6-2016

(Deterministic) hierarchical identity-based encryption from learning with rounding over small modulus

Fuyang FANG

Bao LI

Xianhui LU

Yamin LIU

Dingding JIA

See next page for additional authors

Citation

FANG, Fuyang; LI, Bao; LU, Xianhui; LIU, Yamin; JIA, Dingding; and XUE, Haiyang. (Deterministic) hierarchical identity-based encryption from learning with rounding over small modulus. (2016). *ASIA CCS '16: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Xi'an, China, May 30 - June 3*. 907-912.

Available at: https://ink.library.smu.edu.sg/sis_research/9181

Author
Fuyang FANG, Bao LI, Xianhui LU, Yamin LIU, Dingding JIA, and Haiyang XUE

(Deterministic) Hierarchical Identity-based Encryption from Learning with Rounding over Small Modulus

Fuyang Fang[1,2,3], Bao Li[1,2], Xianhui Lu[1,2], Yamin Liu[1,2]*, Dingding Jia[1,2], Haiyang Xue[1,2]

[1]State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing, China
[2]Data Assurance and Communication Security Research Center, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences, Beijing, China
{fyfang13,lb,xhlu,ymliu,ddjia,hyxue12}@is.ac.cn

ABSTRACT

In this paper, we propose a hierarchical identity-based encryption (HIBE) scheme in the random oracle (RO) model based on the learning with rounding (LWR) problem over small modulus q. Compared with the previous HIBE schemes based on the learning with errors (LWE) problem, the ciphertext expansion ratio of our scheme can be decreased to 1/2. Then, we utilize the HIBE scheme to construct a deterministic hierarchical identity-based encryption (D-HIBE) scheme based on the LWR problem over small modulus. Finally, with the technique of binary tree encryption (BTE) we can construct HIBE and D-HIBE schemes in the standard model based on the LWR problem over small modulus.

Keywords

(hierarchical) identity-based encryption; deterministic (hierarchical) identity-based encryption; learning with rounding

1. INTRODUCTION

Hierarchical identity based encryption (HIBE) is a kind of identity-based encryption (IBE) scheme where any user at each level has the ability to delegate private keys for its next level [9,10]. The constructions of HIBE schemes from lattice have been studied for several years [1,2,7] and all of these schemes are based on the learning with errors (LWE) problem [13]. In these LWE-based constructions, the ciphertext expansion ratios were more than $O(\log q)$ due to the error correction for recovering the messages. As a variant of the LWE problem, the learning with rounding (LWR) problem was proposed by Banerjee et al. in [4] and needs not to sample any additional error item. The authors of [4] proved that the hardness of the LWR problem can be reduced to the LWE problem when the modulus q is super-polynomial. Recently, Alwen et al. [3] and Bogdanov et al. [6] proved that the hardness of the LWR problem can be reduced to the LWE problem when the modulus q is polynomial. Thus, the first question is whether the ciphertext expansion ratio of the HIBE schemes can be reduced with the LWR problem over small modulus instead of the LWE problem.

Escala et al. [8] extended HIBE to the deterministic scenario and proposed the notion of deterministic hierarchical identity-based encryption (D-HIBE or HIB-DE). However, they only constructed a pairing-based D-HIBE scheme in [8]. So far, the only known D-IBE scheme in the standard model based on the LWR assumption was proposed by Xie et al. [14]. However, the security of their scheme required that the modulus q should be super-polynomial and they did not consider the D-HIBE schemes. Therefore, the second question is whether the LWR problem with small modulus can be used to construct D-HIBE schemes.

1.1 Our Contributions

Firstly, we construct an adaptive secure HIBE scheme based on the LWR problem in the random oracle model, where the ciphertext expansion ration in our schemes can be reduced to 1/2. Secondly, we utilize the above HIBE scheme to construct an adaptive secure D-HIBE scheme based on the LWR problem in the random oracle model. Finally, by using the technique of binary tree encryption (BTE) we remove the random oracles and construct HIBE and D-HIBE schemes with selective security based on the LWR problem in the standard model. In the following table 1, we describe the parameter settings of our D-HIBE schemes, with and without random oracles.

Scheme	Model	Secret Key	Public Key	Ciphertext
D-HIBE	RO	$\widetilde{O}(\ell n^2 d^2)$	$\widetilde{O}(\ell n^2 d^2)$	$m \log p$
D-HIBE	Standard	$\widetilde{O}(\ell n^2 d^2)$	$\widetilde{O}(\ell n^2 d^3)$	$m \log p$

In this table, $m = O(n \log q), p \geq m^{\frac{3}{2}d+1}\omega(\log^{2d} m)$ and $q \geq 2mpB$. Let d be the maximum hierarchy depth and ℓ be the depth of the identity in question.

Table 1: The parameter of our D-HIBE schemes

The security of all our schemes are based on the hardness of the LWR problem with small modulus [6].

TECHNIQUES: When constructing LWR-based (H)IBE schemes, we observe that the basis delegation techniques in [1,7] are not applicable to the LWR problem with small

*This author is the corresponding author.

ASIA CCS '16, May 30-June 03, 2016, Xi'an, China
© 2016 ACM. ISBN 978-1-4503-4233-9/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2897845.2897922

modulus. In the proof of security for the scheme in [14], the simulator set $\mathbf{F}_{\boldsymbol{id}^*} := [\mathbf{A}_0|\mathbf{A}_0\mathbf{R}]$. When constructing the challenge ciphertext, the simulator chose m samples $(\mathbf{A}_0, \boldsymbol{b} = \mathbf{A}_0^t\boldsymbol{m} + \boldsymbol{e})$ from LWE distribution and set

$$\boldsymbol{c}^* = \begin{bmatrix} \lfloor \mathbf{A}_0^t\boldsymbol{m} + \boldsymbol{e} \rfloor_p \\ \lfloor \mathbf{R}^t(\mathbf{A}_0^t\boldsymbol{m} + \boldsymbol{e}) \rfloor_p \end{bmatrix} = \begin{bmatrix} \lfloor \mathbf{A}_0^t\boldsymbol{m} \rfloor_p \\ \lfloor \mathbf{R}^t(\mathbf{A}_0^t\boldsymbol{m}) \rfloor_p \end{bmatrix} = \lfloor \mathbf{F}_{\boldsymbol{id}^*}^t\boldsymbol{m} \rfloor_p$$

We notice that the above reduction works when the modulus q is super-polynomial. However, with the polynomial modulus q we cannot give a similar reduction. On one hand, the simulator does not have $\lfloor \mathbf{R}^t(\mathbf{A}_0^t\boldsymbol{m} + \boldsymbol{e}) \rfloor_p = \lfloor \mathbf{R}^t(\mathbf{A}_0^t\boldsymbol{m}) \rfloor_p$ with high probability for polynomial modulus q; On the other hand, the simulator cannot construct the second item of challenge ciphertext $\lfloor (\mathbf{A_0R})^t\boldsymbol{m} \rfloor_p$ from $\lfloor \mathbf{A}_0^t\boldsymbol{m} \rfloor_p$ with the trapdoor $\mathbf{R}$. Furthermore, when constructing the HIBE and D-HIBE schemes the dimension of lattice can not increase along with the hierarchy depth since that the number of the samples of the LWR problem we used is given beforehand.

In [2], Agrawal, Bobeh and Boyen proposed a technique of delegating a short basis without increasing the dimension. We recall their technique briefly as follows: Let $\mathbf{B} = \{\boldsymbol{b}_1, .., \boldsymbol{b}_m\}$ be a short basis of lattice $\varLambda$ and $\mathbf{R}$ be a matrix sampling from a distribution $\mathcal{D}_{m\times m}$ on low norm matrices. Then they can obtain a short basis for lattice $\mathbf{R}\varLambda$ by randomizing the basis $\mathbf{RB} = \{\mathbf{R}\boldsymbol{b}_1, .., \mathbf{R}\boldsymbol{b}_m\}$. More importantly, given a lattice $\varLambda$ without a short basis, they designed another algorithm **SampleRwithBasis** that can sample a low norm matrix $\mathbf{R}$ and a short basis $\mathbf{T}$ for lattice $\mathbf{R}\varLambda$. In their paper they showed that the matrices $\mathbf{R}$, as private keys during the simulation, were statistically close to the distribution of private keys in the real HIBE schemes.

In our work, we show that the technique in [2] can be applied to construct our (H)IBE and D-HIBE schemes.The encryption matrix $\mathbf{F}_{\boldsymbol{id}}$ and the ciphertext $\boldsymbol{c}$ are

$$\mathbf{F}_{\boldsymbol{id}} := \mathbf{A}\mathbf{R}_{\boldsymbol{id}}^{-1}; \quad \boldsymbol{c} := \lfloor \mathbf{F}_{\boldsymbol{id}}^t\boldsymbol{m} \rfloor_p$$

where $\mathbf{A}$ is the public matrix with a short basis $\mathbf{T}_{\mathbf{A}}$. In the proof of security for our scheme, the simulator can set $\mathbf{A} := \mathbf{A}_0\mathbf{R}_{\boldsymbol{id}^*}$ where $\mathbf{A}_0$ is uniform over $\mathbb{Z}_q^{n\times m}$ and obtain that $\mathbf{F}_{\boldsymbol{id}^*} := \mathbf{A}_0\mathbf{R}_{\boldsymbol{id}^*}\mathbf{R}_{\boldsymbol{id}^*}^{-1}$. When constructing the challenge ciphertext, the simulator chooses m samples $(\mathbf{A}_0, \lfloor \mathbf{A}_0^t\boldsymbol{s} \rfloor_p)$ from the LWR distribution and sets

$$\boldsymbol{c}^* = \lfloor \mathbf{A}_0^t\boldsymbol{s} \rfloor_p = \lfloor (\mathbf{A}_0\mathbf{R}_{\boldsymbol{id}^*}\mathbf{R}_{\boldsymbol{id}^*}^{-1})^t\boldsymbol{s} \rfloor_p = \lfloor \mathbf{F}_{\boldsymbol{id}^*}^t\boldsymbol{s} \rfloor_p$$

which solves the problem in simulating the challenge ciphertext. Meanwhile, the dimension of lattice remains unchanged along with the hierarchy depth.

1.2 Related Work

In [5], Bellare et.al extended the notion of lossy trapdoor function (LDTF) [12] to identity-setting and introduced the notion of identity-based LTDF (IB-LTDF), which could be used to construct D-IBE schemes. With IB-LTDF they constructed a pairing-based D-IBE scheme with selective security. In PKC 2014, Escala et al. [8] introduced the notion of hierarchical identity-based trapdoor functions (HIB-TDFs), which was an extension of IB-LDTF [5]. With HIB-TDFs they could construct (D)-HIBE schemes and HIB hedged encryption schemes. They instantiated the HIB-TDFs with pairing and constructed a pairing-based D-HIBE scheme. However, they left possible constructions based on lattice as an open line for future work.

2. PRELIMINARIES

Let λ be the security parameter and we use $negl(\lambda)$ to denote an arbitrary negligible function $f(\lambda)$ such that $f(\lambda) = o(\lambda^{-c})$ for every fixed constant c. We say that a probability is *overwhelming* if it is $1 - negl(\lambda)$. Let $poly(\lambda)$ denotes an unspecified function $f(\lambda) = O(\lambda^c)$ for some constant c. We use $\widetilde{O}(\lambda)$ be a function $f(\lambda)$ if $f(\lambda) = O(\lambda \cdot log^c\lambda)$ for some fixed constant c. We denote by $a \xleftarrow{\$} \mathbb{Z}_q$ that a is randomly chosen from $\mathbb{Z}_q$. We use PPT denotes *probability polynomial-time*. We use A $\approx_c$ B denotes that A is computationally indistinguishable from B.

2.1 Hierarchical IBE and D-HIBE

A HIBE scheme of depth d with the message space $\mathcal{M}$ can be defined by a tuple of PPT algorithms (**KeyGen, Derive, Extract, Enc, Dec**) as below: The probabilistic algorithm **KeyGen** generates the public key PP and the master key msk. The **Derive** algorithm takes as input an identity $\boldsymbol{id} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_\ell\}$ at depth $\ell \leq d$ and the private key $\mathbf{SK}_{\boldsymbol{id}_{\ell-1}}$ 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 $\mathbf{Sk}_{\boldsymbol{id}}$ for identity $\boldsymbol{id}$. The **Extract** algorithm uses the msk to extract a private key $SK_{\boldsymbol{id}}$ corresponding to a given identity $\boldsymbol{id}$. Given a message $\boldsymbol{m} \in \mathcal{M}$ and an identity $\boldsymbol{id}$, the probabilistic algorithm **Enc** uses the PP to encrypt the $\boldsymbol{m}$ with respect to the identity $\boldsymbol{id}$ and outputs a ciphertext $\boldsymbol{c}$. 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 $\perp$. A D-HIBE [8] scheme is similar to the definition of HIBE, except that the **Enc** algorithm in a D-HIBE scheme is a deterministic algorithm.

For the HIBE or D-HIBE system described above, the correctness is that: for any message $\boldsymbol{m} \in \mathcal{M}$, $\boldsymbol{id}$ and (PP, msk) generated by **KeyGen**, $\boldsymbol{c}$ is the ciphertext output by the **Enc**$(PP, \boldsymbol{id}, \boldsymbol{m})$ algorithm, then the **Dec**$(SK_{\boldsymbol{id}}, \boldsymbol{id}, \boldsymbol{c})$ will output $\boldsymbol{m}$ with overwhelming probability.

The INDr-sID-CPA security for HIBE schemes and the PRIV1-IND-sID security [8] for D-HIBE schemes can be defined respectively as follows:

Definition 1. *Let $\mathcal{A}$ be a PPT adversary attacking the HIBE scheme, the advantage of adversary $\mathcal{A}$ is defined as* $\mathbf{Adv}_{\text{HIBE},\mathcal{A}}^{indr\text{-}sid\text{-}cpa} \triangleq$

$$\left| \mathbf{Pr} \begin{bmatrix} & (\boldsymbol{id}^*) \leftarrow \mathcal{A}, (PP, msk) \leftarrow Gen(1^n); \\ b = b' : & (\boldsymbol{m}_0, \boldsymbol{m}_1) \leftarrow \mathcal{A}^{\mathcal{O}}(PP); b \xleftarrow{\$} \{0,1\}; \\ & c^* \leftarrow Enc(PP, \boldsymbol{m}_b, \boldsymbol{id}^*); b' = \mathcal{A}^{\mathcal{O}}(c^*) \end{bmatrix} - \frac{1}{2} \right|$$

where $\mathcal{O}$ denotes that $\mathcal{A}$ can make query on identity $\mathbf{id}$ *which is not a prefix of* $\mathbf{id}^*$ *by calling* ***Extract*** *algorithm. We say a HIBE scheme of depth d is selective secure if for any INDr-sID-CPA adversaries $\mathcal{A}$ there is* $\mathbf{Adv}_{\text{(H)IBE},\mathcal{A}}^{indr\text{-}sid\text{-}cpa} \leq negl(\lambda)$.

Definition 2. *Let $\mathcal{A}$ be a PPT adversary attacking the D-HIBE scheme for any x-sources $\mathcal{M}$, the advantage of $\mathcal{A}$ is defined as* $\mathbf{Adv}_{\text{D-HIBE},\mathcal{A}}^{priv1\text{-}ind\text{-}sid} \triangleq$

$$\left| \mathbf{Pr} \begin{bmatrix} & (\boldsymbol{id}^*) \leftarrow \mathcal{A}, (PP, msk) \leftarrow Gen(1^n); \\ b = b' : & (\boldsymbol{m}_0, \boldsymbol{m}_1) \leftarrow \mathcal{A}^{\mathcal{O}}(PP, \mathcal{M}); b \xleftarrow{\$} \{0,1\}; \\ & c^* \leftarrow Enc(PP, \boldsymbol{m}_b, \boldsymbol{id}^*); b' = \mathcal{A}^{\mathcal{O}}(c^*) \end{bmatrix} - \frac{1}{2} \right|$$

where $\mathcal{O}$ denotes that $\mathcal{A}$ can make query on identity $\mathbf{id}$ *which is not a prefix of* $\mathbf{id}^*$ *by calling* ***Extract*** *algorith-*

m. We say a D-HIBE scheme of depth d is selective secure, if for any PRIV1-IND-sID PPT adversaries $\mathcal{A}$ there is $\mathbf{Adv}^{\textit{priv1-ind-sid}}_{\text{D-HIBE},\mathcal{A}} \leq negl(\lambda)$.

Recall that a random variable $\boldsymbol{S}$ over $\{0,1\}^n$ is called a $x = poly(\lambda)$-source if it has efficient entropy $\boldsymbol{H}_\infty(\boldsymbol{S}) \geq x$.

The adaptive identity security experiment is defined when $\mathcal{A}$ can issue private key queries before $\mathcal{A}$ announces a challenge identity $\boldsymbol{id}^*$ to the simulator $\mathcal{S}$. The restriction is that $\mathcal{A}$ cannot issue a private key query for an identity that is a prefix of $\boldsymbol{id}^*$. The notion of INDr-ID-CPA and PRIV1-IND-ID can be defined using the modified experiment as in above definition 1 and 2 respectively.

2.2 The Learning with Rounding Problem

The learning with rounding (LWR) problem was first proposed by Banerjee, Peikert and Rosen in [4] for constructing pseudorandom functions. Let n, m, $2 < p < q$ be some integers, for a secret $\boldsymbol{s} \in \mathbb{Z}_q^n$, we can define a distribution $\mathcal{D}_{\boldsymbol{s}} \triangleq \{(\boldsymbol{a}, \lfloor\langle \boldsymbol{a}, \boldsymbol{s}\rangle\rfloor_p) | \boldsymbol{a} \xleftarrow{\$} \mathbb{Z}_q^n\}$, where the rounding function $\lfloor\cdot\rfloor_p$: $\mathbb{Z}_q \mapsto \mathbb{Z}_p$ denotes $\lfloor x\rfloor_p = \lfloor (p/q)\cdot x\rfloor \bmod p$. The LWR problem ($\mathbf{LWR}_{m,n,q,p}$) is to distinguish $\mathcal{D}_{\boldsymbol{s}}$ from the uniform distribution $\mathcal{U}(\mathbb{Z}_q^n) \times \lfloor\mathcal{U}(\mathbb{Z}_q)\rfloor_p$, given m independent samples. If $p|q$, then $\lfloor\mathcal{U}(\mathbb{Z}_q)\rfloor_p$ is itself uniform over $\mathbb{Z}_p$.

For the hardness of the LWR problem, Banerjee et al. [4] gave a direct reduction from the LWE problem when the modulus q was super-polynomial. In [3], Alwen et al. gave a reduction that allowed for a polynomial modulus q. Recently, Bogdanov et al. [6] showed a new reduction that did not impose any number theoretic restriction on the modulus q and the theorem is as follows,

THEOREM 1 ([6]). *For every $\epsilon > 0$, n, m, $q > 2mpB$, and if there is an algorithm $\mathcal{B}$ such that*

$$|\mathrm{Pr}_{\mathbf{A},\boldsymbol{s}}[\mathcal{B}(\mathbf{A}, \lfloor\mathbf{A}^t\boldsymbol{s}\rfloor_p) = 1] - \mathrm{Pr}_{\mathbf{A},\mathbf{u}}[\mathcal{B}(\mathbf{A}, \lfloor\mathbf{u}\rfloor_p) = 1]| \geq \epsilon$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\boldsymbol{s} \leftarrow \{0,1\}^n$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, there exists an algorithm $\mathbf{L}$ that runs in time polynomial in n, m, the number of divisors of q, and the running time of $\mathcal{B}$ such that

$$|\mathrm{Pr}_{\mathbf{A},\boldsymbol{s}}[\mathbf{L}(\mathbf{A}, \mathbf{A}^t\boldsymbol{s} + \boldsymbol{e}) = \boldsymbol{s}] \geq (\frac{\epsilon}{4qm} - \frac{2^n}{p^m})^2 \cdot \frac{1}{(1+2Bp/q)^m}$$

for any error distribution $\boldsymbol{e}$ that is B-bounded and balanced in each coordinate. A distribution χ supported over the integers is called B-bounded if $\mathbf{Pr}_{e\leftarrow\chi}[\|e\| \geq B] \leq 2^{-\Omega(m)}$ and we can let $B > \sqrt{n}$.

Remark: 1. If q is a prime, the secret $\boldsymbol{s}$ can be chosen any distribution over $\mathbb{Z}_q^n$. In our work we set the distribution on secret $\boldsymbol{s}$ be any distribution which satisfies $\mathbf{H}_\infty(\boldsymbol{s}) \geq x \geq t\log q + \omega(\log n)$, assuming the $\mathrm{LWE}_{t,n,q}$ problem for x-source secret is hard; 2. If q is not prime, then we set the secret $\boldsymbol{s}$ uniformly chosen from $\mathbb{Z}_q^{n*}$, where $\mathbb{Z}_q^{n*} = \{\boldsymbol{s} \in \mathbb{Z}_q^n : \gcd(s_1, ..., s_n, q) = 1\}$. The condition $\boldsymbol{s} \in \mathbb{Z}_q^{n*}$ is satisfied for at least $1 - O(1/2^n)$ fraction of secret $\boldsymbol{s} \in \mathbb{Z}_q^n$.

2.3 Some Algorithms

In this section we describe some algorithms that will be used in our work.

LEMMA 1 ([11]). *Given any integer n, $q \geq 2$, $m \approx 2n\log q$, there exists an algorithm $\boldsymbol{GenTrap}(1^n, 1^m, q)$ that outputs a parity-check matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{X}$ with a tag $\boldsymbol{H}$ such that the distribution of $\mathbf{A}$ is statistically close to the uniform. With the trapdoor we can construct a short basis $\mathbf{T_A}$ for lattice $\Lambda_q^\perp(\mathbf{A})$ and the parameter satisfies*

$$s_1(\mathbf{X}) \leq 1.6\sqrt{n\log q} \text{ and } \|\widetilde{\mathbf{T_A}}\| \leq 3.8\sqrt{n\log q}$$

LEMMA 2 ([7]). *Given a basis $\mathbf{T_A}$ for lattice $\Lambda_q^\perp(\mathbf{A})$ and a parameter $\sigma \geq \|\widetilde{\mathbf{T_A}}\|\cdot\omega(\sqrt{\log m})$, there is an algorithm $\boldsymbol{RandBasis}(\mathbf{T_A}, \sigma)$ outputs a new basis $\mathbf{T'_A}$ of $\Lambda_q^\perp(\mathbf{A})$ with overwhelming probability such that $\|\widetilde{\mathbf{T'_A}}\|, \|\mathbf{T'_A}\| \leq \sigma\cdot\sqrt{m}$, where the distribution of $\mathbf{T'_A}$ does not depend on $\mathbf{T_A}$.*

Similar with [2] in the basis delegation mechanism, we require that low norm matrix $\mathbf{R}$ is invertible over $\mathbb{Z}_q$ where each column of $\mathbf{R}$ is low norm vector. In the following paper, we denote the $(\mathcal{D}_{\mathbb{Z}_q^m,\sigma_\mathbf{R}})^m$, shortly $\mathcal{D}_{m\times m}$, as the distribution of matrix $\mathbf{R}$, where $\sigma_\mathbf{R} = \sqrt{n\log q}\cdot\omega(\sqrt{\log m})$. Then we will recall two main algorithms [2] that were used to delegate the basis without increasing the dimension of lattice.

LEMMA 3 ([2]). *Let $m \geq 2n\log q$ and $q > 2$ be an integer. Given the matrix $\mathbf{R}$ sampled from distribution $\mathcal{D}_{m\times m}$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a basis $\mathbf{T_A}$ for lattice $\Lambda_q^\perp(\mathbf{A})$ and parameter $\sigma > \|\widetilde{\mathbf{T_A}}\|\cdot\sigma_\mathbf{R}\cdot\sqrt{m}\cdot\omega(\log^{3/2} m)$, there exists an algorithm $\boldsymbol{BasisDel}(\mathbf{A}, \mathbf{R}, \mathbf{T_A}, \sigma)$ outputs a basis $\mathbf{T_B}$ for the lattice $\Lambda_q^\perp(\mathbf{B})$ with overwhelming probability, where $\mathbf{B} = \mathbf{AR}^{-1}$. The basis $\mathbf{T_B}$ satisfies $\|\mathbf{T_B}\| \leq \sigma\cdot\sqrt{m}$. Then $\mathbf{T_B}$ is distributed statistically close to distribution $\boldsymbol{RandBasis}$ $(\mathbf{T}, \sigma)$ where $\mathbf{T}$ is any basis of lattice $\Lambda_q^\perp(\mathbf{AR}^{-1})$ satisfying $\|\widetilde{\mathbf{T}}\| \leq \sigma/\omega(\sqrt{\log m})$.*

Remark: If $\mathbf{R}$ is a product of ℓ matrices sampled from $\mathcal{D}_{m\times m}$ then the bound on σ degrades to $\sigma > \|\widetilde{\mathbf{T_A}}\|\cdot(\sigma_\mathbf{R}\cdot\sqrt{m}\cdot\omega(\sqrt{\log m}))^\ell\cdot\omega(\log m)$.

LEMMA 4 ([2]). *Let $m \geq 2n\log q$ and $q > 2$. For all but at most q^{-n} fraction of rank n matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ the algorithm $\boldsymbol{SampleRwithBasis}(\mathbf{A})$ outputs a matrix $\mathbf{R} \in \mathbb{Z}^{m\times m}$ sampled from a distribution statistically close to $\mathcal{D}_{m\times m}$. The generated short basis $\mathbf{T_B}$ for $\Lambda_q^\perp(\mathbf{B})$ satisfies $\|\widetilde{\mathbf{T_B}}\| \leq \sigma_\mathbf{R}/\omega(\sqrt{\log m})$ with overwhelming probability.*

Next, when given any short basis for a lattice rather than a trapdoor [11], we will introduce an algorithm for inverting the LWR instances.

LEMMA 5. *For any $n \geq 1$, $q \geq 2$, enough large $m \geq O(n\log q)$, given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T_A} = \{\mathbf{t}_1, .., \mathbf{t}_m\} \in \mathbb{Z}^{m\times m}$for lattice $\Lambda_q^\perp(\mathbf{A})$ with $\|\mathbf{T_A}\| \leq p/(2\sqrt{m})$, some vector $\boldsymbol{c} \in \mathbb{Z}_p^m$ such that $\boldsymbol{c} = \lfloor\mathbf{A}^t\boldsymbol{s}\rfloor_p$, there exists an algorithm $\mathbf{LWRInvert2}(\mathbf{T_A}, \mathbf{A}, \boldsymbol{c})$ that can output $\boldsymbol{s}$ with overwhelming probability.*

PROOF. When given m LWR samples ($\mathbf{A}$, $\boldsymbol{c} = \lfloor\mathbf{A}^t\boldsymbol{s}\rfloor_p$), we firstly do the following transformation: $\boldsymbol{c}' = \lceil(q/p)\cdot\boldsymbol{c}\rceil = \lceil(q/p)((p/q)\mathbf{A}^t\boldsymbol{s} + \boldsymbol{e}')\rceil = \mathbf{A}^t\boldsymbol{s} + \boldsymbol{e}$, where $\boldsymbol{e}' \in (-1, 0]^m$ and $\boldsymbol{e} \in (-q/p, 0]^m$. With the basis $\mathbf{T_A}$ for lattice $\Lambda_q^\perp(\mathbf{A})$, then we have $\boldsymbol{c}'' = \mathbf{T}_\mathbf{A}^t\cdot\boldsymbol{c}' = \mathbf{T}_\mathbf{A}^t\cdot\mathbf{A}^t\boldsymbol{s} + \mathbf{T}_\mathbf{A}^t\cdot\boldsymbol{e} = \mathbf{T}_\mathbf{A}^t\boldsymbol{e}\ \textit{mod}\ q$. For each coordinate of $\boldsymbol{c}''$, we have $\|\boldsymbol{c}''\|_\infty = \max_{i=1}^m\{\|\langle\mathbf{t}_i, \boldsymbol{e}\rangle\|\} \leq \|\mathbf{T_A}\|\cdot\sqrt{m}q/p$. With the condition that $\|\mathbf{T_A}\| \leq p/(2\sqrt{m})$, then we have $\|\boldsymbol{c}''\|_\infty \leq q/2$. Therefore, $\boldsymbol{c}'' = \mathbf{T}_\mathbf{A}^t\boldsymbol{e}$ and we can recover $\boldsymbol{e}$. Then with Gaussian elimination we can recover the secret $\boldsymbol{s}$ from $\boldsymbol{c}' - \boldsymbol{e} = \mathbf{A}^t\boldsymbol{s}$ mod q. □

3. OUR LWR-BASED SCHEMES

In this section, we use the technique in [2] to construct our HIBE and D-HIBE schemes in random oracle model. Similar with [2], we also use a hash function $\mathbf{H}$: $\{0,1\}^* \to \mathbb{Z}^{m\times m}$ to map the identity $\boldsymbol{id}$ to a matrix in $\mathbb{Z}^{m\times m}$. We require that the output $\mathbf{H}(\boldsymbol{id})$ is distributed according to $\mathcal{D}_{m\times m}$ over the choice of the random oracle $\mathbf{H}$.

3.1 A HIBE scheme with Random Oracles

In this section, we describe our HIBE scheme of depth d that is adaptive secure in the random oracle model. The algorithm of our HIBE scheme is described as follows:

KeyGen$(1^\lambda)\to$ (PP, *msk*): The algorithm calls **GenTrap**$(1^n,\ 1^m,\ q)$ to sample an uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$ and chooses a matrix $\mathbf{A}_1$ uniformly from $\mathbb{Z}_q^{n\times m}$. Then the public key PP and master key *msk* are

$$\text{PP} := (\mathbf{A}, \mathbf{A}_1); msk := (\mathbf{T_A})$$

Derive(PP, *msk*, $\boldsymbol{id}_{\ell+1}$, $\mathbf{SK}_{\boldsymbol{id}_\ell}$) $\to \mathbf{SK}_{\boldsymbol{id}_{\ell+1}}$: Given the input with public parameter PP, a private key $\mathbf{SK}_{\boldsymbol{id}_\ell}$ corresponding to an identity $\boldsymbol{id}_\ell = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_\ell\}$ and an identity $\boldsymbol{id}_{\ell+1} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_{\ell+1}\}$, the algorithm works as follows:

1. Inquire the hash function $\mathbf{H}$ with identity $\boldsymbol{id}_i$ for $i = 1$ to ℓ and set $\mathbf{R}_{\boldsymbol{id}_\ell} = \mathbf{H}(\boldsymbol{id}_\ell)\cdots\mathbf{H}(\boldsymbol{id}_2)\mathbf{H}(\boldsymbol{id}_1)$. Then compute $\mathbf{F}_{\boldsymbol{id}_\ell} = \mathbf{AR}_{\boldsymbol{id}_\ell}^{-1}$ and $\mathbf{SK}_{\boldsymbol{id}_\ell}$ is a short basis for $\Lambda_q^\perp(\mathbf{F}_{\boldsymbol{id}_\ell})$;
2. Compute $\mathbf{R} = \mathbf{H}(\boldsymbol{id}_{\ell+1})$ and let $\mathbf{F}_{\boldsymbol{id}_{\ell+1}} = \mathbf{F}_{\boldsymbol{id}_\ell}\mathbf{R}^{-1}$;
3. Evaluate $\mathbf{T}' \leftarrow$ **BasisDel**($\mathbf{F}_{\boldsymbol{id}_\ell}$, $\mathbf{R}$, $\mathbf{SK}_{\boldsymbol{id}_\ell}$, $\sigma_{\ell+1}$) to obtain a short random basis for $\Lambda_q^\perp(\mathbf{F}_{\boldsymbol{id}_{\ell+1}})$, where $\sigma_{\ell+1} > \|\widetilde{\mathbf{SK}}_{\boldsymbol{id}_\ell}\| \cdot \sigma_{\mathbf{R}} \cdot \sqrt{m} \cdot \omega(\log^{3/2} m)$;
4. Output the private key $\mathbf{SK}_{\boldsymbol{id}_{\ell+1}} = \mathbf{T}'$.

Enc(PP, $\boldsymbol{id}$, $\boldsymbol{m}$) $\to \boldsymbol{c}$: Given the input with public parameter PP, an identity $\boldsymbol{id}$ of depth $|\boldsymbol{id}| = \ell$ and a message $\boldsymbol{m} \in \mathbb{Z}_p^m$, the algorithm computes $\mathbf{R}_{\boldsymbol{id}} := \mathbf{H}(\boldsymbol{id}_\ell)\cdots\mathbf{H}(\boldsymbol{id}_2)\mathbf{H}(\boldsymbol{id}_1)$ and lets $\mathbf{F}_{\boldsymbol{id}} = \mathbf{AR}_{\boldsymbol{id}}^{-1}$. Then the algorithm randomly chooses $\boldsymbol{s} \leftarrow \mathbb{Z}_q^{n*}$ and outputs the ciphertext $\boldsymbol{c} = (\mathbf{c_0}, \mathbf{c_1})$, where

$$\boldsymbol{c}_0 := \lfloor \mathbf{F}_{\boldsymbol{id}}^t \boldsymbol{s} \rceil_p; \boldsymbol{c}_1 := \lfloor \mathbf{A}_1^t \boldsymbol{s} \rceil_p + \boldsymbol{m} \bmod p$$

Dec(PP, $\mathbf{SK}_{\boldsymbol{id}}$, $\boldsymbol{c}$) $\to \boldsymbol{m}$ or $\perp$: Given the input with public parameter PP, an identity $\boldsymbol{id}$ of depth $|\boldsymbol{id}| = \ell$ with the private key $\mathbf{SK}_{\boldsymbol{id}}$ and a ciphertext $\boldsymbol{c}$, the algorithm computes $\mathbf{F}_{\boldsymbol{id}} = \mathbf{AR}_{\boldsymbol{id}}^{-1} \in \mathbb{Z}_q^{n\times m}$ as before and restores the randomness $\boldsymbol{s}$ from **LWRInvert2**($\mathbf{SK}_{\boldsymbol{id}}$, $\mathbf{F}_{\boldsymbol{id}}$, $\mathbf{c_0}$). Finally, the algorithm recovers the message $\boldsymbol{m} = \mathbf{c_1} - \lfloor \mathbf{A}_1^t \boldsymbol{s} \rceil_p \bmod p$.

3.1.1 Parameter and Correctness

In this scheme, the choice of parameter should match the requirement of algorithm **BasisDel** for different depth $\ell < d$ in the hierarchy. The parameter and correctness of this scheme are stated in Lemma 6.

LEMMA 6. *For any $n \geq 1$, $m \geq O(n \log q)$, $p > m^{\frac{3}{2}d+1} \cdot \omega(\log^{2d} m)$, $q \geq 4mpB$ and p divides q, given the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$ generated by **GenTrap**$(1^n,\ 1^m,\ q)$, the decryption algorithm **Dec**(*PP*, $\mathbf{SK}_{\boldsymbol{id}}$, $\mathbf{c}$) *will output $\boldsymbol{m}$ with overwhelming probability over all choices of* PP, *msk and the message $\boldsymbol{m} \xleftarrow{\$} \mathbb{Z}_p^m$.*

PROOF. Algorithm **BasisDel** at level ℓ can operate correctly when σ_ℓ satisfies $\sigma_\ell > \|\widetilde{\mathbf{SK}}_{\boldsymbol{id}_{\ell-1}}\| \cdot \sigma_{\mathbf{R}} \cdot \sqrt{m} \cdot \omega(\log^{3/2} m)$ according to lemma 3. From Lemma 2, the private key $\mathbf{SK}_{\boldsymbol{id}_{\ell-1}}$ at level $\ell - 1$ satisfies $\|\widetilde{\mathbf{SK}}_{\boldsymbol{id}_{\ell-1}}\| \leq \sigma_{\ell-1} \cdot \sqrt{m}$ with overwhelming probability. Then the requirement on σ_ℓ follows from $\sigma_\ell > \sigma_{\ell-1} \cdot m^{3/2} \cdot \omega(\log^2 m)$, with which we have

$$\begin{cases} \sigma_\ell & > & \sigma_1 \cdot [m^{3/2} \cdot \omega(\log^2 m)]^{\ell-1} \\ \sigma_1 & > & \|\widetilde{\mathbf{T_A}}\| \cdot \sigma_{\mathbf{R}} \cdot \sqrt{m} \cdot \omega(\log^{3/2} m) \end{cases}$$

and imply that $\sigma_\ell > \|\widetilde{\mathbf{T_A}}\| \cdot m^{\frac{3}{2}\ell - \frac{1}{2}} \cdot \omega(\log^{2\ell} m)$. Thus, under such condition on σ_ℓ the **BasisDel** can delegate the private key $\mathbf{SK}_{\boldsymbol{id}_\ell}$ at level ℓ. From Lemma 1, we have

$$\|\mathbf{SK}_{\boldsymbol{id}_\ell}\| \leq \|\widetilde{\mathbf{T_A}}\| \cdot m^{\frac{3}{2}\ell} \cdot \omega(\log^{2\ell} m) \leq m^{\frac{3}{2}\ell + \frac{1}{2}} \cdot \omega(\log^{2\ell} m)$$

With the parameter that $p > m^{\frac{3}{2}d+1} \cdot \omega(\log^{2d} m)$, we have $\|\mathbf{SK}_{\boldsymbol{id}_\ell}\| \leq p/(2\sqrt{m})$, which meets the requirement of Lemma 5, for each $\ell = 1, ..., d$. Given the ciphertext $\boldsymbol{c} = (\boldsymbol{c}_0, \boldsymbol{c}_1)$, we can reconstruct the randomness $\boldsymbol{s}$ correctly with overwhelming probability by the LWR inversion algorithm **LWRInvert2**($\mathbf{SK}_{\boldsymbol{id}}$, $\mathbf{F}_{\boldsymbol{id}}$, $\mathbf{c_0}$). With the randomness $\boldsymbol{s}$, the decryption algorithm can deterministically reconstruct the message $\boldsymbol{m}$ from $\boldsymbol{c}_1 - \lfloor \mathbf{A}_1^t \boldsymbol{s} \rceil_p \bmod p$. □

3.1.2 INDr-ID-CPA Security

Based on the LWR assumption we can prove our HIBE scheme with random oracles is adaptive secure.

THEOREM 2. *If there is an INDr-ID-CPA adversary $\mathcal{A}$ attacking the* HIBE *scheme with the parameter in Lemma 6, the hash function $\mathbf{H}$ is a random oracle defined as before and $\mathbf{Q_H}$ is the maximum number of queries to $\mathbf{H}$ that $\mathcal{A}$ can issue, then there exists an algorithm $\mathcal{B}$ attacking the* $\text{LWR}_{2m,n,q,p}$ *problem. In particular, the advantage of $\mathcal{A}$ is*

$$\mathbf{Adv}_{\text{HIBE},\mathcal{A}}^{indr\text{-}id\text{-}cpa} \leq d\mathbf{Q}_{\mathbf{H}}^d \cdot \mathbf{Adv}_{\text{LWR}_{2m,n,q,p}}^{\mathcal{B}} + negl(\lambda)$$

(Proof of sketch). The algorithm $\mathcal{B}$ randomly chooses d indexes $\mathbf{Q}_1^*$, ..., $\mathbf{Q}_d^* \xleftarrow{\$} [\mathbf{Q_H}]$ and samples d random matrices $\mathbf{R}_1^*, ..., \mathbf{R}_d^* \leftarrow \mathcal{D}_{m\times m}$. Then $\mathcal{B}$ constructs $\mathbf{A}_0$ from the given LWR challenge, chooses a random $r \in [d]$ uniformly and sets $\mathbf{A} := \mathbf{A}_0\mathbf{R}_r^* \cdots \mathbf{R}_1^*$. By calling **SampleRwithBasis**($\mathbf{A}_i$) for $\mathbf{A}_i := \mathbf{A}(\mathbf{R}_{i-1}^* \cdots \mathbf{R}_2^*\mathbf{R}_1^*)^{-1}$, $\mathcal{B}$ can answer hash queries and private key queries for every node in the hierarchy except for the challenge identity $\boldsymbol{id}^*$. Then $\mathcal{B}$ generate a ciphertext respect to $\boldsymbol{id}^*$ with the given LWR challenge and send it to $\mathcal{A}$. We observe that $\mathcal{B}$'s advantage is that same as $\mathcal{A}$'s, conditioned on $\mathcal{B}$ not aborting. By a standard argument, the probability that $\mathcal{B}$ does not abort is $\mathbf{Pr}[\mathcal{S} \text{ not abort}] \geq \frac{1}{d\mathbf{Q}_{\mathbf{H}}^d} - negl(\lambda)$.

3.2 A D-HIBE scheme with Random Oracles

From the above HIBE scheme, we can easily construct our D-HIBE scheme with random oracles based on the LWR problem. The algorithm of our D-HIBE scheme in the random oracle model is described as follows:

KeyGen$(1^\lambda)\to$ (PP, *msk*): The algorithm calls **GenTrap**$(1^n,\ 1^m,\ q)$ to sample a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_A}$ for $\Lambda_q^{\perp}(\mathbf{A})$. Then the public key PP and master key *msk* are

$$\text{PP} := \mathbf{A}; msk := \mathbf{T_A}$$

Derive(PP, *msk*, $\boldsymbol{id}_{\ell+1}$,$\mathbf{SK}_{\boldsymbol{id}_\ell}$) $\to \mathbf{SK}_{\boldsymbol{id}_{\ell+1}}$: The **Derive** algorithm is as the same with the algorithm in our above HIBE scheme in the random oracle model.

Enc(PP, $\boldsymbol{id}$, $\boldsymbol{m}$) $\to \boldsymbol{c}$: Given the input with the public key PP, an identity $\boldsymbol{id}$ of depth $|\boldsymbol{id}| = \ell$ and a message $\boldsymbol{m} \in \{0,1\}^n$, the algorithm computes $\mathbf{R}_{\boldsymbol{id}} := \mathbf{H}(\boldsymbol{id}_\ell)\cdots\mathbf{H}(\boldsymbol{id}_2)\mathbf{H}(\boldsymbol{id}_1)$ and lets $\mathbf{F}_{\boldsymbol{id}} = \mathbf{AR}_{\boldsymbol{id}}^{-1}$. Finally, the algorithm outputs the ciphertext $\boldsymbol{c} := \lfloor \mathbf{F}_{\boldsymbol{id}}^t \boldsymbol{m} \rfloor_p$.

Dec(PP, $\mathbf{SK}_{\boldsymbol{id}}$, $\boldsymbol{c}$) $\to \boldsymbol{m}$ or $\perp$: Given the input with public key PP, an identity $\boldsymbol{id}$ of depth $|\boldsymbol{id}| = \ell$ with the private key $\mathbf{SK}_{\boldsymbol{id}}$ and a ciphertext $\boldsymbol{c}$, the algorithm constructs $\mathbf{F}_{\boldsymbol{id}} = \mathbf{AR}_{\boldsymbol{id}}^{-1}$ as before and then recovers the message $\boldsymbol{m}$ from **LWRInvert2**($\mathbf{SK}_{\boldsymbol{id}}$, $\mathbf{F}_{\boldsymbol{id}}$, $\boldsymbol{c}$).

3.2.1 Parameter and Correctness

In this scheme, the choice of parameter is similar with Lemma 6, the difference is the setting of the parameter for the underlying LWR problem. The parameter and correctness of this scheme are stated in Lemma 7.

LEMMA 7. *For any $n \geq 1$, $m \geq O(n \log q)$, $p > m^{\frac{3}{2}d+1} \cdot \omega(\log^{2d} m)$ and $q \geq 2mpB$ is a prime, given the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_A}$ for $\Lambda_q^{\perp}(\mathbf{A})$ generated by **GenTrap**$(1^n, 1^m, q)$, the decryption algorithm **Dec**(PP, $\mathbf{SK}_{\boldsymbol{id}}, \boldsymbol{c}$) will output $\boldsymbol{m}$ with overwhelming probability over all choices of* PP, *msk and the message $\boldsymbol{m} \in \{0,1\}^n$.*

PROOF. This proof is similar to Lemma 6, except that the choice of p and q. According to the requirement of the $\text{LWR}_{m,n,q,p}$ problem in 1, we need $q \geq 2mpB$ is a prime, which allows that the choice on the message $\boldsymbol{m}$ can be any distribution over $\mathbb{Z}_q^n$. □

3.2.2 PRIV1-IND-ID Security

For security, the messages are chosen from the distributions with sufficient entropy over $\{0,1\}^n$. Based on the hardness of the LWR problem we can prove our D-HIBE scheme with random oracles is PRIV1-IND-ID secure.

THEOREM 3. *If there is an PRIV1-IND-ID adversary $\mathcal{A}$ attacking the* D-HIBE *scheme with above parameter in Lemma 7 for any x sources messages where $x \geq t\log q + \omega(\log n)$, the hash function $\mathbf{H}$ is a random oracle defined as before and $\mathbf{Q_H}$ is the maximum number of queries to $\mathbf{H}$ that $\mathcal{A}$ can issue, then there exists an algorithm $\mathcal{B}$ attacking the* $\text{LWR}_{m,n,q,p}$ *problem. In particular, the advantage of $\mathcal{A}$ is*

$$\mathbf{Adv}_{\text{D-HIBE},\mathcal{A}}^{priv1\text{-}ind\text{-}id} \leq d\mathbf{Q}_{\mathbf{H}}^d \cdot \mathbf{Adv}_{\text{LWR}_{m,n,q,p}}^{\mathcal{B}} + negl(\lambda)$$

(Proof of sketch). Similarly to the proof in Theorem 2, we can construct an algorithm $\mathcal{B}$ which can utilize the ability of adversary $\mathcal{A}$ to solve the $\text{LWR}_{m,n,q,p}$ problem. The difference from the proof in Theorem 2 is the phase of generating the challenge ciphertext. Given the challenge identity $\boldsymbol{id}^*$=$\{\boldsymbol{i}_1^*, ..., \boldsymbol{i}_\ell^*\}$ of length $|\mathbf{id}^*| = \ell$ and messages $\boldsymbol{m}_0$, $\boldsymbol{m}_1$ $\in \{0,1\}^n$, $\mathcal{B}$ sets $\mathbf{F}_{\boldsymbol{id}^*} := \mathbf{A} \cdot \mathbf{R}_{1,\boldsymbol{i}_1^*}^{-1} \cdots \mathbf{R}_{\ell,\boldsymbol{i}_{\ell^*}}^{-1} = \mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ and outputs the ciphertext $\boldsymbol{c}^* = \lfloor \mathbf{F}_{\boldsymbol{id}^*}^t \boldsymbol{m}_b \rfloor_p = \lfloor \mathbf{A}_0 \boldsymbol{m}_b \rfloor_p$. Under the LWR assumption, we have $(\mathbf{A}_0, \lfloor \mathbf{A}_0^t \boldsymbol{m}_0 \rfloor_p) \approx_c (\mathbf{A}_0, \lfloor \mathbf{u} \rfloor_p) \approx_c (\mathbf{A}_0, \lfloor \mathbf{A}_0^t \boldsymbol{m}_1 \rfloor_p)$, where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$.

Next, we use the notion of binary tree encryption, which means that the identities at each level are binary, to remove the random oracles in the above HIBE and D-HIBE schemes. Then, we will introduce our HIBE and D-HIBE schemes with depth d that are selective-secure in the standard model.

3.3 A HIBE Scheme in the Standard Model

The algorithm of our HIBE scheme in the standard model is described as follows:

KeyGen(1^λ)$\to$ (PP, *msk*): The algorithm calls **GenTrap**(1^n, 1^m, q) to sample a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_A}$ for $\Lambda_q^{\perp}(\mathbf{A})$, samples $2d$ matrices $\mathbf{R}_{1,0}, \mathbf{R}_{1,1}, ..., \mathbf{R}_{d,0}, \mathbf{R}_{d,1}$ from the distribution $\mathcal{D}_{m\times m}$ and chooses a random matrix $\mathbf{A}_1$ uniformly from $\mathbb{Z}_q^{n\times m}$. Then the public key PP and master key *msk* are

$$\text{PP} := (\mathbf{A}, \mathbf{A}_1, \mathbf{R}_{1,0}, \mathbf{R}_{1,1}, ..., \mathbf{R}_{d,0}, \mathbf{R}_{d,1}); msk := (\mathbf{T_A})$$

Derive(PP, *msk*, $\boldsymbol{id}_{\ell+1}$, $\mathbf{SK}_{\boldsymbol{id}_\ell}$) $\to \mathbf{SK}_{\boldsymbol{id}_{\ell+1}}$: Given the input with public key PP, a private key $\mathbf{SK}_{\boldsymbol{id}_\ell}$ corresponding to an identity $\boldsymbol{id}_\ell = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_\ell\} \in \{0,1\}^\ell$, where $\ell < d$ and an identity $\boldsymbol{id}_{\ell+1} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_{\ell+1}\}$, the **Derive** algorithm works as follow:

1. Compute $\mathbf{F}_{\boldsymbol{id}_\ell} = \mathbf{AR}_{1,\boldsymbol{i}_1}^{-1} \cdots \mathbf{R}_{\ell,\boldsymbol{i}_\ell}^{-1}$ and $\mathbf{SK}_{\boldsymbol{id}_\ell}$ is a short basis for $\Lambda_q^{\perp}(\mathbf{F}_{\boldsymbol{id}_\ell})$;
2. Let $\mathbf{F}_{\boldsymbol{id}_{\ell+1}} = \mathbf{F}_{\boldsymbol{id}_\ell}\mathbf{R}_{\ell+1,\boldsymbol{i}_{\ell+1}}^{-1} \in \mathbb{Z}_q^{n\times m}$;
3. Evaluate $\mathbf{T}' \leftarrow$ **BasisDel**($\mathbf{F}_{\boldsymbol{id}_\ell}$, $\mathbf{R}_{\ell+1,\boldsymbol{i}_{\ell+1}}$, $\mathbf{SK}_{\boldsymbol{id}_\ell}$, $\sigma_{\ell+1}$) to obtain a short random basis for $\Lambda_q^{\perp}(\mathbf{F}_{\boldsymbol{id}_{\ell+1}})$, where $\sigma_{\ell+1} > \|\widetilde{\mathbf{SK}}_{\boldsymbol{id}_\ell}\| \cdot \sigma_{\mathbf{R}} \cdot \sqrt{m} \cdot \omega(\log^{3/2} m)$;
4. Output the private key $\mathbf{SK}_{\boldsymbol{id}_{\ell+1}} = \mathbf{T}'$.

Enc(PP, $\boldsymbol{id}$, $\boldsymbol{m}$) $\to \boldsymbol{c}$: Given the input with public key PP, an identity $\boldsymbol{id} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_\ell\}$ of depth $|\boldsymbol{id}| = \ell$ and a message $\boldsymbol{m} \leftarrow \mathbb{Z}_p^m$, the algorithm sets $\mathbf{F}_{\boldsymbol{id}} = \mathbf{AR}_{1,\boldsymbol{i}_1}^{-1} \cdots \mathbf{R}_{\ell,\boldsymbol{i}_\ell}^{-1} \in \mathbb{Z}_q^{n\times m}$ and then outputs the ciphertext $\boldsymbol{c} = (\mathbf{c_0}, \mathbf{c_1})$, where

$$\boldsymbol{c}_0 := \lfloor \mathbf{F}_{\boldsymbol{id}}^t \boldsymbol{s} \rfloor_p;\ \boldsymbol{c}_1 := \lfloor \mathbf{A}_1^t \boldsymbol{s} \rfloor_p + \boldsymbol{m} \bmod p$$

Dec(PP, $\mathbf{SK}_{\boldsymbol{id}}$, $\boldsymbol{c}$) $\to \boldsymbol{m}$ or $\perp$: Given the input with public key PP, an identity $\boldsymbol{id}$ of depth $|\boldsymbol{id}| = \ell$ with the private key $\mathbf{SK}_{\boldsymbol{id}}$ and a ciphertext $\boldsymbol{c}$, the algorithm computes $\mathbf{F}_{\boldsymbol{id}} = \mathbf{AR}_{1,\boldsymbol{i}_1}^{-1} \cdots \mathbf{R}_{\ell,\boldsymbol{i}_\ell}^{-1}$ and restores $\boldsymbol{s}$ from **LWRInvert2**($\mathbf{SK}_{\boldsymbol{id}}$, $\mathbf{F}_{\boldsymbol{id}}$, $\mathbf{c_0}$). Finally, the algorithm recover the message $\boldsymbol{m} = \boldsymbol{c}_1 - \lfloor \mathbf{A}_1^t \boldsymbol{s} \rfloor_p \bmod p$.

The parameter settings and correctness of this scheme follow from Lemma 6. Based on the LWR assumption we can prove our HIBE scheme in the standard model is selective secure.

THEOREM 4. *If there is an INDr-sID-CPA adversary $\mathcal{A}$ attacking the* HIBE *scheme with the parameter in Lemma 6, then there exists an algorithm $\mathcal{B}$ attacking the* $\text{LWR}_{2m,n,q,p}$ *problem. In particular, the advantage of $\mathcal{A}$ is*

$$\mathbf{Adv}_{\text{HIBE},\mathcal{A}}^{indr\text{-}sid\text{-}cpa} \leq \mathbf{Adv}_{\text{LWR}_{2m,n,q,p}}^{\mathcal{B}} + negl(\lambda)$$

(Proof of sketch). When given the identity $\boldsymbol{id}^*$=$\{\boldsymbol{i}_1^*, ..., \boldsymbol{i}_\ell^*\}$ of length $|\mathbf{id}^*| = \ell$ which will be challenged by $\mathcal{A}$, $\mathcal{B}$ randomly chooses ℓ matrices $\mathbf{R}_{1,\boldsymbol{i}_1^*}, ..., \mathbf{R}_{\ell,\boldsymbol{i}_\ell^*} \sim \mathcal{D}_{m\times m}$, constructs $\mathbf{A}_0$ from the given LWR challenge and sets $\mathbf{A}$:=

$\mathbf{A}_0\mathbf{R}_{\ell,\boldsymbol{i}^*_\ell}\cdots\mathbf{R}_{1,\boldsymbol{i}^*_1}$. For every matrix $\mathbf{F}_k = \mathbf{A}\cdot\mathbf{R}^{-1}_{1,\boldsymbol{i}^*_1}\cdots\mathbf{R}^{-1}_{k,\boldsymbol{i}^*_k}$, where $k = 1, ..., d-1$, $\mathcal{B}$ calls **SampleRwithBasis**($\mathbf{F}_k$) to answer private key queries for every node in the hierarchy except for the challenge identity. Moreover, for the challenge identity it can generate a ciphertext that will help it solve the given LWR challenge with parameter in Lemma 6.

3.4 A D-HIBE Scheme in the Standard Model

The algorithm of our D-HIBE scheme in the standard model is similar with the construction of our HIBE scheme in the standard model and we describe it as follows:

> **KeyGen**(1^λ)→ (PP, *msk*): The algorithm calls **GenTrap**(1^n,1^m,q) to sample a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a short basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$ and samples $2d$ matrices $\mathbf{R}_{1,0}, \mathbf{R}_{1,1}, ..., \mathbf{R}_{d,0}, \mathbf{R}_{d,1}$ from the distribution $\mathcal{D}_{m\times m}$. Then the public key PP and master key *msk* are
>
> $$\text{PP} := (\mathbf{A}, \mathbf{R}_{1,0}, \mathbf{R}_{1,1}, ..., \mathbf{R}_{d,0}, \mathbf{R}_{d,1});\ msk := (\mathbf{T_A})$$
>
> **Derive**(*msk*, $\boldsymbol{id}$, $\mathbf{SK}_{\boldsymbol{id}_\ell}$) → $\mathbf{SK}_{\boldsymbol{id}}$: The **Derive** algorithm is as the same with the algorithm in our above HIBE scheme in the standard model.
>
> **Enc**(PP, $\boldsymbol{id}$, $\boldsymbol{m}$) → $\boldsymbol{c}$: Given the input with public parameter PP, an identity $\boldsymbol{id} = \{\boldsymbol{i}_1, ..., \boldsymbol{i}_\ell\}$ of depth $|\boldsymbol{id}| = \ell$ and a message $\boldsymbol{m} \in \{0,1\}^n$, the algorithm sets $\mathbf{F}_{\boldsymbol{id}} = \mathbf{A}\mathbf{R}^{-1}_{1,\boldsymbol{i}_1}\cdots\mathbf{R}^{-1}_{\ell,\boldsymbol{i}_\ell} \in \mathbb{Z}_q^{n\times m}$ and outputs the ciphertext $\boldsymbol{c} := \lfloor \mathbf{F}^t_{\boldsymbol{id}}\boldsymbol{m}\rceil_p \in \mathbb{Z}_p^m$.
>
> **Dec**(PP, $\mathbf{SK}_{\boldsymbol{id}}$, $\boldsymbol{c}$) → $\boldsymbol{m}$ or ⊥: Given the input with public parameter PP, an identity $\boldsymbol{id}$ of depth $|\boldsymbol{id}| = \ell$ with the private key $\mathbf{SK}_{\boldsymbol{id}}$ and a ciphertext $\boldsymbol{c}$, the algorithm sets $\mathbf{F}_{\boldsymbol{id}} = \mathbf{A}\mathbf{R}^{-1}_{1,\boldsymbol{i}_1}\cdots\mathbf{R}^{-1}_{\ell,\boldsymbol{i}_\ell} \in \mathbb{Z}_q^{n\times m}$ and restores the message $\boldsymbol{m}$ from **LWRInvert2**($\mathbf{SK}_{\boldsymbol{id}}$, $\mathbf{F}_{\boldsymbol{id}}$, $\boldsymbol{c}$).

The parameter settings and correctness of this scheme follow from Lemma 7. Based on the LWR assumption we can prove our D-HIBE scheme in the standard model is selective secure for any x sources messages.

THEOREM 5. *If there is an PRIV1-IND-sID adversary $\mathcal{A}$ attacking the* D-HIBE *scheme with the parameter in Lemma 7 for any x sources messages where $x \geq t\log q + \omega(\log n)$, then there exists an algorithm $\mathcal{B}$ attacking the $LWR_{m,n,q,p}$ problem. In particular, the advantage of $\mathcal{A}$ is*

$$\mathbf{Adv}^{priv1\text{-}ind\text{-}sid}_{\text{D-HIBE},\mathcal{A}} \leq \mathbf{Adv}^{\mathcal{B}}_{\text{LWR}_{m,n,q,p}} + negl(\lambda)$$

(Proof of sketch). Similarly with Theorem 3 and Theorem4, we can construct an algorithm $\mathcal{B}$ which can utilize the ability of adversary $\mathcal{A}$ to solve the $\text{LWR}_{m,n,q,p}$ problem. The difference from the proof in Theorem 4 is the phase of challenge ciphertext. Given the challenge identity $\boldsymbol{id}^*$=$\{\boldsymbol{i}^*_1, ..., \boldsymbol{i}^*_\ell\}$ of length $|\mathbf{id}^*| = \ell$ and messages $\boldsymbol{m}_0$, $\boldsymbol{m}_1 \in \{0,1\}^n$, $\mathcal{B}$ sets $\mathbf{F}_{\boldsymbol{id}^*} := \mathbf{A}\cdot\mathbf{R}^{-1}_{1,\boldsymbol{i}^*_1}\cdots\mathbf{R}^{-1}_{\ell,\boldsymbol{i}_{\ell^*}} = \mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ and outputs the ciphertext $\boldsymbol{c}^* = \lfloor \mathbf{F}^t_{\boldsymbol{id}^*}\boldsymbol{m}_b\rceil_p = \lfloor \mathbf{A}_0\boldsymbol{m}_b\rceil_p$. Under the LWR assumption, we have $(\mathbf{A}_0, \lfloor \mathbf{A}_0^t\boldsymbol{m}_0\rceil_p) \approx_c (\mathbf{A}_0, \lfloor \mathbf{u}\rceil_p) \approx_c (\mathbf{A}_0, \lfloor \mathbf{A}_0^t\boldsymbol{m}_1\rceil_p)$, where $\mathbf{u} \overset{\$}{\leftarrow} \mathbb{Z}_q^m$.

4. CONCLUSION

In summary, we propose two HIBE schemes and two D-HIBE schemes with and without random oracles based on the LWR problem over small modulus.

5. ACKNOWLEDGMENTS

This research is supported by the National Nature Science Foundation of China (No.61379137, No.61272040 and No.61502480), the National Basic Research Program of China (973 project) (No.2013CB338002), and IIE's Cryptography Research Project (No.Y4Z0061403).

6. REFERENCES

[1] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H) IBE in the standard model. In *Advances in Cryptology–EUROCRYPT 2010*, pages 553–572. Springer, 2010.

[2] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Advances in Cryptology–CRYPTO 2010*, pages 98–115. Springer, 2010.

[3] J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs. Learning with rounding, revisited. In *Advances in Cryptology–CRYPTO 2013*, pages 57–74. Springer, 2013.

[4] A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Advances in Cryptology - EUROCRYPT 2012*, pages 719–737. Springer, 2012.

[5] M. Bellare, E. Kiltz, C. Peikert, and B. Waters. Identity-based (lossy) trapdoor functions and applications. In *Advances in Cryptology - EUROCRYPT 2012*, pages 228–245. Springer, 2012.

[6] A. Bogdanov, S. Guo, D. Masny, S. Richelson, and A. Rosen. On the hardness of learning with rounding over small modulus. In *Theory of Cryptography*, pages 209–224. Springer, 2016.

[7] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of cryptology*, 25(4):601–639, 2012.

[8] A. Escala, J. Herranz, B. Libert, and C. Ràfols. Identity-based lossy trapdoor functions: new definitions, hierarchical extensions, and implications. In *Public-Key Cryptography–PKC 2014*, pages 239–256. Springer, 2014.

[9] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In *Advances in cryptology - ASIACRYPT 2002*, pages 548–566. Springer, 2002.

[10] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2002*, pages 466–481. Springer, 2002.

[11] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology–EUROCRYPT 2012*, pages 700–718. Springer, 2012.

[12] C. Peikert and B. Waters. Lossy Trapdoor Functions and their Applications. In *STOC*, pages 187–196, 2008.

[13] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 84–93. ACM, 2005.

[14] X. Xie, R. Xue, and R. Zhang. Deterministic public key encryption and identity-based encryption from lattices in the auxiliary-input setting. In *Security and Cryptography for Networks*, pages 1–18. Springer, 2012.