Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Efficient hierarchical identity based encryption scheme in the standard model over lattices*

Feng-he WANG†‡1, Chun-xiao WANG†1, Zhen-hua LIU2

([1]Department of Mathematics and Physics, Shandong Jianzhu University, Jinan 250014, China)

([2]School of Mathematics and Statistics, Xidian University, Xi'an 710071, China)

†E-mail: fenghe2166@163.com; xiao2166@126.com

Received July 13, 2015; Revision accepted Jan. 21, 2016; Crosschecked July 11, 2016

Abstract: Using lattice basis delegation in a fixed dimension, we propose an efficient lattice-based hierarchical identity based encryption (HIBE) scheme in the standard model whose public key size is only $(dm^2 + mn)\log q$ bits and whose message-ciphertext expansion factor is only $\log q$, where d is the maximum hierarchical depth and (n, m, q) are public parameters. In our construction, a novel public key assignment rule is used to averagely assign one random and public matrix to two identity bits, which implies that d random public matrices are enough to build the proposed HIBE scheme in the standard model, compared with the case in which $2d$ such public matrices are needed in the scheme proposed at Crypto 2010 whose public key size is $(2dm^2 + mn + m)\log q$. To reduce the message-ciphertext expansion factor of the proposed scheme to $\log q$, the encryption algorithm of this scheme is built based on Gentry's encryption scheme, by which m^2 bits of plaintext are encrypted into $m^2 \log q$ bits of ciphertext by a one time encryption operation. Hence, the presented scheme has some advantages with respect to not only the public key size but also the message-ciphertext expansion factor. Based on the hardness of the learning with errors problem, we demonstrate that the scheme is secure under selective identity and chosen plaintext attacks.

Key words: Hierarchical identity based encryption scheme, Lattice-based cryptography, Standard model, Learning with errors problem, Gaussian

http://dx.doi.org/10.1631/FITEE.1500219 **CLC number:** TP309

1 Introduction

Hierarchical identity based encryption (HIBE) is an important cryptographic notation, in which every entity is arranged by a directed tree (Gentry and Silverberg, 2002; Horwitz and Lynn, 2002). In such a tree, the secret key of each child entity is provided by its parent entities. This process is called a delegation process. Note that the delegation process is one-way, which means that a child node cannot use its secret key to recover the secret key of its parent. As a result, the child entity can decrypt every message intended for it, or for its children, but it cannot decrypt messages intended for any other node in the tree, inside its parents.

The best known HIBE constructions, both with and without random oracles, are based on bilinear maps (Boneh *et al.*, 2005; Boyen and Waters, 2006; Gentry and Halevi, 2009; Waters, 2009). More recent HIBE schemes are built over lattices (Agrawal *et al.*, 2010a; 2010b; Cash *et al.*, 2010).

Lattice-based cryptography is the typical post-quantum cryptography, which remains secure even under a quantum attack (Micciancio and Regev, 2004; Cash *et al.*, 2010; Gentry *et al.*, 2010; Wang *et al.*, 2013; 2016). The core technologies of the

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61303198, 61471409, 61472470, and 61402112) and the Natural Science Foundation of Shandong Province, China (No. ZR2013FQ031)

ORCID: Feng-he WANG, http://orcid.org/0000-0002-5510-3133

known lattice-based HIBE constructions are called the lattice basis delegation technologies and are used to delegate the secret keys to the child entities. Formally, let the parent entity's secret key be a 'short' basis $\boldsymbol{T}$ of a certain integer lattice Λ. To delegate the secret key to a child entity, by the lattice basis delegation algorithm, the parent generates a new lattice Λ' derived from Λ and uses $\boldsymbol{T}$ to create a random short basis for Λ' as the secret key to a child entity. Two lattice delegation algorithms with dimension extension were proposed by Agrawal *et al.* (2010a) and Cash *et al.* (2010), respectively. In these algorithms, the dimension of Λ' is larger than that of Λ, which implies that the private keys and ciphertexts become longer and longer as one descends into the hierarchy. Agrawal *et al.* (2010b) proposed a lattice delegation algorithm in a fixed dimension, which operates 'in place', i.e., without increasing the dimension of the lattices involved. Consequently, private keys and ciphertexts have the same length for each node in the hierarchy. With the help of the lattice delegation algorithm in a fixed dimension, an efficient HIBE scheme with random oracle and a HIBE without random oracle were proposed by Agrawal *et al.* (2010b). If we focus on the lattice-based HIBE scheme in the standard model proposed by Agrawal *et al.* (2010b), it is considerably less efficient than the underlying random-oracle system introduced by Agrawal *et al.* (2010b) because of its large public key size and message-ciphertext expansion factor. The reason for this may consist of two aspects:

1. These constructions view an identity as a bit sequence and then assign a matrix to every bit. Then the public keys of these HIBE schemes would consist of $2d$ random public matrices $\boldsymbol{R}_i$ (d is the maximum hierarchical depth). Hence, the public key size in these schemes are as large as $(2dm^2 + mn + n)\log q$ bits, where m and n are the column and row numbers of the lattice matrix respectively, and q is the modular number.

2. There is a large expansion from plaintext to ciphertext in these schemes. In fact, the message-ciphertext expansion factor of the scheme in Agrawal *et al.* (2010b) would be as large as $m \log q + 1$.

Singh *et al.* (2012; 2014) proposed adaptively secure HIBE schemes with small public parameters. In these schemes, the l-bit identity strings are denoted according to their levels l'. With the help of this idea, $l'' = l/l'$ public matrices are required to construct HIBE schemes. Both schemes in Singh *et al.* (2012; 2014) have a large expansion from plaintext to ciphertext. Moreover, the identity bits are assigned to public matrices by means of the above assignment rule.

In this study, we first present a public key assignment rule, by which averagely two identity bits are assigned to one public matrix. Then we combine the presented public key assignment rule with the lattice basis delegation in a fixed dimension (Agrawal *et al.*, 2010b) to design an efficient delegation algorithm for the lattice-based HIBE scheme, in which only d random matrices $\boldsymbol{R}_i$ are needed (compared with the case in which $2d$ matrices with the same size are needed in Agrawal *et al.* (2010b) and Cash *et al.* (2010)). Then an efficient HIBE scheme in the standard model is naturally proposed. Its encryption algorithm is inspired by the public key encryption scheme from Gentry *et al.* (2010). In our construction, the public key size is efficiently reduced to $(dm^2 + mn)\log q$ bits and the message-ciphertext expansion factor is only $\log q$, which implies that an m^2-bit message can be encrypted into an $m^2 \log q$-bit ciphertext. Therefore, compared with the scheme proposed by Agrawal *et al.* (2010b), the proposed scheme would be more efficient with respect to both the public key size and the message-ciphertext expansion factor.

Based on the hardness of the decision variant learning with errors problem, we prove that the scheme is secure against the selective identities and the chosen message attacks in the standard model.

2 Preliminaries

2.1 Notations

Bold lower-case and bold upper-case letters are used to denote vectors and matrices, respectively. When a function is written as $\omega(f(n))$, it means that the function $\omega(f(n))$ grows faster than $cf(n)$ for every constant $c > 0$. We use poly(n) to denote an unspecified function $f(n) = O(n^c)$ for some constant c. When we consider the length of a vector, we always consider its Euclidean norm, which is written as $||\cdot||$. By convention, the norm of a matrix is defined as the norm of its longest column. For any matrix $\boldsymbol{T}$, $\tilde{\boldsymbol{T}}$ denotes the Gram-Schmidt orthogonalized matrix. Denote $D_{\Lambda,\sigma,\boldsymbol{c}}$ as the Gaussian distribution with center $\boldsymbol{c}$ and parameter σ over lattice Λ. Denote Unif$(\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^{m\times m})$ as the uniform distribution over $\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^{m\times m}$.

2.2 Lattice

Given a set of n linearly independent vectors $\boldsymbol{B}=\{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_n\}$, a lattice Λ generated by $\boldsymbol{B}$ is defined as $\Lambda=\{\boldsymbol{Bc}|\boldsymbol{Bc}=c_1\boldsymbol{b}_1+c_2\boldsymbol{b}_2+\ldots+c_n\boldsymbol{b}_n,\ c_i\in\mathbb{Z},\ i=1,2,\ldots,n\}$. We call $\boldsymbol{B}$ a basis of Λ. The basis of a lattice is called the trapdoor basis, if all vectors from such a basis have small norms. We will restrict our attention to a special class of q-ary lattices, which is more easily described by a matrix. More precisely, given integers (q,m,n) and a matrix $\boldsymbol{A}\in\mathbb{Z}_q^{n\times m}$, the m-dimensional q-ary lattice is defined as

$$\Lambda_q^{\perp}(\boldsymbol{A})=\{\boldsymbol{e}\in\mathbb{Z}_q^m,\boldsymbol{Ae}=\boldsymbol{0}\ (\mathrm{mod}\ q)\}.$$

2.3 Gaussian distribution

The discrete Gaussian distribution over a lattice has been widely used in lattice-based cryptography. Given a parameter $\sigma>0$ and center $\boldsymbol{c}$, the discrete Gaussian function on $\mathbb{R}^m$ is defined as

$$\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})=\exp\left(-\frac{\pi||\boldsymbol{x}-\boldsymbol{c}||^2}{\sigma^2}\right).$$

Given a random matrix $\boldsymbol{A}\in\mathbb{Z}_q^{n\times m}$, the discrete Gaussian distribution on lattice $\Lambda_q^{\perp}(\boldsymbol{A})$ is defined as

$$D_{\Lambda_q^{\perp}(\boldsymbol{A}),\sigma,\boldsymbol{c}}(\boldsymbol{x})=\frac{\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})}{\rho_{\sigma,\boldsymbol{c}}(\Lambda_q^{\perp}(\boldsymbol{A}))}.$$

In fact, $D_{\Lambda_q^{\perp}(\boldsymbol{A}),\sigma,\boldsymbol{c}}(\boldsymbol{x})$ is a 'conditional' distribution, which is defined by sampling $\boldsymbol{x}\in\mathbb{R}^n$ from a Gaussian distribution with parameter σ, then under the condition of the event $\boldsymbol{x}\in\Lambda_q^{\perp}(\boldsymbol{A})$. If $\boldsymbol{c}=\boldsymbol{0}$, $\rho_{\sigma,\boldsymbol{0}}$ and $D_{\Lambda_q^{\perp}(\boldsymbol{A}),\sigma,\boldsymbol{0}}$ are abbreviated as ρ_σ and $D_{\Lambda_q^{\perp}(\boldsymbol{A}),\sigma}$, respectively.

The core advantage of the discrete Gaussian distribution on the lattice for cryptographic applications is that a Gaussian distributed vector almost perfectly conceals the information about the trapdoor basis of the lattice. Then the trapdoor basis can be used as a trapdoor of the lattice-based cryptosystem. We next introduce the following important notion about the Gaussian distribution on the lattice:

Given an n-dimensional lattice Λ and $\epsilon>0$, an important notion called the smoothing parameter, $\eta_\epsilon(\Lambda)$, is defined to be the smallest positive σ satisfying $\rho_{1/\sigma}(\Lambda^*\backslash\{0\})\leq\epsilon$, where '*' means dual lattice (Micciancio and Regev, 2004). For almost all matrices $\boldsymbol{A}\in\mathbb{Z}_q^{n\times m}$, there is a negligible ϵ satisfying $\eta_\epsilon(\Lambda_q^{\perp}(\boldsymbol{A}))\leq\omega(\sqrt{\log m})$.

2.4 Gaussian sampling and lattice basis delegation

There is a useful design tool for (H)IBE over lattice called the preimage sampling function (PSF), which is defined by Gaussian sampling (Gentry *et al.*, 2008; Hu *et al.*, 2014). The PSF is a discrete Gaussian sampling algorithm, essentially, by which a short basis of a lattice suffices to act as a trapdoor of a one-way function defined over this lattice. Lemma 1 shows how to generate a random lattice and its trapdoor basis (Alwen and Peikert, 2009). Then Lemma 2 defines a PSF which uses a trapdoor basis of the lattice as a trapdoor.

Lemma 1 (Trapdoor sampling algorithm (Alwen and Peikert, 2009)) Inputting 1^n and parameters $q=\mathrm{poly}(n)$ and $m>5n\log q$, there is a probabilistic polynomial-time (PPT) algorithm which outputs a matrix $\boldsymbol{A}\in\mathbb{Z}_q^{n\times m}$ whose distribution is statistically close to the uniform distribution, and a full-rank set $\boldsymbol{S}\subset\Lambda_q^{\perp}(\boldsymbol{A})$ which satisfies $\|\boldsymbol{S}\|\leq O(n\log q)$. Moreover, $\boldsymbol{S}$ can be efficiently converted to a trapdoor basis $\boldsymbol{T}$ of the lattice $\Lambda_q^{\perp}(\boldsymbol{A})$.

Lemma 2 (Preimage sampling function (Gentry *et al.*, 2008)) Given a trapdoor basis $\boldsymbol{T}$ of an n-dimensional lattice $\Lambda_q^{\perp}(\boldsymbol{A})$, a Gaussian parameter $\sigma>||\widetilde{\boldsymbol{B}}||\cdot\omega(\sqrt{\log n})$, and a center $\boldsymbol{c}\in\mathbb{R}^n$, there is a PPT algorithm, PreSample$(\boldsymbol{A},\boldsymbol{T},\sigma,\boldsymbol{c})$, outputting vector $\boldsymbol{e}$ from a distribution that is statistically close to the Gaussian distribution $D_{\Lambda,\sigma,\boldsymbol{c}}$.

Three lattice basis delegation algorithms have been proposed to design HIBE schemes (Agrawal *et al.*, 2010a; 2010b; Cash *et al.*, 2010). As we have analyzed in Section 1, the public key size and the ciphertext length in these schemes (Agrawal *et al.*, 2010a; 2010b; Cash *et al.*, 2010) are still large. In this study, we use the lattice basis delegation algorithm in a fixed dimension to design a more efficient selective secure HIBE scheme in the standard model.

Before introducing the basis delegation in a fixed dimension, we should first introduce the distribution on matrices whose columns are low norm vectors (Agrawal *et al.*, 2010b).

If $\boldsymbol{R} \pmod q \in \mathbb{Z}_q^{m\times m}$ is invertible, then $\boldsymbol{R}$ is defined to be $\mathbb{Z}_q$-invertible in $\mathbb{Z}^{m\times m}$. If σ is a Gaussian parameter of $D^m_{\mathbb{Z}^m,\sigma}$, we say that a matrix is distributed according to $\mathcal{D}_{m\times m}$ when it is $\mathbb{Z}_q$-invertible and also distributed according to $D^m_{\mathbb{Z}^m,\sigma}$. Thus, the matrix that is distributed according to $\mathcal{D}_{m\times m}$ would be a low norm matrix with an overwhelming probability for having a suitable parameter.

Lemma 3 Let $m > 2n\log q$ and $q > 2$ be parameters. For all n-dimensional matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ except a fraction of at most q^{-n}, there is a PPT algorithm that outputs matrix $\boldsymbol{R} \in \mathbb{Z}^{m\times m}$ whose distribution is statistically close to $\mathcal{D}_{m\times m}$. Moreover, given a trapdoor basis $\boldsymbol{T}$ of $\varLambda_q^{\perp}(\boldsymbol{A})$ and a matrix $\boldsymbol{R}$, there is a PPT algorithm, $\text{BasisDel}(\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{T}, \sigma)$, that outputs a trapdoor basis $\boldsymbol{T}_\text{B}$ of $\varLambda_q^{\perp}(\boldsymbol{A}\boldsymbol{R}^{-1})$ such that $\|\widetilde{\boldsymbol{T}}_\text{B}\| \leq \sigma/\omega(\log q)$ with an overwhelming probability.

Proof The proof of Lemma 3 was shown in Agrawal *et al.* (2010b). Here, we give a simple proof. We first give the PPT algorithm (Algorithm 1) to sample a matrix with a low norm.

Let $\boldsymbol{T}$ be the canonical basis of the lattice $\varLambda_q^{\perp}(\boldsymbol{A})$ and $\sigma_{\boldsymbol{R}}$ be a Gaussian parameter.

Algorithm 1 Small matrix generation

```
Input: $\varLambda_q^{\perp}(\boldsymbol{A})$, $\boldsymbol{T}$, $\sigma_{\boldsymbol{R}}$
Output: $\boldsymbol{R}(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m) \in \mathbb{Z}^{m\times m}$
1: for $i = 1$ to $m$ do
2:     $\boldsymbol{r}_i \leftarrow \text{PreSample}(\mathbb{Z}, \boldsymbol{T}, \sigma_{\boldsymbol{R}}, 0)$
3: end for
4: if $\boldsymbol{R}$ is invertible in $\mathbb{Z}^{m\times m}$ then
5:     return $\boldsymbol{R}$
6: else
7:     Repeat lines 1–3
8: end if
```

Next, we show that the trapdoor basis of $\varLambda_q^{\perp}(\boldsymbol{A}\boldsymbol{R}^{-1})$ can be computed efficiently by a trapdoor basis $\boldsymbol{T}$ of $\varLambda_q^{\perp}(\boldsymbol{A})$ and matrix $\boldsymbol{R}$ (Algorithm 2).

Algorithm 2 Trapdoor basis generation

```
Input: $\boldsymbol{A}, \boldsymbol{T}, \boldsymbol{R}, \sigma$
Output: $\boldsymbol{T}_\text{B}$ // a basis of lattice $\varLambda_q^{\perp}(\boldsymbol{A}\boldsymbol{R}^{-1})$
$\boldsymbol{T}'_\text{B} = \boldsymbol{R}\boldsymbol{T}$
Convert $\boldsymbol{T}'_\text{B}$ to be a basis $\boldsymbol{T}''_\text{B}$ of $\varLambda_q^{\perp}(\boldsymbol{A}\boldsymbol{R}^{-1})$ by Lemma 1
Produce a random basis $\boldsymbol{T}_\text{B}$ from $\boldsymbol{T}''_\text{B}$ by the algorithm in Cash et al. (2010)
```

Therefore, Lemma 3 has been proved.

Lemma 3 is used to design the Extract and Derive algorithms in the HIBE scheme (Agrawal *et al.*, 2010b).

2.5 Learning with errors (LWE) problem and LWE-based encryption

We introduce a lattice problem called the 'learning with errors' (LWE) problem.

Definition 1 For parameters (n, m, q), $\boldsymbol{s} \in \mathbb{Z}_q^n$, and an error distribution χ over $\mathbb{Z}_q^m$, $\mathbb{A}_{\boldsymbol{s},\chi}$ is a distribution obtained by computing $\{\boldsymbol{A}, (\boldsymbol{A}^\text{T}\boldsymbol{s}+\boldsymbol{x}) \pmod q\}$, where $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ is chosen uniformly and randomly and error vector $\boldsymbol{x}$ is distributed according to the error distribution χ. The LWE problem can be defined as 'computing $\boldsymbol{s}$ with a noticeable probability by giving a sample from $\mathbb{A}_{\boldsymbol{s},\chi}$'. The decision variant LWE problem is to distinguish $\mathbb{A}_{\boldsymbol{s},\chi}$ from the uniform distribution.

Regev (2005) showed that for noise distribution $\bar{\boldsymbol{\varPhi}}_\alpha^m$, the hardness of the LWE problem is based on the hardness of the shortest independent vectors problem (SIVP) in the worst case under a quantum reduction. The standard error distribution $\bar{\boldsymbol{\varPhi}}_\alpha^m$ is a Gaussian distribution over $\mathbb{Z}_q^m$ with deviation $q\alpha > \sqrt{n}$. An error vector can be sampled according to the distribution $\bar{\boldsymbol{\varPhi}}_\alpha^m$ as follows: sample m numbers $\eta_1, \eta_2, \ldots, \eta_m$ according to a Gaussian distribution D_α over $\mathbb{R}$, and then compute $e_i = \lfloor q\eta_i \rceil \pmod q$ ($\lfloor x \rceil$ denotes the integer closest to x). Let $\boldsymbol{e} = (e_1, e_2, \ldots, e_m)$ be an error vector in an LWE problem.

Note that a trapdoor basis of the integer lattice $\varLambda_q^{\perp}(\boldsymbol{A})$ can be used to solve an LWE instance $\boldsymbol{y} = (\boldsymbol{A}^\text{T}\boldsymbol{s} + \boldsymbol{e}) \pmod q$ as follows (details are referred to Gentry *et al.* (2010)):

1. Compute $\boldsymbol{T}\boldsymbol{y} = \boldsymbol{T}\boldsymbol{e} \pmod q$. Due to the fact that both $\boldsymbol{T}$ and $\boldsymbol{e}$ are with short norm, $\boldsymbol{T}\boldsymbol{e} \pmod q = \boldsymbol{T}\boldsymbol{e}$ holds with an overwhelming probability.

2. Compute $\boldsymbol{e} = \boldsymbol{T}^{-1}\boldsymbol{T}\boldsymbol{e} \pmod q$.

3. Find vector $\boldsymbol{s}$ from $\boldsymbol{A}, \boldsymbol{e}$, and $\boldsymbol{y}$.

Gentry *et al.* (2010) used an LWE-based trapdoor one-way function to design a chosen plaintext attack (CPA) secure public key encryption algorithm. We introduce it as follows:

1. KeyGen

Generate a random matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor basis $\boldsymbol{T}$ of lattice $\varLambda_q^{\perp}(\boldsymbol{A})$ by the trapdoor sampling algorithm in Lemma 1. Matrix $\boldsymbol{A}$ is the public key and $\boldsymbol{T}$ the secret key.

2. Encryption

Given a message $\boldsymbol{M} \in \mathbb{Z}_2^{m\times m}$, the encrypter randomly and uniformly chooses $\boldsymbol{S} \in \mathbb{Z}_q^{n\times m}$ and an 'error matrix' $\boldsymbol{X} \in \mathbb{Z}_q^{m\times m}$ according to the distribution $\bar{\boldsymbol{\Phi}}_\alpha^{m\times m}$. Then the ciphertext is $\boldsymbol{C} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{S} + 2\boldsymbol{X} + \boldsymbol{M}) \pmod q$.

3. Decryption

Compute $\boldsymbol{E} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{C} \pmod q$ and then output $\boldsymbol{M} = \boldsymbol{T}^{-\mathrm{T}}\boldsymbol{E} \pmod 2$.

The message-ciphertext factor of the above scheme is as small as $\log q$. We use this scheme to design our novel HIBE scheme.

2.6 Hierarchical identity based encryption

A HIBE scheme consists of five algorithms: Setup, Extract, Derive, Encrypt, and Decrypt (Fig. 1).

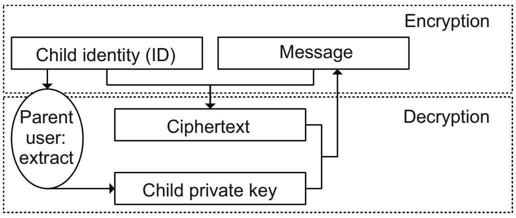

Fig. 1 Hierarchical identity based encryption

1. Setup(λ)

Let λ be a security parameter. Then using λ as the input, the Setup(λ) algorithm outputs the master public key **MPK** and a master secret key **MSK**.

2. Extract(**MSK**; $\mathbf{id}_{|1}$)

The private key generator (PKG) uses the Extract(**MSK**; $\mathbf{id}_{|1}$) algorithm to generate a private key $\mathbf{SK}_{\mathbf{id}_{|1}}$ for identity vector $\mathbf{ID}_{|1}$, in which both the master secret key and an identity vector are the inputs of the Extract(**MSK**; $\mathbf{id}_{|1}$) algorithm.

3. Derive($\mathbf{MPK}; \mathbf{SK}_{\mathbf{id}_{|l}}; \mathbf{id}$)

Given **MPK**, $\mathbf{SK}_{\mathbf{id}_{|l}}$ for $\mathbf{id}_{|l}$, and an identity vector **id**, the Derive($\mathbf{MPK}; \mathbf{SK}_{\mathbf{id}_{|l}}; \mathbf{id}$) algorithm creates a secret key for the $(l+1)$-depth identity vector $\mathbf{id}_{|l+1}$, where $\mathbf{id}_{|l+1}$ is formed by concatenating **id** to the end of $\mathbf{id}_{|l}$.

4. Encrypt($\mathbf{MPK}; \boldsymbol{M}; \mathbf{id}_{|l}$)

Inputting **MPK**, $\mathbf{id}_{|l}$, and a message $\boldsymbol{M}$, the Encrypt($\mathbf{MPK}; \boldsymbol{M}; \mathbf{id}_{|l}$) algorithm outputs a ciphertext $\boldsymbol{C}$.

5. Decrypt($\mathbf{MPK}; \boldsymbol{C}; \mathbf{SK}_{\mathbf{id}_{|l}}$)

Inputting **MPK**, $\boldsymbol{C}$, and $\mathbf{SK}_{\mathbf{id}_{|l}}$, the Decrypt($\mathbf{MPK}; \boldsymbol{C}; \mathbf{SK}_{\mathbf{id}_{|l}}$) algorithm would output the message $\boldsymbol{M}$, if $\boldsymbol{C}$ is an encryption to $\mathbf{id}_{|l}$ and $\mathbf{SK}_{\mathbf{id}_{|l}}$ is for the same $\mathbf{id}_{|l}$.

2.7 Security definition

The security definition of the HIBE scheme is described by a security game between the challenger and the adversary. The adversary can adaptively choose the identity vector to attack in the standard IBE security model (Boneh and Franklin, 2001). A weaker notion of (H)IBE is called the selective security model (Canetti *et al.*, 2003), in which the adversary is forced to announce the target identity that it wishes to attack before the master public key is generated.

For parameter λ, message space M_λ, ciphertext space C_λ, and the maximum depth d of the hierarchy, the selective security game proceeds as follows:

1. Setup

The adversary first receives a hierarchical depth d and then it is asked to announce a target identity $\boldsymbol{I}^* = (\mathrm{id}_1^*, \mathrm{id}_2^*, \ldots, \mathrm{id}_k^*)$ where $k < d$. The challenger generates **MPK** by running the Setup(λ) algorithm.

2. Phase 1

The adversary adaptively chooses some identity vectors to query their secret keys, under the condition that no queries are prefixes of $\boldsymbol{I}^*$. For a queried identity, the challenger obtains a secret key by running the Derive($\mathbf{MPK}; \mathbf{SK}_{\mathbf{id}_{|l}}; \mathbf{id}$) algorithm, and then sends the secret key to the adversary as the answer.

3. Challenge

When the adversary decides to finish Phase 1, it is asked to output a challenged plaintext $\boldsymbol{M} \in M_\lambda$. The challenger randomly chooses $b \in \{0,1\}$ and a ciphertext $\boldsymbol{C} \in C_\lambda$. If $b=0$, the challenger sets $\boldsymbol{C}_b = \mathrm{Encrypt}(\mathbf{MPK}; \boldsymbol{I}^*; \boldsymbol{M})$. If $b=1$, the challenger sets $\boldsymbol{C}_b = \boldsymbol{C}$. The challenger sends the challenge $\boldsymbol{C}_b$ to the adversary.

4. Phase 2

The adversary makes additional adaptive secret key queries the same as in Phase 1 and the challenger responds as before.

5. Guess

Finally, the adversary makes a guess $b' \in \{0,1\}$ and wins if $b = b'$. The advantage of the adversary in attacking the HIBE scheme is defined as Adv =

$p(b = b') - 1/2$.

A HIBE scheme is selectively secure if the advantage is negligible for any PPT adversary to win the above game.

3 Lattice-based hierarchical identity based encryption scheme

3.1 Public key assignment rule

Since known HIBE schemes in the standard model assign every identity bit to a random matrix, the public keys of the HIBE scheme in the standard model would consist of at least $2d$ random matrices. To reduce the public key size of the lattice-based HIBE scheme, we propose a novel public key assignment rule which assigns on average two identity bits to only one random matrix.

Let $\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_d$ be d matrices which are distributed according to $\mathcal{D}_{m\times m}$ and identity $\mathbf{id}_{|d} = (\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_d)$, then the novel public key assignment algorithm works as simply as Algorithm 3 shows.

Algorithm 3 Assignment rule

```
Input: $\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_d$; $\mathbf{id}_{|d} = (\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_d)$
Output: $\{\boldsymbol{R}_{i_1}, \boldsymbol{R}_{i_2}, \ldots, \boldsymbol{R}_{i_{d^*}}\}$
    // return $\boldsymbol{R}_{i_j}$ for $\mathrm{id}_{i_j} = 1$
for $i = 1$ to $d$ do
    if $\mathrm{id}_i$=1 then
        return $\boldsymbol{R}_i$
    else
        Output nothing
    end if
end for
```

Since the chosen matrices $\{\boldsymbol{R}_{i_1}, \boldsymbol{R}_{i_2}, \ldots, \boldsymbol{R}_{i_{d^*}}\}$ for $\mathrm{id}_{i_1} = \mathrm{id}_{i_2} = \ldots = \mathrm{id}_{i_{d^*}} = 1$ are considered as their positions when they are used in the basis delegation algorithm, the following lemma holds:

Lemma 4 The proposed public key assignment rule is a one-to-one map between the identity bits and the orderly subset of the public matrix set $\{\boldsymbol{R}_i | 1 \le i \le d\}$.

Proof The proof of the above conclusion would be clear if we could prove that two different orderly subsets of $\{\boldsymbol{R}_i | 1 \le i \le d\}$ correspond to two different identities. Suppose there are two different subsets of $\{\boldsymbol{R}_i | 1 \le i \le d\}$ which correspond to two identities with the same length. Then there must be a matrix $\boldsymbol{R}_j$ which belongs to one subset but does not belong to the other subset. Then there is only one identity that satisfies $\mathrm{id}_j = 1$. Therefore, the two corresponding identities are not equal.

Suppose identity $\mathbf{id}_{|d}$ is an output of a secure hash function. Then the 0-1 distribution of $\mathbf{id}_{|d}$ is close to being balanced. Therefore, our assignment rule maps about two identity bits to a matrix. As a result, we need only d random matrices to construct a lattice-based HIBE in the standard model used in this study.

Fig. 2 shows how the proposed assignment rule is used in the Extract algorithm of the lattice-based HIBE scheme.

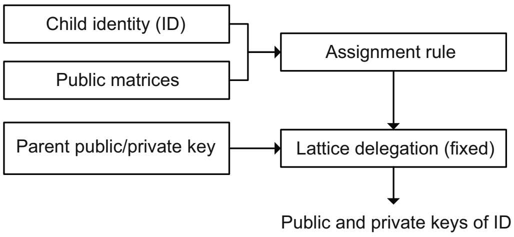
Child identity (ID)
Public matrices
Assignment rule
Parent public/private key
Lattice delegation (fixed)
Public and private keys of ID

Fig. 2 The Extract algorithm of the lattice-based HIBE scheme

3.2 Our scheme

Let n, m, $q = \mathrm{poly}(n)$ be parameters. Given a maximum hierarchical depth d, denote $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_d)$ and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ as the vector forms of the Gaussian parameter and noise parameter, respectively.

Given an identity $\mathbf{id}_{|l} = (\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_l)$ for $l < d$, which is an output of a secure hash function, the scheme operates as follows:

1. Setup(λ)

PKG generates the master public key and the master secret key as follows:

(1) Generate $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and the trapdoor basis $\boldsymbol{T} \in \mathbb{Z}_q^{m\times m}$ by running the algorithm in Lemma 1.

(2) Sample d matrices $\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_d$ from distribution $\mathcal{D}_{m\times m}$ by the first algorithm in Lemma 3.

Then the master public key, **MPK**, and master secret key, **MSK**, are defined as follows:

$$\mathbf{MPK} = (\boldsymbol{A}, \boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_d),\ \mathbf{MSK} = \boldsymbol{T}.$$

2. Derive($\mathbf{MPK}; \mathbf{SK}_{\mathbf{id}_{|l}}; \mathbf{id}$)

Given **MPK**, a 'parent' identity $\mathbf{id}_{|l} = (\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_l)$ with its secret key $\mathbf{SK}_{\mathbf{id}_{|l}}$, and a 'child' identity $\mathbf{id}_{|k} = (\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_l, \ldots, \mathrm{id}_k)$ for $k - l \leq d$, run the Derive($\mathbf{MPK}; \mathbf{SK}_{\mathbf{id}_{|l}}; \mathbf{id}$) algorithm:

(1) Run the public key assignment rule in Section 3.1 to choose $\boldsymbol{R}_i$. More precisely, for $l \leq i \leq k$, if $\mathrm{id}_i = 1$, matrix $\boldsymbol{R}_{i-l}$ is chosen. Otherwise, if $\mathrm{id}_i = 0$, no matrix is chosen. Suppose $\mathrm{id}_{j_i} = 1$, where $i = 1, 2, \ldots, l^*$, $j_i > l$. If the public key of the 'parent' identity is $\boldsymbol{F}_{\mathbf{id}_{|l}}$, then the public key of the 'child' identity is $\boldsymbol{F}_{\mathbf{id}_{|k}}$: $\boldsymbol{F}_{\mathbf{id}_{|k}} = \boldsymbol{F}_{\mathbf{id}_{|l}} \boldsymbol{R}_{j_1-l}^{-1} \boldsymbol{R}_{j_2-l}^{-1} \ldots \boldsymbol{R}_{j_{l^*}-l}^{-1}$.

(2) Evaluate $\mathbf{SK}_{\mathbf{id}_{|k}} \leftarrow \mathrm{BasisDel}(\boldsymbol{F}_{\mathbf{id}_{|l}}, \boldsymbol{R}_{j_1-l}\boldsymbol{R}_{j_2-l}\ldots\boldsymbol{R}_{j_{l^*}-l}, \mathbf{SK}_{\mathbf{id}_{|l}}, \sigma_l)$ to create a short random basis for lattice $\Lambda_q^{\perp}(\boldsymbol{F}_{\mathbf{id}_{|k}})$ (Lemma 3).

(3) Output the delegated private key $\mathbf{SK}_{\mathbf{id}_{|k}}$ and public key $\boldsymbol{F}_{\mathbf{id}_{|k}}$.

3. Extract

For the 1-level identity $\mathbf{id}_{|1}$, the Extract algorithm works in the same way as the Derive($\mathbf{MPK}; \mathbf{SK}_{\mathbf{id}_{|l}}; \mathbf{id}$) algorithm in which $\boldsymbol{F}_{\mathbf{id}_{|0}} = \boldsymbol{A}$ and $\mathbf{SK}_{\mathbf{id}_{|0}} = \mathbf{MSK}$.

4. Encrypt($\mathbf{MPK}; \mathbf{id}_{|l}; \boldsymbol{M}$)

Inputting $\mathbf{MPK}$, $\mathbf{id}_{|l} = (\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_l) \in \{0,1\}^l$ with depth l, and a message matrix $\boldsymbol{M} \in \mathbb{Z}_2^{m\times m}$, Encrypt($\mathbf{MPK}; \mathbf{id}_{|l}; \boldsymbol{M}$) operates as follows:

(1) Choose the public matrices $\boldsymbol{R}_i$ according to the public key assignment rule. Let $\mathrm{id}_{j_1} = \mathrm{id}_{j_2} = \ldots = \mathrm{id}_{j_l^*} = 1$, where l^* is the hamming weight of $\mathbf{id}_{|l}$.

(2) Compute $\boldsymbol{F}_{\mathbf{id}_{|l}} = \boldsymbol{A}\boldsymbol{R}_{j_1}^{-1}\boldsymbol{R}_{j_2}^{-1}\ldots,\boldsymbol{R}_{j_l^*}^{-1} \in \mathbb{Z}_q^{n\times m}$.

(3) Choose $\boldsymbol{S} \leftarrow \mathbb{Z}_q^{n\times m}$ and a noise matrix $\boldsymbol{X} \leftarrow \boldsymbol{\Phi}_{\alpha_l}^{m\times m}$.

(4) Output the ciphertext $\boldsymbol{C}$:

$$\boldsymbol{C} = (\boldsymbol{F}_{\mathbf{id}_{|l}}^{\mathrm{T}}\boldsymbol{S} + 2\boldsymbol{X} + \boldsymbol{M}) \pmod q.$$

5. Decrypt($\mathbf{SK}_{\mathbf{id}_{|l}}; \boldsymbol{C}; \mathbf{MPK}$)

Compute $\boldsymbol{E} = \mathbf{SK}_{\mathbf{id}_{|l}}^{\mathrm{T}}\boldsymbol{C} \pmod q$ and $\boldsymbol{M} = \mathbf{SK}_{\mathbf{id}_{|l}}^{-\mathrm{T}}\boldsymbol{E} \pmod 2$.

3.3 Example

An example with small parameters is given in this subsection to show the operation of the proposed scheme and to show how the decryption algorithm works. For simplicity, we consider a two-dimensional lattice, in which $n = 1$, $m = 2$, and $q = 3139$.

We use only Algorithm 1 when we call the algorithm in Lemma 3, also for simplicity. Clearly, it will not influence the correction of the given example if we omit Algorithm 2 in this example.

1. Setup

Let $\boldsymbol{A} = \begin{pmatrix}-731 & 43\end{pmatrix}$ denote a two-dimensional lattice whose trapdoor basis is

$$\boldsymbol{T} = \begin{pmatrix}13 & -3\\ 75 & 22\end{pmatrix}.$$

Then we choose two small matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ as follows:

$$\boldsymbol{R}_1 = \begin{pmatrix}-6 & 2\\ 8 & -1\end{pmatrix}, \boldsymbol{R}_2 = \begin{pmatrix}11 & -1\\ -13 & 3\end{pmatrix}.$$

It can be checked that both $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are $\mathbb{Z}_q$-invertible. Moreover, we can compute $\boldsymbol{R}_1^{-1} \pmod{3139}$ and $\boldsymbol{R}_2^{-1} \pmod{3139}$ as follows:

$$\boldsymbol{R}_1^{-1} \pmod{3139} = \begin{pmatrix}314 & 628\\ -627 & -1255\end{pmatrix},$$

$$\boldsymbol{R}_2^{-1} \pmod{3139} = \begin{pmatrix}471 & 157\\ -1098 & -1412\end{pmatrix}.$$

2. Derive

Given identity $\mathbf{id} = (0, 1)$, generate the private key

$$\begin{aligned}\boldsymbol{T}_{\mathbf{id}} &= \boldsymbol{R}_2\boldsymbol{T} \pmod{3139}\\ &= \begin{pmatrix}11 & -1\\ -13 & 3\end{pmatrix}\begin{pmatrix}13 & -3\\ 75 & 22\end{pmatrix} \pmod{3139}\\ &= \begin{pmatrix}68 & -55\\ 56 & 105\end{pmatrix}.\end{aligned}$$

3. Encrypt

Given the message matrix $\boldsymbol{M} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$, do the following:

(1) Compute the encryption matrix of $\mathbf{id} = (0, 1)$,

$$\boldsymbol{F}_{\mathbf{id}} = \boldsymbol{A}\boldsymbol{R}_2^{-1} \pmod{3139} = (860, 301).$$

(2) Choosing $\boldsymbol{S} = (137, 312)$ and $\boldsymbol{X} = \begin{pmatrix}2 & -2\\ 4 & 6\end{pmatrix}$, output the ciphertext $\boldsymbol{C}$:

$$\begin{aligned}\boldsymbol{C} &= \left(\boldsymbol{F}_{\mathbf{id}}^{\mathrm{T}}\boldsymbol{S} + 2\boldsymbol{X} + \boldsymbol{M}\right) \pmod{3139}\\ &= \begin{pmatrix}1680 & 1503\\ 434 & -251\end{pmatrix}.\end{aligned}$$

4. Decrypt

(1) Compute

$$\begin{aligned}\boldsymbol{E} &= \boldsymbol{T}_{\mathbf{id}}^{\mathrm{T}}\boldsymbol{C} \pmod{3139}\\ &= \begin{pmatrix} 68 & 56\\ -55 & 105\end{pmatrix}\begin{pmatrix}1680 & 1503\\ 434 & -251\end{pmatrix} \pmod{3139}\\ &= \begin{pmatrix}428 & 256\\ 255 & 845\end{pmatrix}.\end{aligned}$$

Remark 1 We can check $\boldsymbol{E} = \boldsymbol{T}_{\mathbf{id}}^{\mathrm{T}}(2\boldsymbol{X}+\boldsymbol{M})$ over an integer, i.e.,

$$\boldsymbol{T}_{\mathbf{id}}^{\mathrm{T}}(2\boldsymbol{X}+\boldsymbol{M}) = \begin{pmatrix}428 & 256\\ 255 & 845\end{pmatrix} \text{ (over an integer).}$$

(2) Compute

$$\boldsymbol{T}_{\mathbf{id}}^{-1} = \begin{pmatrix} \dfrac{21}{1022} & \dfrac{11}{1022}\\ \dfrac{-14}{5\times 511} & \dfrac{17}{5\times 511}\end{pmatrix} \text{ (over } \mathbb{R}\text{).}$$

(3) Compute

$$\begin{aligned}&\boldsymbol{T}_{\mathbf{id}}^{-\mathrm{T}}\boldsymbol{E} \pmod 2\\ &= \begin{pmatrix} \dfrac{21}{1022} & \dfrac{-14}{5\times 511}\\ \dfrac{11}{1022} & \dfrac{17}{5\times 511}\end{pmatrix}\begin{pmatrix}428 & 256\\ 255 & 845\end{pmatrix} \pmod 2\\ &= \begin{pmatrix}3 & -2\\ 4 & 7\end{pmatrix} \pmod 2\\ &= \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} = \boldsymbol{M}.\end{aligned}$$

Hence, the ciphertext is decrypted correctly.

4 Analysis of the proposed HIBE scheme

4.1 Setting parameters and correctness

With a secure parameter n, in order for the proposed scheme to work correctly, the following things are required:

1. The trapdoor sample algorithm operates to ensure that the Setup algorithm is correct, which needs $m > 6n\log q$ and $q = \mathrm{poly}(n)$ (Alwen and Peikert, 2009),

2. The lattice basis delegation algorithm in a fixed dimension used in the Derive algorithm of the proposed scheme can operate, which needs (Agrawal *et al.*, 2010b)

$$\sigma_l > ||\widetilde{\mathbf{SK}}_{\mathbf{id}_{|l-1}}||\sigma_{l-1}\sqrt{m}\omega\left((\log\, m)^{3/2}\right).$$

So,

$$\sigma_l \geq \sigma_{l-1}m^{3/2}\omega(\log n^{3/2}).$$

3. The LWE-based trapdoor one-way function operates to ensure that the decryption algorithm at l-level operates correctly, which requires that the error term in the decryption algorithm should be less than $q/2$ with high probability $\alpha_l < 1/(\sigma_{l-1}m\omega(\log m))$ and $q \geq \sigma_l m^{3/2}\omega(\log m)$ (Gentry *et al.*, 2010).

Let d be the maximum depth of the hierarchy. To satisfy the above requirements, we set (m, q, σ, α) as follows:

$$\begin{cases} m = dn\log n,\\ q = m^{3/(2d)+2}\omega\left((\log n)^{2d+1}\right),\\ \sigma_l = m^{3/(2l)}\omega\left((\log n)^{2l}\right),\\ \alpha_l < \dfrac{1}{\sigma_{l-1}m\omega(\log m)}.\end{cases}$$

Given the above parameters, PKG can extract the private key for a 1-level user, and an l-level user can also derive the private key for a k-level user ($l \leq k \leq d$). The decryption algorithm also operates correctly.

The correctness of the proposed scheme is proved.

4.2 Security

Theorem 1 If the decision variant LWE problem with the error distribution $\bar{\boldsymbol{\Phi}}_{\alpha}^{m}$ is hard, the proposed HIBE scheme is secure under the selective identities and CPAs.

Proof Suppose there is an adversary $\mathcal{A}$ against the selective identity CPA security with advantage ϵ. Then we first construct a distinguisher $\mathcal{D}$ with the advantage of at least $\epsilon/2$ between two distributions:

$$\Big\{(\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{S}+\boldsymbol{X}) : \boldsymbol{A} \in \mathbb{Z}_q^{n\times m}, \boldsymbol{S} \in \mathbb{Z}_q^{n\times m},$$

$$\boldsymbol{X} \leftarrow \boldsymbol{\Phi}_{\alpha}^{m\times m}, \alpha < \frac{1}{\sigma_d m\omega(\log m)}\Big\}$$

and

$$\{\mathrm{Unif}(\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times m})\}.$$

A selective identity adversary $\mathcal{A}$ outputs the challenge identity $\mathbf{id}^* = (\mathrm{id}_1^*, \mathrm{id}_2^*, \ldots, \mathrm{id}_k^*)$. Suppose the hamming weight of the challenge identity is k^*, and $\mathrm{id}_{j_1}^* = \mathrm{id}_{j_2}^* = \ldots = \mathrm{id}_{j_{k^*}}^* = 1$.

1. Simulation of the attack environment

$\mathcal{D}$ receives a challenging instance $(\boldsymbol{A}_0, \boldsymbol{B})$ from one of two challenge distributions. Then $\mathcal{D}$ prepares the simulated attack environment for the adversary $\mathcal{A}$:

(1) Randomly sample k^* matrices $\boldsymbol{R}_{j_1}, \boldsymbol{R}_{j_2}, \ldots, \boldsymbol{R}_{j_{k^*}}$ from the distribution $\mathcal{D}_{m\times m}$. Set

$$\boldsymbol{A} = \boldsymbol{A}_0\boldsymbol{R}_{j_{k^*}}\boldsymbol{R}_{j_{k^*-1}}\ldots\boldsymbol{R}_{j_1}.$$

For every $i = j_l \in \{j_1, j_2, \ldots, j_{k^*}\}$, set $\boldsymbol{R}_i = \boldsymbol{R}_{j_l}$.

(2) For $i \notin \{j_1, j_2, \ldots, j_{k^*}\}$ and $i \leq d$, generate $\boldsymbol{A}_i \in \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{T}_i \in \mathbb{Z}_q^{m\times m}$ by running the trapdoor sample algorithm in Lemma 1. Then $\boldsymbol{A}_i\boldsymbol{T}_i = \boldsymbol{0} \pmod q$ and $\|\boldsymbol{T}_i\| \leq O(n\log q)$.

(3) Generate a short invertible matrix denoted by $\boldsymbol{R}'_i$ by running the PSF algorithm in Lemma 2 at most m^2 times (Cash *et al.*, 2010):

$$\boldsymbol{R}'_i \leftarrow \text{PreSample}(\boldsymbol{A}_i, \boldsymbol{T}_i, \sigma_d, \boldsymbol{A}_0\boldsymbol{R}_{j_{k^*}}\boldsymbol{R}_{j_{k^*-1}}\ldots\boldsymbol{R}_{j_{i^*}}),$$

where $j_{i^*} \in \{j_1, j_2, \ldots, j_{k^*}\}$ is the first number which is larger than i. Therefore, $\boldsymbol{A}_i\boldsymbol{R}'_i = (\boldsymbol{A}_0\boldsymbol{R}_{j_{k^*}}\boldsymbol{R}_{j_{k^*-1}}\ldots\boldsymbol{R}_{j_{i^*}}) \pmod q$ and $\|\boldsymbol{R}'_i\| \leq \sigma_d\sqrt{m}$.

Then set $\boldsymbol{R}_i = \boldsymbol{R}'_i$ for $i \notin \{j_1, j_2, \ldots, j_{k^*}\}$ and $i \leq d$.

(4) Send the public key $\{\boldsymbol{A}, \boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_d\}$ to the adversary $\mathcal{A}$ (other parameters are as shown as in our scheme).

2. Secret key queries

$\mathcal{A}$ makes key-extraction queries on identities $\mathbf{id}_{|l}$ that are not prefixes of $\mathbf{id}^*$.

Let $|\mathbf{id}| = l \leq d$, where $|\cdot|$ represents the corresponding length. To simplify the description assume $l = d$ (the case $l < d$ is just as easy). Since $\mathbf{id}_{|l}$ are random and not prefixes of $\mathbf{id}^*$, there must be the first position i_0 that satisfies $\text{id}_{i_0} = 1, \text{id}^*{}_{i_0} = 0$. Note that the distinguisher $\mathcal{D}$ holds the trapdoor basis $\boldsymbol{T}_i$ of lattice $\varLambda_q^{\perp}(\boldsymbol{A}_0\boldsymbol{R}_{j_{k^*}}\boldsymbol{R}_{j_{k^*-1}}\ldots\boldsymbol{R}_{j_i^*}\boldsymbol{R}_{i_0}^{-1})$. Then $\mathcal{D}$ does as follows to answer the secret key query for identity $\mathbf{id}_{|l}$:

(1) Choose the matrices $\boldsymbol{R}_{i_j}$ where $\text{id}_{i_j} = 1$ and $i_j > i_0$, as shown in the encryption algorithm. Suppose the number of these matrices $\boldsymbol{R}_{i_j}$ is j'. Hence, the public key matrix of $\mathbf{id}_{|l}$ is

$$\boldsymbol{F}_{\mathbf{id}_{|l}} = \boldsymbol{A}_0\boldsymbol{R}_{j_{k^*}}\boldsymbol{R}_{j_{k^*-1}}\ldots\boldsymbol{R}_{j_{i^*}}\boldsymbol{R}_{i_0}^{-1}\boldsymbol{R}_{i_1}^{-1}\ldots\boldsymbol{R}_{i_{j'}}^{-1}.$$

(2) Generate a trapdoor basis $\boldsymbol{T}_{\mathbf{id}_{|l}}$ of lattice $\varLambda_q^{\perp}(\boldsymbol{F}_{\mathbf{id}_{|l}})$ by using the lattice basis delegation algorithm in a fixed dimension (Lemma 3). Letting $\boldsymbol{A}' = \boldsymbol{F}_{\mathbf{id}_{|l}}\boldsymbol{R}_{i_{j'}}\boldsymbol{R}_{i_{j'-1}}\ldots\boldsymbol{R}_{i_1}$, there is

$$\boldsymbol{T}_{\mathbf{id}_{|l}} \leftarrow \text{BasisDel}(\boldsymbol{A}', \boldsymbol{R}_{i_{j'}}\boldsymbol{R}_{i_{j'-1}}\ldots\boldsymbol{R}_{i_1}, \boldsymbol{T}_{i_0}, \sigma_l).$$

3. Challenge

$\mathcal{A}$ outputs a challenge message $\boldsymbol{M}_0 \in \mathbb{Z}_2^{m\times m}$. Then for a random bit $b \in \{0,1\}$, $\mathcal{D}$ returns $(\boldsymbol{M}_0 + 2\boldsymbol{B}) \pmod q$ as the challenge ciphertext.

4. Phase 2

$\mathcal{A}$ can make more secret key queries which are answered by $\mathcal{D}$ in the same manner as before.

5. Guess

Finally, $\mathcal{A}$ outputs a guess bit. $\mathcal{D}$ outputs 1 if $\mathcal{A}$ guesses the right b, and 0 otherwise.

Then we can analyze the advantage of the distinguisher $\mathcal{D}$ as follows. If $\boldsymbol{B}$ is a uniformly random matrix, $(\boldsymbol{M}_0 + 2\boldsymbol{B}) \pmod q$ is also a uniformly random matrix. So, $\mathcal{D}$ would output 1 with probability 1/2 in this case. If $\boldsymbol{B} = (\boldsymbol{A}_0^{\mathrm{T}}\boldsymbol{S} + \boldsymbol{X}) \pmod q$, the distribution of the challenge ciphertext $(\boldsymbol{A}_0^{\mathrm{T}}\boldsymbol{S}' + 2\boldsymbol{X} + \boldsymbol{M}) \pmod q$ is identical to that of the output of the Encrypt($\mathbf{MPK}$; $\mathbf{id}^*$; $\boldsymbol{M}$) algorithm in which $\boldsymbol{S}' = 2\boldsymbol{S}$ is uniformly distributed. Hence, $\mathcal{D}$ outputs 1 with probability $(1+\epsilon)/2$. Then the advantage of distinguisher $\mathcal{D}$ is $\epsilon/2$.

It is clear that $\mathcal{D}$ can be used to solve the LWE problem with error distribution $\bar{\boldsymbol{\varPhi}}_{\alpha}^{m}$.

4.3 Efficiency

The main advantage of this scheme, compared with the scheme in Agrawal *et al.* (2010b), is that the public key size is efficiently reduced. More precisely, the proposed scheme consists of only d random public matrices, which means that the public key length of this scheme is $(dm^2 + mn)\log q$, while the scheme in Agrawal *et al.* (2010b) consists of $2d$ equal-size matrices, which means that the public key length is $(2dm^2 + mn + n)\log q$. On the other hand, the message-ciphertext expansion factor of this scheme is also controlled to be $\log q$, which implies that an m^2-bit message can be encrypted in an $m^2\log q$-bit ciphertext by one time encryption operation. Comparison of the proposed scheme with the schemes in Singh *et al.* (2012; 2014) shows that the proposed scheme shares some advantages about the space size, especially about the message-ciphertext expansion factor.

Let d be the maximum hierarchical depth, and l'' the length of identity at the ith level for $1 \leq i \leq d$. If we suppose that the secure parameter n is the same in Agrawal *et al.* (2010b), Singh *et al.* (2012; 2014), and this study, then Table 1 gives the details of space efficiency comparison among the four schemes.

In fact, it is possible to combine the proposed assignment rule with the main ideas of Singh *et al.* (2012; 2014) to design a more efficient HIBE scheme with much smaller public parameters. The main idea is that, if we denote an identity string as $(\mathbf{id}_1, \mathbf{id}_2, \ldots, \mathbf{id}_l)$ where $\mathbf{id}_i = (\mathrm{id}_{i_1}, \mathrm{id}_{i_2}, \ldots, \mathrm{id}_{i_{l''}})$, then we need only l'' matrices $\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_{l''}$ to design our HIBE scheme. One aspect that needs improvement in the proposed scheme is that it can achieve only the selective security. More work should be done to find out how to achieve the full security.

Since the main computation operations of the lattice-based HIBE scheme are Gaussian sampling and modular multiplication, we can compare the computation efficiency by the number of these main computation operations, for messages with the same length. The proposed scheme can encrypt m^2 bits by one encryption operation, while the schemes in Agrawal *et al.* (2010b) and Singh *et al.* (2012; 2014) can encrypt only 1 bit by one encryption operation. We should consider the computation cost for a 1-bit message when we compare the computation efficiency. Table 2 shows the details of the computation cost when encrypting a 1-bit message.

5 Conclusions

We have proposed an efficient lattice-based HIBE scheme. The scheme has been proved secure under the selective identities and chosen plaintext attacks in the standard model. Furthermore, compared with a known efficient lattice-based HIBE scheme without random oracles, the proposed scheme has some advantages with respect to the public key size, the message-ciphertext expansion factor, and the computation cost. There are still many open problems that need to be studied, for example, how to design an efficient lattice-based function encryption scheme (Agrawal *et al.*, 2012) and attribute-based encryption (Cheng *et al.*, 2013).

Table 1 Comparison of space efficiency

Scheme	Public key length (bit)	Message-ciphertext expansion factor
Agrawal *et al.* (2010b)	$(2dm^2+mn+n)\log q$	$m\log q+1$
Singh *et al.* (2012)	$(dl''+2)mn\log q$	$[(l+1)m+1]\log q$
Singh *et al.* (2014)	$(l''+2)mn\log q$	$[(l+1)m+1]\log q$
Our scheme	$(dm^2+mn)\log q$	$\log q$

References

Agrawal, S., Boneh, D., Boyen, X., 2010a. Efficient lattice (H)IBE in the standard model. Proc. 29th Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, p.553-572. http://dx.doi.org/10.1007/978-3-642-13190-5_28

Agrawal, S., Boneh, D., Boyen, X., 2010b. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. Proc. 30th Annual Cryptology Conf., p.98-115. http://dx.doi.org/10.1007/978-3-642-14623-7_6

Agrawal, S., Boyen, X., Vaikuntanathan, V., *et al.*, 2012. Functional encryption for threshold functions (or fuzzy IBE) from lattices. Proc. 15th Int. Conf. on Practice and Theory in Public Key Cryptography, p.280-297. http://dx.doi.org/10.1007/978-3-642-30057-8_17

Alwen, J., Peikert, C., 2009. Generating shorter bases for hard random lattices. Proc. 26th Int. Symp. on Theoretical Aspects of Computer Science, p.75-86. http://dx.doi.org/10.4230/LIPIcs.STACS.2009.1832

Boneh, D., Franklin, M., 2001. Identity-based encryption from the Weil pairing. Proc. 21st Annual Int. Cryptology Conf., p.213-229. http://dx.doi.org/10.1007/3-540-44647-8_13

Boneh, D., Boyen, X., Goh, E.J., 2005. Hierarchical identity based encryption with constant size ciphertext. Proc. 24th Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, p.440-456. http://dx.doi.org/10.1007/11426639_26

Boyen, X., Waters, B., 2006. Anonymous hierarchical identity-based encryption (without random oracles). Proc. 26th Annual Int. Cryptology Conf., p.290-307. http://dx.doi.org/10.1007/11818175_17

Canetti, R., Halevi, S., Katz, J., 2003. A forward-secure public-key encryption scheme. Proc. Int. Conf. on the Theory and Applications of Cryptographic Techniques, p.255-271. http://dx.doi.org/10.1007/3-540-39200-9_16

Table 2 Comparison of computation efficiency for each bit message

Scheme	Gaussian sampling		Modular multiplication	
	Encrypt	Decrypt	Encrypt	Decrypt
Agrawal *et al.* (2010b)	2	1	$(l-1)m^2+2mn+n$	m
Singh *et al.* (2012)	1	1	$(l''+l+1)mn+m$	$(l+1)m$
Singh *et al.* (2014)	1	1	$(ll'')mn+m^2+(l''+1)mn^2$	$(l+1)m$
Our scheme	$1/m$	0	$n+j^*-1$	2

Cash, D., Hofheinz, D., Kiltz, E., *et al.*, 2010. Bonsai trees, or how to delegate a lattice basis. Proc. 29th Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, p.523-552. http://dx.doi.org/10.1007/978-3-642-13190-5_27

Cheng, Y., Wang, Z.Y., Ma, J., *et al.*, 2013. Efficient revocation in ciphertext-policy attribute-based encryption based cryptographic cloud storage. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **14**(2):85-97. http://dx.doi.org/10.1631/jzus.C1200240

Gentry, C., Halevi, S., 2009. Hierarchical identity based encryption with polynomially many levels. Proc. 6th Theory of Cryptography Conf., p.437-456. http://dx.doi.org/10.1007/978-3-642-00457-5_26

Gentry, C., Silverberg, A., 2002. Hierarchical ID-based cryptography. Proc. 8th Int. Conf. on the Theory and Application of Cryptology and Information Security, p.548-566. http://dx.doi.org/10.1007/3-540-36178-2_34

Gentry, C., Peikert, C., Vaikuntanathan, V., 2008. Trapdoors for hard lattices and new cryptographic constructions. Proc. 40th Annual ACM Symp. on Theory of Computing, p.197-206. http://dx.doi.org/10.1145/1374376.1374407

Gentry, C., Halevi, S., Vaikuntanathan, V., 2010. A simple BGN-type cryptosystem from LWE. Proc. 29th Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques, p.506-522. http://dx.doi.org/10.1007/978-3-642-13190-5_26

Horwitz, J., Lynn, B., 2002. Toward hierarchical identity-based encryption. Proc. Int. Conf. on the Theory and Applications of Cryptographic Techniques, p.466-481. http://dx.doi.org/10.1007/3-540-46035-7_31

Hu, Y.P., Lei, H., Wang, F.H., *et al.*, 2014. Gaussian sampling of lattices for cryptographic applications. *Sci. China Inform. Sci.*, **57**(7):072112.1-072112.8. http://dx.doi.org/10.1007/s11432-013-4843-4

Micciancio, D., Regev, O., 2004. Worst-case to average-case reductions based on Gaussian measures. Proc. 45th Annual IEEE Symp. on Foundations of Computer Science, p.372-381. http://dx.doi.org/10.1109/FOCS.2004.72

Regev, O., 2005. On lattices, learning with errors, random linear codes, and cryptography. Proc. 37th Annual ACM Symp. on Theory of Computing, p.84-93. http://dx.doi.org/10.1145/1060590.1060603

Singh, K., Pandurangan, C., Banerjee, A.K., 2012. Adaptively secure efficient lattice (H)IBE in standard model with short public parameters. Proc. 2nd Int. Conf. on Security, Privacy, and Applied Cryptography Engineering, p.153-172. http://dx.doi.org/10.1007/978-3-642-34416-9_11

Singh, K., Pandu Rangan, C., Banerjee, A.K., 2014. Efficient lattice HIBE in the standard model with shorter public parameters. Proc. 2nd IFIP TC5/8 Int. Conf. on Information and Communication Technology, p.542-553. http://dx.doi.org/10.1007/978-3-642-55032-4_56

Wang, F.H., Hu, Y.P., Wang, B.C., 2013. Lattice-based linearly homomorphic signature scheme over binary field. *Sci. China Inform. Sci.*, **56**(11):112108.1-112108.9. http://dx.doi.org/10.1007/s11432-012-4681-9

Wang, F.H., Liu, Z.H., Wang, C.X., 2016. Full secure identity-based encryption scheme with short public key size over lattices in the standard model. *Int. J. Comput. Math.*, **93**(6):854-863. http://dx.doi.org/10.1080/00207160.2015.1029464

Waters, B., 2009. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. Proc. 29th Annual Int. Cryptology Conf., p.619-636. http://dx.doi.org/10.1007/978-3-642-03356-8_36