{\log n})$ with all but negligible probability. Because $\mathsf{R} \xleftarrow{\$} \{-1,1\}^{m\times \ell nk}$, then we have $\|\mathsf{R}\| \leq O(\sqrt{m+\ell nk})$. Since $\boldsymbol{e}_{\boldsymbol{id}}^t = (\boldsymbol{e}_1^t, \boldsymbol{e}_2^t)$ is sampled by $\mathsf{SamplePre}(\mathbf{F}_{\boldsymbol{id}}, \mathsf{SK}_{\boldsymbol{id}}, \tau_\ell, \boldsymbol{u})$, let $\tau_\ell = \sigma_\ell \cdot \sqrt{m+\ell nk} \cdot \omega(\sqrt{\log(m+\ell nk)}) \geq \|\widetilde{\mathsf{SK}_{\boldsymbol{id}}}\| \cdot \omega(\sqrt{\log(m+\ell nk)})$, we have $\|\boldsymbol{e_{id}}\| \leq \tau_\ell \cdot \sqrt{m+\ell nk}$ with overwhelming probability. In addition, we have $\|\boldsymbol{e}_1 + \mathsf{R}\cdot\boldsymbol{e}_2\| \leq \|\boldsymbol{e}_1\| + \|\mathsf{R}\cdot\boldsymbol{e}_2\| \leq \sigma_\ell(m+\ell nk)^{3/2}\cdot\omega(\sqrt{\log(m+\ell nk)})$. Since $\boldsymbol{x}_1 \leftarrow D^m_{\mathbb{Z},\alpha_\ell q}$, then we have $\|\boldsymbol{x}_1^t(\boldsymbol{e}_1+\mathsf{R}\cdot\boldsymbol{e}_2)\| \leq \|\boldsymbol{e}_1 + \mathsf{R}\cdot\boldsymbol{e}_2\| \cdot \alpha_\ell q\omega(\sqrt{\log m}) \leq \alpha_\ell q \sigma_\ell (m+\ell nk)^{3/2} \cdot \omega(\log(m+\ell nk))$. Therefore, the error item during the process of decryption is bounded by

$$\|x_0 - [\boldsymbol{x}_1^t|\boldsymbol{x}_2^t]\boldsymbol{e_{id}}\| \leq \alpha_\ell q\omega(\sqrt{\log n}) + \alpha_\ell q\sigma_\ell(m+\ell nk)^{3/2}\cdot\omega(\log(m+\ell nk))$$

with all but negligible probability. Under the assumption in the lemma, the Dec algorithm of the above HIBE will have negligible decryption error. That completes the proof. □

To satisfy the requirement of correctness and security, taking n as security parameter, we set the parameters as

$$\begin{array}{ll} m = O(n\log q) = O(dn\log n) & , \ q = \tilde{O}((4d)^{d/2} n^{d/2}) \\ \sigma_\ell = 2^d (m+\ell nk)^{\frac{\ell}{2}} \omega(\log^{\frac{\ell}{2}}(m+\ell nk)) & , \ 1/\alpha_\ell = \sigma_\ell (m+\ell nk)^{3/2} \omega(\log(m+\ell nk)) \end{array}$$

The parameters	The sizes (bits)
mpk	$(m+dnk+1)n\log q = \tilde{O}(d^2n^2)$
$\mathsf{SK}_{\boldsymbol{id}}$ at level ℓ	$(m+\ell nk)^2\log(\sigma_\ell\sqrt{m+\ell nk}) = \tilde{O}((\frac{d}{\log n}+\ell)\cdot(d+\ell)^2n^2)$
ct at level ℓ	$(m+\ell nk)\log q = \tilde{O}(nd(d+\ell))$
Error rate at level ℓ	$\tilde{O}(2^d d^{(\ell+3)/2} n^{(\ell+3)/2})$

4 Conclusion

In this paper, we introduce a trade off of the sizes of public parameter and ciphertext and the size of private key in the selective-secure LWE-based HIBE scheme in the standard model. We obtain this trade-off by adjusting the base of the gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times n\lceil\log_b q\rceil}$ defined in [10]. By setting $b = 2^d$, the size of the master public key and ciphertext at level ℓ can de reduced by a factor of $O(d)$ and $O(\ell)$ respectively, at the cost of increasing the size of private key by a factor of $O(\frac{d}{\ell\log n}+1)$. And the parameters in ABB10b scheme except the private key is competitive with our HIBE scheme only when $\lambda = 1$.

Acknowledgments. We thank the anonymous ICICS'2017 reviewers for their helpful comments. This work is supported by the National Cryptography Development Fund MMJJ20170116 and the National Nature Science Foundation of China (No. 61602473, No. 61502480, No. 61672019, No. 61772522, No. 61379137, No. 61572495) and National Basic Research Programm of China (973 project, No. 2014CB340603).

References

1. 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). https://doi.org/10.1007/978-3-642-13190-5_28
2. 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). https://doi.org/10.1007/978-3-642-14623-7_6
3. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_42
4. Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 209–224. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_9
5. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
6. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197–206 (2008)
7. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_34
8. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_31
9. Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: more compact IBEs from ideal lattices and bilinear maps. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 682–712. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_23
10. 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). https://doi.org/10.1007/978-3-642-29011-4_41
11. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93 (2005)
12. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5