entropy

MDPI

Article

Adaptively Secure Efficient (H)IBE over Ideal Lattice with Short Parameters

Yuan Zhang, Yuan Liu, Yurong Guo, Shihui Zheng and Licheng Wang *

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; zy_bupt@bupt.edu.cn (Y.Z.); yuanl_1011@163.com (Y.L.); gyr_bupt@163.com (Y.G.); shihuizh@bupt.edu.cn (S.Z.)

* Correspondence: wanglc@bupt.edu.cn

Received: 6 October 2020; Accepted: 30 October 2020; Published: 2 November 2020

Abstract: Identity-based encryption (IBE), and its hierarchical extension (HIBE), are interesting cryptographic primitives that aim at the implicit authentication on the users' public keys by using users' identities directly. During the past several decades, numerous elegant pairing-based (H)IBE schemes were proposed. However, most pairing-related security assumptions suffer from known quantum algorithmic attacks. Therefore, the construction of lattice-based (H)IBE became one of the hot directions in recent years. In the setting of most existing lattice-based (H)IBE schemes, each bit of a user's identity is always associated with a parameter matrix. This always leads to drastic but unfavorable increases in the sizes of the system public parameters. To overcome this issue, we propose a flexible trade-off mechanism between the size of the public parameters and the involved computational cost using the blocking technique. More specifically, we divide an identity into l' segments and associate each segment with a matrix, while increasing the lattice modulo slightly for maintaining the same security level. As a result, for the setting of 160-bit identities, we show that the size of the public parameters can be reduced by almost 89.7% (resp. 93.8%) while increasing the computational cost by merely 5.2% (resp. 12.25%) when l' is a set of 16 (resp. 8). Finally, our IBE scheme is extended to an HIBE scheme, and both of them are proved to achieve the indistinguishability of ciphertexts against adaptively chosen identity and chosen plaintext attack (IND-ID-CPA) in the standard model, assuming that the well-known ring learning with error (RLWE) problem over the involved ideal lattices is intractable, even in the post-quantum era.

Keywords: lattice; IBE; adaptive security; short parameter; standard model; RLWE

1. Introduction

Identity-based encryption (IBE), first introduced by Shamir [1], is an interesting public-key encryption mechanism. It reduces the complexity of system and the cost of establishing public-key infrastructure. The public keys are users' identities directly, and the corresponding private keys can only be generated by the private-key generator (PKG). Moreover, IBEs can be used for confidential communication, network protocols, digital signatures, etc. In 2001, Boneh and Franklin [2] constructed the first practical IBE scheme under the bilinear Diffe–Hellman (BDH) assumption. Then, Canetti et al. [3] constructed an IBE scheme in the standard model, and they gave the security proof in the selective-ID model. In this model, the adversary must announce the target identity at the beginning. Boneh and Boyen [4] proposed a fully (adaptively) secure IBE scheme. Their scheme is too inefficient to be practical since it requires numerous exponentiation operations and group operations. In the adaptive-ID model, the adversary can announce the target identity after private key queries. In 2005, Waters [5] constructed the first efficient fully secure IBE scheme and showed that a selectively secure scheme can be improved to adaptive security. Furthermore, there are many IBE constructions [6–13] based on pairing or quadratic residues which cannot resist quantum computing.

Lattice-based cryptography has become the focus of research in recent years because it is flexible in construction and resistant to quantum computing. Regev [14] defined the learning with error (LWE) problem and gave a reduction from the worst-case lattice problems. Stehlé [15] and Lyubashevsky [16] defined the ring learning with error (RLWE) problem, which led to new cryptographic applications.

In 2008, Gentry et al. [17] proposed the first LWE-based IBE scheme in the random oracle model. Their scheme relied on the Dual-Regev encryption scheme and became an example of an LWE-based IBE scheme. Agrawal et al. [18] then construct an efficient selectively secure IBE scheme based on LWE problem in the standard model. They also give an adaptively secure IBE scheme, but each bit of a user's identity is associated with a parameter matrix. This always leads to drastic but unfavorable increases in the sizes of the system public parameters. To solve this drawback, Singh et al. [19] constructed efficient adaptively secure (hierarchical) IBE schemes with short parameters using the blocking technique [20,21]. In 2016, Yamada [22] constructed an adaptively secure IBE scheme with short parameters using injective map and homomorphic computation. Zhang et al. [23] proposed an adaptively secure IBE scheme which achieved shorter public parameters, but their scheme only achieved Q-bounded security. In 2017, Yamada [24] constructed new adaptively secure IBE schemes via new partitioning functions, but the public parameters in their scheme are larger than [23]. Moreover, there are many other IBE constructions [25–32] based on the LWE problem.

Compared with the LWE problem, the RLWE problem is more practical in construction because of smaller storage and faster calculation. In particular, we can use fast Fourier transform (FFT) or number theoretic transform (NTT) to accelerate polynomial multiplications. In 2013, Yang et al. [33] construct a selectively secure IBE scheme over ideal lattice in the standard model. Their construction is a ring variant of Agrawal's selective-ID scheme [18]. In 2014, Ducas et al. [34] propose an efficient IBE scheme over Number Theory Research Unit (NTRU) lattice. (NTRU is a ring-based public key cryptosystem, which was proposed by Hoffstein [35] in 1998. The lattice specified in their scheme is often called the NTRU lattice.) Their construction is a NTRU variant of the scheme by [17]. In order to achieve shorter public parameters, Katsumata [36] constructs an adaptively secure IBE scheme over ideal lattice using Yamada's method [22]. In 2018, Bert et al. [37] construct an efficient IBE scheme and give an efficient implementation. Their construction uses the ring-version trapdoor of Micciancio [38] which is efficient and easy to implement. However, their scheme only achieves selective security. Therefore, it is meaningful to construct adaptively secure efficient (H)IBE schemes over ideal lattice with shorter parameters.

Our contribution. In this paper, we first construct an adaptively secure IBE scheme over ideal lattice with short parameters. In the setting of the most existing lattice-based (H)IBE schemes, the public parameters are generally composed of $l+2$ matrices, where l is the bit length of user's identity. Using the blocking technique, we can reduce the number of elements in public parameters from $l+2$ to $l/\beta+2$ where β is a flexible constant. However, this leads to a reduction in the security. We need to increase the lattice modulo q to achieve the same security level as [18], but it causes an increase in computational cost. Therefore, we make a trade-off between storage space and computational cost. For $l = 160$, the size of public parameters can be reduced by almost 89.7% while increasing the computational cost by only 5.2%. When β is set of 20 (resp. 10), the public parameters only contain 10 (resp. 18) vectors. According to our performance analysis, our scheme can achieve shorter public parameters and better computational efficiency. In addition, we use the gadget-based trapdoor as [37,38] which is simple, efficient and smaller in storage than a basis. Finally, we extend our IBE scheme to a hierarchical IBE scheme, and both of them are proved achieving the indistinguishability of ciphertexts against adaptively chosen identity and chosen plaintext attack (IND-ID-CPA) in the standard model.

The rest of this paper is organized as follows. Section 2 is preliminaries. Sections 3 and 4 describe our adaptively secure IBE and HIBE schemes. In Section 5, we analyse the trade-off and compare with other constructions. In Section 6, we summarize this paper.

2. Preliminaries

Notation. In this paper, we use uppercase letters to represent matrix (i.e., A), and lowercase letters to represent constant or polynomial (i.e., l or u). We use uppercase bold letters to represent polynomial matrices (i.e., $\boldsymbol{R}$), and lowercase bold letters to represent polynomial vectors (i.e., $\boldsymbol{a}$). We use negligible function to represent the function $\epsilon(n)$ which is less than all polynomial fractions for sufficiently large n. We use overwhelming probability to indicate that the event happens with probability $1-\epsilon(n)$.

2.1. IBE and Hierarchical IBE

HIBE system contains four algorithms [7,8]. For identity $id = (id_1, \cdots, id_l)$, we describe the HIBE system as follows.

Setup(d, λ): On input a security parameter λ and a maximum depth d, the algorithm outputs the public parameters PP and master key MK.

Derive$(PP, id|id_l, SK_{id|id_{l-1}}, MK)$: On input public parameters PP, master key MK, identity $id|id_l$ at depth l, and private key $SK_{id|id_{l-1}}$ at depth $l-1$, it outputs the private key $SK_{id|id_l}$ at depth l.

Encrypt$(PP, \mu, id|id_l)$: On input public parameters PP, an identity $id|id_l$ at depth l and a message μ, the algorithm outputs a ciphertext CT.

Decrypt$(PP, CT, SK_{id|id_l})$: On input public parameters PP, a ciphertext CT and a private key $SK_{id|id_l}$, the algorithm outputs the message μ.

IBE system is the same as above HIBE system when $d = 1$. Compared with HIBE, there is an algorithm *Extract* instead of algorithm *Derive*. The algorithm *Extract* inputs public parameters PP, identity id, master key MK, and it outputs the corresponding private key SK_{id}.

Security Game. We use an indistinguishable from random game to define the adaptive security of (H)IBE, which means that adversary can not distinguish between challenge ciphertext and random ciphertext. Let $\mathcal{M}_\lambda$ and $\mathcal{C}_\lambda$ be the message space and ciphertext space where λ is a security parameter. For a maximum depth d, the following defines the game.

Setup: The challenger runs algorithm Setup(d, λ) and sends the public parameters PP to the adversary.

Phase 1: The adversary performs private key queries $q_1, \cdots, q_m$, and the event q_i corresponds to the identity id_i. The challenger runs algorithm Extract to generate the private key sk_i corresponding to id_i and sends it to the adversary.

Challenge: The adversary submits a plaintext $M \in \mathcal{M}_\lambda$ and a target identity id^* which can not appear in Phase 1. Then the challenger chooses a random bit $r \in \{0, 1\}$ and a random ciphertext $C \in \mathcal{C}_\lambda$. If $r = 0$, the challenger sets the challenge ciphertext $C^* :=$ Encrypt (PP, M, id^*). Otherwise, it sets the challenge ciphertext $C^* = C$. The challenger sends C^* to the adversary.

Phase 2: The adversary performs adaptive queries $q_{m+1}, \cdots, q_n$. The event q_i corresponds to the identity id_i which can not be id^*. The challenger responds as in Phase 1.

Guess: The adversary outputs a guess $r' \in \{0, 1\}$, and wins if $r' = r$.

The adversary $\mathcal{A}$ described above is a IND-ID-CPA attacker. We define the advantage of $\mathcal{A}$ as

$$Adv_{d,\varepsilon,\mathcal{A}}(\lambda) = |Pr[r' = r] - \tfrac{1}{2}|$$

Definition 1. *If for all IND-ID-CPA attackers $\mathcal{A}$, the advantage $Adv_{d,\varepsilon,\mathcal{A}}(\lambda)$ is a negligible function, then the HIBE scheme ε is IND-ID-CPA security. The security model of IBE is the same as above model with $d = 1$.*

The following Definition 2 defines the abort-resistant hash functions [18,19], which is used in our security proof.

Definition 2 ([18,19])**.** *Let $\mathcal{H} := \{H : X \to Y\}$ be a family of hash functions and Y contains element 0. For a set $\bar{x} = \{x_0, x_1, \cdots, x_Q\} \in X^{Q+1}$ with $x_0 \notin \{x_1, \cdots, x_Q\}$, we define the non-abort probability of $\bar{x}$*

$$\alpha(\bar{x}) := Pr[H(x_0) = 0 \wedge H(x_1) \neq 0 \wedge \cdots \wedge H(x_Q) \neq 0]$$

where the probability is over the random choice of H in $\mathcal{H}$. For $\alpha(\bar{x}) \in [\alpha_{min}, \alpha_{max}]$, the hash family $\mathcal{H}$ is $(Q, \alpha_{min}, \alpha_{max})$ abort-resistant.

We use the abort-resistant hash family similar to [5,18]. Let q be a prime and $(Z_q^{l'})^* := Z_q^{l'} \backslash \{0^{l'}\}$; we define the hash family $\mathcal{H}_{\text{Wat}} : \left\{ H_h : (Z_q^{l'})^* \to Z_q \right\}$ as $H_h(id) := 1 + \sum_{i=1}^{l'} h_i b_i \in Z_q$ where $id = (b_1, \ldots, b_{l'}) \in (Z_q^{l'})^*$ and $h = (h_1, \ldots, h_{l'}) \in Z_q^{l'}$.

2.2. Integer Lattice and Ideal Lattice

Definition 3. *Let q be a prime, $A \in Z_q^{n \times m}$ and $u \in Z_q^n$; we define integer lattice as:*

$$\Lambda_q(A) := \{e \in Z^m \ s.t. \ \exists s \in Z_q^n \ where \ A^\top s = e \mod q\}$$
$$\Lambda_q^\perp(A) := \{e \in Z^m \ s.t. \ Ae = 0 \mod q\}$$
$$\Lambda_q^u(A) := \{e \in Z^m \ s.t. \ Ae = u \mod q\}$$

Ideal Lattice. Let n be a power of 2; we define the modular polynomial $f(x) = x^n + 1$. Then, we define the ring polynomial R as $R = Z[x]/f(x)$. For a modulus q, we define the ring polynomial R_q as $R_q = Z_q[x]/f(x)$. Therefore, elements in R_q are polynomials with coefficients less than q. The following definition from [16,37] defines the Decision RLWE problem.

Definition 4 (Decision RLWE)**.** *Given a vector of m uniformly random polynomials $\boldsymbol{a} = (a_1, \cdots, a_m)^\top \in R_q^m$, and $\boldsymbol{b} = \boldsymbol{a}s + e$ where $s \in R_q$ and $e \in D_{R^m,\sigma}$. Then, distinguish $(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{a}s + e)$ from uniform $(\boldsymbol{a}, \boldsymbol{b})$.*

Similar to [18], we use $\|\tilde{S}\|$ to denote the Gram–Schmidt norm of S where $S = \{s_1, \cdots, s_k\}$ in $\mathbb{R}^m$. We use $D_{L,\sigma,c}$ to denote the discrete Gaussian distribution with center c and parameter σ over a set L. Moreover, the following theorem from [18,39] defines an algorithm *ExtendBasis* which is used in our HIBE construction.

Theorem 1 ([18,39])**.** *Let $A_i \in Z_q^{n \times m_i}$ where $i = 1, 2, 3$, and $A := (A_1|A_2|A_3)$. We define the algorithm $ExtendBasis(A_1, A_2, A_3, T_2)$ which outputs a basis T_A of $\Lambda_q^\perp(A)$ where T_2 is a basis of $\Lambda_q^\perp(A_2)$.*

2.3. Trapdoors on Lattice

Our constructions require the notion of trapdoor which is first introduced by Ajtai [40]. For a short basis T_A of $\Lambda_q^\perp(A)$, we can get short vectors in $\Lambda_q^\perp(A)$ from a Gaussian distribution. We use the g-trapdoor introduced by Micciancio [38] and the following definition from [37] defines the ring variant of the g-trapdoor.

Definition 5 (g-trapdoor)**.** *For $k = \lceil \log_2 q \rceil$, $m > k$, let $\boldsymbol{a}$ be a vector in R_q^m and $\boldsymbol{g}$ be a vector in R_q^k. The g-trapdoor for $\boldsymbol{a}$ is a polynomial matrix $\boldsymbol{T_a}$ in $R^{(m-k)\times k}$ following a discrete Gaussian distribution of parameter σ, and satisfying $\boldsymbol{a}^\top \begin{pmatrix} \boldsymbol{T_a} \\ I_k \end{pmatrix} = h\boldsymbol{g}^\top$ for some invertible element $h \in R_q$. The polynomial h is the tag associated to trapdoor $\boldsymbol{T_a}$.*

In our construction, we need a trapdoor generation algorithm (*TrapGen*) and preimage sampling algorithm (*SamplePre*) from [37], and both of them are described as follows.

Algorithm *TrapGen* inputs a modulus q, a Gaussian parameter σ, a polynomial vector $\boldsymbol{a}' \in R_q^{m-k}$ and a polynomial $h \in R_q$. It returns a polynomial vector $\boldsymbol{a} \in R_q^m$, a trapdoor $\boldsymbol{T_a} \in R^{(m-k)\times k}$ with tag h. We use vector $\boldsymbol{a}'$, gadget vector $\boldsymbol{g}$ and trapdoor $\boldsymbol{T_a}$ to construct the target vector $\boldsymbol{a}$. The trapdoor $\boldsymbol{T_a}$ is choosing from a gaussian distribution with parameter σ. In our construction, the target vector $\boldsymbol{a}$ is part of public parameter and the trapdoor $\boldsymbol{T_a}$ is the master key.

Algorithm *SamplePre* inputs a vector $\boldsymbol{a} \in R_q^m$, a trapdoor $\boldsymbol{T_a} \in R^{(m-k)\times k}$ with tag $h \in R_q$, a polynomial $u \in R_q$ and a Gaussian parameter σ. It returns a vector $\boldsymbol{x} \in R_q^m$ following a discrete Gaussian distribution of parameter ξ, and satisfying $\boldsymbol{a}^\top \boldsymbol{x} = u$. To find a vector $\boldsymbol{x}$ satisfing $\boldsymbol{a}^\top \boldsymbol{x} = u$, we need to find a vector $\boldsymbol{z}$ that satisfies $\boldsymbol{g}^\top \boldsymbol{z} = h^{-1} \cdot (u - \boldsymbol{a}^\top \boldsymbol{p})$ where $\boldsymbol{p}$ is a perturbation vector. Then, we get $\boldsymbol{x} = \boldsymbol{p} + \begin{pmatrix} \boldsymbol{T_a} \\ I_k \end{pmatrix} \boldsymbol{z}$ such that $\boldsymbol{a}^\top \boldsymbol{x} = \boldsymbol{a}^\top \boldsymbol{p} + \boldsymbol{a}^\top \begin{pmatrix} \boldsymbol{T_a} \\ I_k \end{pmatrix} \boldsymbol{z} = \boldsymbol{a}^\top \boldsymbol{p} + h\boldsymbol{g}^\top \boldsymbol{z} = \boldsymbol{a}^\top \boldsymbol{p} + h \cdot h^{-1}(u - \boldsymbol{a}^\top \boldsymbol{p}) = u$. In our construction, the target vector $\boldsymbol{x}$ is used to construct the private keys.

2.4. Sampling Algorithms

Our constructions require a vector of form $\boldsymbol{f} = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{R}^\top \boldsymbol{a} + \boldsymbol{b} \end{pmatrix} \in R_q^{2m}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors in R_q^m. Matrix $\boldsymbol{R} \in R^{m\times m}$ consists of polynomials with coefficients $\{1, -1\}$. We can get the private key by sampling short vectors in $\Lambda_q^u(\boldsymbol{f})$ for some $u \in R_q$. Algorithm $SampleLeft$ is used in our construction and algorithm *SampleRight* is used in our security proof.

Algorithm $SampleLeft$ needs a vector of form $\boldsymbol{f}_1 := \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{m}_1 \end{pmatrix}$. It inputs a trapdoor $\boldsymbol{T_a}$ of $\Lambda_q^\perp(\boldsymbol{a})$ and returns a short vector $\boldsymbol{s} \in \Lambda_q^u(\boldsymbol{f}_1)$. The description of SampleLeft is shown in Algorithm 1. By algorithm *SamplePre* and 1, we have $\boldsymbol{a}^\top \boldsymbol{s}_1 = u - \boldsymbol{m}_1^\top \boldsymbol{s}_2$. Then, $\boldsymbol{f}_1^\top \boldsymbol{s} = \boldsymbol{a}^\top \boldsymbol{s}_1 + \boldsymbol{m}_1^\top \boldsymbol{s}_2 = u - \boldsymbol{m}_1^\top \boldsymbol{s}_2 + \boldsymbol{m}_1^\top \boldsymbol{s}_2 = u$. Therefore, we get a short vector $\boldsymbol{s} \in R^{m+m_1}$ distributed statistical close to $D_{\Lambda_q^u(\boldsymbol{f}_1),\sigma}$.

Algorithm 1 SampleLeft$(\boldsymbol{a}, \boldsymbol{m}_1, \boldsymbol{T_a}, u, \sigma)$.

Input: Polynomial vectors $\boldsymbol{a} \in R_q^m$ and $\boldsymbol{m}_1 \in R_q^{m_1}$, a trapdoor $\boldsymbol{T_a}$ of $\Lambda_q^\perp(\boldsymbol{a})$, a polynomial $u \in R_q$ and a Gaussian parameter σ;

Output: A short vector $\boldsymbol{s} \in R_q^{m+m_1}$ following the Gaussian distribution $D_{\Lambda_q^u(\boldsymbol{f}_1),\sigma}$ with $\boldsymbol{f}_1 := \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{m}_1 \end{pmatrix}$.

1: Sample a random vector $\boldsymbol{s}_2 \leftarrow D_{R^{m_1},\sigma}$;
2: Sample $\boldsymbol{s}_1 \leftarrow SamplePre(\boldsymbol{a}, \boldsymbol{T_a}, y, \sigma)$, where $y = u - \boldsymbol{m}_1^\top \boldsymbol{s}_2 \in R_q$;
3: **return** $\boldsymbol{s} \leftarrow (\boldsymbol{s}_1, \boldsymbol{s}_2) \in R^{m+m_1}$.

Algorithm *SampleRight* needs a vector of form $\boldsymbol{f}_2 := \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{R}^\top \boldsymbol{a} + \boldsymbol{b} \end{pmatrix}$. It inputs a trapdoor $\boldsymbol{T_b}$ of $\Lambda_q^\perp(\boldsymbol{b})$ and returns a short vector $\boldsymbol{s} \in \Lambda_q^\perp(\boldsymbol{f}_2)$. The description of SampleRight is shown in Algorithm 2. In HIBE, we also need an algorithm *ExtendBasis* which is similar to Theorem 1. By algorithm *SamplePre* and 2, we have $\boldsymbol{f}_2 \boldsymbol{s} = u$ and then we get a short vector $\boldsymbol{s} \in R_q^{m+k}$ distributed statistically close to $D_{\Lambda_q^u(\boldsymbol{f}_2),\sigma}$.

Algorithm 2 SampleRight$(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{T_b}, u, \sigma)$.

Input: Polynomial vectors $\boldsymbol{a} \in R_q^k$ and $\boldsymbol{b} \in R_q^m$, a matrix of polynomial $\boldsymbol{R} \in R_q^{k\times m}$, a trapdoor $\boldsymbol{T_b}$ of $\Lambda_q^\perp(\boldsymbol{b})$, a polynomial $u \in R_q$ and a Gaussian parameter σ;

Output: A short vector $\boldsymbol{s} \in R_q^{m+k}$ following the Gaussian distribution $D_{\Lambda_q^u(\boldsymbol{f}_2),\sigma}$ with $\boldsymbol{f}_2 := \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{R}^\top \boldsymbol{a} + \boldsymbol{b} \end{pmatrix}$.

1: Select $m + k$ linearly indepndent vectors in $\Lambda_q^\perp(\boldsymbol{f}_2)$ and construct $\boldsymbol{T}_{\boldsymbol{f}_2}$;
2: Convert $\boldsymbol{T}_{\boldsymbol{f}_2}$ into a basis $\boldsymbol{T}'_{\boldsymbol{f}_2}$ of $\Lambda_q^\perp(\boldsymbol{f}_2)$ where $\|\widetilde{\boldsymbol{T}_{\boldsymbol{f}_2}}\| = \|\widetilde{\boldsymbol{T}'_{\boldsymbol{f}_2}}\|$;
3: Sample $\boldsymbol{s} \leftarrow SamplePre(\boldsymbol{f}_2, \boldsymbol{T}'_{\boldsymbol{f}_2}, u, \sigma)$;
4: **return** $\boldsymbol{s} \in \Lambda_q^u(\boldsymbol{f}_2)$.

3. Adaptively Secure IBE

Agrawal [18] converted their selectively secure IBE to an adaptively secure IBE using the technique of Waters [5]. Though the private key size and ciphertext size are the same, the size of the public parameters is too large. In this section, we construct an adaptively security IBE over ideal lattice and reduce the size of the public parameters using the blocking technique.

3.1. The IBE Construction

The identity id is an l bits string in $\{0,1\}^l$. We divide id into l' segments $(b_1, b_2, \cdots, b_{l'})$, where b_i is a $l/l' = \beta$ bits string. Then, we describe our IBE construction as follows.

Setup(λ): On input a security parameter λ and other parameters q, n, m, σ, α, do:

1. Run $(\boldsymbol{a}_0, \boldsymbol{T}_{\boldsymbol{a}_0}) \leftarrow TrapGen(q, n)$, where $\boldsymbol{a}_0$ is a vector in R_q^m with a trapdoor $\boldsymbol{T}_{\boldsymbol{a}_0} \in R_q^{(m-k)\times k}$;
2. Select $l'+1$ uniformly random vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{l'}, \boldsymbol{b} \in R_q^m$, and these vectors are used to form the public parameters;
3. Select a uniformly random polynomial $u \in R_q$;
4. Output the public parameters $PP = (\boldsymbol{a}_0, \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{l'}, \boldsymbol{b}, u)$ and master key $MK = (\boldsymbol{T}_{\boldsymbol{a}_0})$.

Extract(PP, MK, id): On input public parameters PP, master key MK and identity $id = (b_1, b_2, \cdots, b_{l'})$, do:

1. Set $\boldsymbol{a}_{id} = \boldsymbol{b} + \sum_{i=1}^{l'} b_i \cdot \boldsymbol{a}_i \in R_q^m$ and $\boldsymbol{f} = \begin{pmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_{id} \end{pmatrix} \in R_q^{2m}$. They are used to generate the private key;
2. Run $\boldsymbol{s} \leftarrow SampleLeft(\boldsymbol{a}_0, \boldsymbol{a}_{id}, \boldsymbol{T}_{\boldsymbol{a}_0}, u, \sigma)$, where $\boldsymbol{s}$ is a vector in R_q^{2m};
3. Output the private key $SK = \boldsymbol{s} \in R_q^{2m}$.

Encrypt(PP, id, m): On input public parameters PP, an identity $id = (b_1, b_2, \cdots, b_{l'})$, and a message $\mu \in \{0,1\}^n$, do:

1. Set $\boldsymbol{a}_{id} = \boldsymbol{b} + \sum_{i=1}^{l'} b_i \cdot \boldsymbol{a}_i \in R_q^m$ and $\boldsymbol{f} = \begin{pmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_{id} \end{pmatrix} \in R_q^{2m}$. They are used to generate the ciphertext;
2. Select a uniformly random polynomial $t \in R_q$;
3. Select l' matrices $\boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_{l'}$ in $R^{m\times m}$ which consist of uniformly random polynomials with coefficient $\{1, -1\}$. Define $\boldsymbol{R}_{id} = \sum_{i=1}^{l'} b_i \boldsymbol{R}_i$ and its coefficients are in $\{-l'(2^\beta - 1), l'(2^\beta - 1)\}$;
4. Select noise polynomial $x \leftarrow D_{R_q,\sigma}$, noise vector $\boldsymbol{y} \leftarrow D_{R_q^m,\sigma}$ and set $\boldsymbol{z} \leftarrow \boldsymbol{R}_{id}^\top \cdot \boldsymbol{y} \in R_q^m$;
5. Set $c_0 = u \cdot t + x + \mu \cdot \lfloor q/2 \rfloor \in R_q$, and $\boldsymbol{c}_1 = \boldsymbol{f} \cdot t + \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} \in R_q^{2m}$;
6. Output the ciphertext $CT = (c_0, \boldsymbol{c}_1) \in R_q \times R_q^{2m}$.

Decrypt(PP, SK, CT): On input public parameters PP, a private key $SK = \boldsymbol{s}$, and a ciphertext $CT = (c_0, \boldsymbol{c}_1)$, do:

1. Compute $w = c_0 - \boldsymbol{s}^\top \cdot \boldsymbol{c}_1 \in R_q$, and w_i denotes the coefficient of w;
2. Compare w_i and $\lfloor q/2 \rfloor$ treating them as integer in Z, if $|w - \lfloor q/2 \rfloor| < \lfloor q/4 \rfloor$, output 1, otherwise output 0.

3.2. Parameters and Correctness

In this section, we prove the correctness of the above IBE scheme. During decryption, we have

$$\begin{aligned} w &= c_0 - \boldsymbol{s}^\top \cdot \boldsymbol{c}_1 \\ &= u \cdot t + x + \mu \cdot \lfloor q/2 \rfloor - \boldsymbol{s}^\top (\boldsymbol{f} \cdot t + \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix}) \\ &= \mu \cdot \lfloor q/2 \rfloor + \underbrace{x - \boldsymbol{s}^\top \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix}}_{error\ term} \end{aligned} \tag{1}$$

In order to decrypt correctly, the error term $x - \boldsymbol{s}^\top \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix}$ should be bounded by $\lfloor q/4 \rfloor$. Then, we need the following two lemmas to analyze the error rate of decryption.

Lemma 1 ([41]). *Let $c \geq 1$, $C = c \cdot exp(\frac{1-c^2}{2}) < 1$ and $x \leftarrow D_{Z^n,s}$; then, for any real $s > 0$ and any integer $n \geq 1$, we have*

$$Pr\left[\|x\| \geq cs\sqrt{n/2\pi}\right] \leq C^n \tag{2}$$

Lemma 2 ([42]). *For any real $s > 0$, $T > 0$, and any $x \in R^n$, we have*

$$Pr\left[|< x, D_{Z^n,s} >| \geq Ts\|x\|\right] < 2exp(-\pi T^2) \tag{3}$$

Theorem 2. *Let $q \leq 4[l'(2^\beta - 1)\sqrt{mn} + 1]\delta c\sigma\sqrt{mn/2\pi}$, $c \geq 1$, $t > 15$, the above IBE scheme decrypts correctly with overwhelming probability.*

Proof of Theorem 2. Letting $\boldsymbol{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$ with $\boldsymbol{s}_1, \boldsymbol{s}_2 \in R^m$, we have $\boldsymbol{s}^\top \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{s}_1^\top \cdot \boldsymbol{y} + \boldsymbol{s}_2^\top \cdot \boldsymbol{z}$. Since $\boldsymbol{z} = \boldsymbol{R}_{id}^\top \cdot \boldsymbol{y}$, we have $\|\boldsymbol{z}\| = \|\boldsymbol{R}_{id} \cdot \boldsymbol{y}\| \leq \|\boldsymbol{R}_{id}\| \cdot \|\boldsymbol{y}\| = l'(2^\beta - 1)\sqrt{mn}\|\boldsymbol{y}\|$.

Similar to [33], we compute the decryption error rate with Lemma 2 as

$$\begin{aligned} &Pr\left[(l'(2^\beta - 1)\sqrt{mn} + 1)\sqrt{m}|< x, y >| \geq q/4\right] \\ &= Pr\left[|< x, y >| \geq q/(4(l'(2^\beta - 1)\sqrt{mn} + 1)\sqrt{m})\right] \\ &= Pr\left[|< x, y >| \geq T\delta\|x\|\right] \\ &< 2exp(-\pi T^2) \end{aligned} \tag{4}$$

For $c \geq 1$, we have $\|x\| \leq c\sigma\sqrt{n/2\pi}$ with Lemma 1. Then,

$$T = \frac{q}{4[l'(2^\beta - 1)\sqrt{mn} + 1]\sqrt{m}\delta\|x\|} \geq \frac{q}{4[l'(2^\beta - 1)\sqrt{mn} + 1]\delta c\sigma\sqrt{mn/2\pi}} \tag{5}$$

When T is sufficiently large, the decryption error rate $2exp(-\pi T^2)$ is a negligible function, and we can decrypt correctly with overwhelming probability. □

Similar to [18,19,37], we need to set the parameters as follows:

- the error term is less than $q/4$ (*i.e.* $q \leq 4[l'(2^\beta - 1)\sqrt{mn} + 1]\delta c\sigma\sqrt{mn/2\pi}$),
- that algorithm TrapGen can operate (*i.e.* $m = O(n \log q)$),
- that σ is sufficiently large for sampling algorithm (i.e., $\sigma > \|\widetilde{T_B}\|2^\beta l'\sqrt{m}\omega\sqrt{\log m} = 2^\beta l'\sqrt{m}\omega\sqrt{\log m}$),
- that reduction applies (i.e., the number of private key queries $Q \leq \frac{q}{2}$).

3.3. Security Proof

In this section, we give the security proof of our IBE scheme. We describe the definition of abort-resistant hash functions in Definition 2.

Lemma 3. *Let q be a prime, the hash family $\mathcal{H}_{Wat}$ is $(Q, \frac{1}{q}(1 - \frac{Q}{q}), \frac{1}{q})$ abort-resistant where $0 < Q < q$.*

Proof of Lemma 3. Let $\bar{id}$ be a set of $(id_0, id_1, \cdots, id_Q)$ where $id_0 \notin \{id_1, \cdots, id_Q\}$. For $i = 0, \cdots, Q + 1$, S_i denotes the set of functions $H(id_i) = 0$ in $\mathcal{H}_{Wat}$. We have $|S_i| = q^{l'-1}$ and $|S_0 \cap S_j| \leq q^{l'-2}$ with $j > 0$. For $i = 1, \cdots, Q$, the set of $H(id_0) = 0$ and $H(id_i) \neq 0$ is defined as $S := S_0 \setminus (S_1 \cup \cdots \cup S_Q)$. Then, we have

$$|S| = |S_0 \setminus (S_1 \cup \cdots \cup S_Q)| \geq |S_0| - \textstyle\sum_{i=1}^{Q} |S_0 \cap S_i| \geq q^{l'-1} - Qq^{l'-2}$$

The non-abort probability of $\bar{id}$ is $|S|/q^{l'} \geq \frac{1}{q}(1 - \frac{Q}{q})$. Since $|S| \leq |S_0|$, the no-abort probability is $|S|/q^{l'} \leq |S_0|/q^{l'} \leq \frac{1}{q}$ at most. □

Theorem 3. *The IBE system with parameters (n, m, q, σ) is IND-ID-CPA secure in the standard model under the hardness of RLWE.*

Proof of Theorem 3. The proof proceeds in a sequence of games, and the first game is the same as the security game in Definition 1. In game i, we use W_i to denote that the adversary guesses the challenge message correctly. Then, the advantage of adversary in game i is $|Pr[W_i] - \frac{1}{2}|$.

Game 0. The original IND-ID-CPA game between an adversary $\mathcal{A}$ and a challenger.

Game 1. The challenger builds the public parameters $PP = (\boldsymbol{a}_0, \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{l'}, \boldsymbol{b}, u)$ in the original game. These vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{l'}, \boldsymbol{b}$ are chosen uniformly from R_q^m. The Game 1 challenger chooses l' random matrices $\boldsymbol{R}_i^* \in R^{m \times m}$ and random polynomials $h_i \in Z_q$ at the setup phase. Matrix $\boldsymbol{R}_i^*$ consists of uniformly random polynomials with coefficient $\{-1.1\}$. Then the challenger generates vectors $\boldsymbol{a}_0$ and $\boldsymbol{b}$ as in original game, and constructs vector $\boldsymbol{a}_i$ as

$$\boldsymbol{a}_i \leftarrow (\boldsymbol{R}_i^*)^\top \cdot \boldsymbol{a}_0 - h_i \cdot \boldsymbol{b} \in R_q^m,\ i \in [1, l']$$

The matrix $\boldsymbol{R}_i^*$ is used to build vector $\boldsymbol{a}_i$ and challenge ciphertext CT^* (i.e. $\boldsymbol{z} \leftarrow (\boldsymbol{R}_{id}^*)^\top y \in R_q^m$ where $\boldsymbol{R}_{id}^* = \Sigma_{i=1}^{l'} b_i^* \cdot \boldsymbol{R}_i^* \in R^{m \times m}$). Set $\boldsymbol{R}^* := (\boldsymbol{R}_1^*, \boldsymbol{R}_2^*, \cdots, \boldsymbol{R}_{l'}^*)$, the distributions

$$(\boldsymbol{a}_0\ ,\ \boldsymbol{a}_0^\top \cdot \boldsymbol{R^*}\ ,\ (\boldsymbol{R}^*)^\top \boldsymbol{y}) \text{ and } (\boldsymbol{a}_0\ ,\ ((\boldsymbol{a}_1')^\top | \cdots | (\boldsymbol{a}_{l'}')^\top)\ ,\ (\boldsymbol{R}^*)^\top \boldsymbol{y})$$

are statistically close. The vectors $\boldsymbol{a}_i'$ are uniformly random elements in R_q^m. For $\boldsymbol{z} \leftarrow (\boldsymbol{R}_{id}^*)^\top \cdot \boldsymbol{y}$, the distributions

$$(\boldsymbol{a}_0\ ,\ \boldsymbol{a}_0^\top \cdot \boldsymbol{R_1^*}\ , \cdots,\ \boldsymbol{a}_0^\top \cdot \boldsymbol{R_{l'}^*}\ ,\ \boldsymbol{z}) \text{ and } (\boldsymbol{a}_0\ ,\ (\boldsymbol{a}_1')^\top\ , \cdots,\ (\boldsymbol{a}_{l'}')^\top\ ,\ \boldsymbol{z})$$

are statistically close. In adversary's view, the vectors $\boldsymbol{a}_0^\top \cdot \boldsymbol{R_i^*}$ are statistically close to uniformly random elements $(\boldsymbol{a}_i')^\top$ and independent of vector $\boldsymbol{z}$. Therefore, in adversary's view, the vector $\boldsymbol{a}_i$ are uniformly random vectors as in Game 0. This shows that

$$Pr[W_0] = Pr[W_1] \tag{6}$$

Game 2. In Game 2, we add an abort event and the rest is the same as Game 1. We use the abort-resistant $\mathcal{H}_{Wat}$ introduced in Lemma 3. In the Setup phase, the challenger chooses a function $H \in \mathcal{H}_{Wat}$ and reserves it to itself. Then, the challenger answers key queries and sends challenge ciphertext to adversary as in Game 1. We use $id_1, \cdots, id_Q$ to denote the identities that the adversary queries. We use id^* to denote the challenge identity which is not in $\{id_1, \cdots, id_Q\}$. In the Guess phase, the adversary returns a guess $r' \in \{0, 1\}$. Then, the challenger performs as follows:

1. **Abort check** [18]: For $i = 1, \cdots, Q$, the game proceeds normally if $H(id^*) = 0$ and $H(id_i) \neq 0$. Otherwise, it resets r' and aborts the game. However, the game proceeds normally in the adversary's view.
2. **Artificial abort** [5,18]: The challenger chooses a bit $\Gamma \in \{0, 1\}$ such that $Pr[\Gamma = 1] = \gamma(I)$. If there is no abort $\gamma(I) = 0$, otherwise, $\gamma(I) = 1$. If $\Gamma = 1$ or $\gamma(I) = 1$, the challenger resets r' and aborts the game.

For identities $I = (id^*, id_1, \cdots, id_Q)$, we use $\epsilon(I)$ to denote the probability of non-abort when the adversary performs these private key queries. Moreover, we use ϵ_{max} and ϵ_{min} to denote the maximum and minimum of $\epsilon(I)$.

Lemma 4 ([18]). *For $i = 1, 2$, let W_i be the event that the adversary wins the Game i. Then,*

$$\left|\Pr[W_2] - \tfrac{1}{2}\right| \geq \epsilon_{\min} \left|\Pr[W_1] - \tfrac{1}{2}\right| - \tfrac{1}{2}(\epsilon_{\max} - \epsilon_{\min})$$

According to [18], they show that $\epsilon_{max} - \epsilon_{min}$ is less than $\epsilon_{\min} \left| \Pr[W_1] - \frac{1}{2} \right|$. Since $q \geq 2Q$, we have $\epsilon_{min} = \frac{1}{q}(1 - \frac{Q}{q}) \geq \frac{1}{2q}$. Then,

$$\left| \Pr[W_2] - \frac{1}{2} \right| \geq \frac{1}{2}\epsilon_{\min} \left| \Pr[W_1] - \frac{1}{2} \right| \geq \frac{1}{4q} \left| \Pr[W_1] - \frac{1}{2} \right| \tag{7}$$

Game 3. In Game 3, we change the method of generating $\boldsymbol{a}_0$ and $\boldsymbol{b}$ in PP. Vector $\boldsymbol{a}_0$ is generated as a random element in R_q^m and vector $\boldsymbol{b}$ is generated by algorithm TrapGen. The challenger also gets a trapdoor $T_{\boldsymbol{b}}$ of $\Lambda_q^{\perp}(\boldsymbol{b})$. The construction $\boldsymbol{a}_i \leftarrow (\boldsymbol{R}_i^*)^{\top} \cdot \boldsymbol{a}_0 - h_i \cdot \boldsymbol{b} \in R_q^m$ is the same as in Game 2. To answer the private key query of $id = (b_1, b_2, \cdots, b_{l'})$, the challenger generates the corresponding private key $SK_{id} = \boldsymbol{s}$ from $\Lambda_q^u(\boldsymbol{f}_{id})$. Let

$$\boldsymbol{f}_{id} := \begin{pmatrix} \boldsymbol{a}_0 \\ \boldsymbol{b} + \Sigma_{i=1}^{l'} b_i \cdot \boldsymbol{a}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{a}_0 \\ (\boldsymbol{R}_{id})^{\top} \cdot \boldsymbol{a}_0 - h_{id} \cdot \boldsymbol{b} \end{pmatrix} \tag{8}$$

where $\boldsymbol{R}_{id} = \Sigma_{i=1}^{l'} b_i \cdot \boldsymbol{R}_i^* \in R_q^{m \times m}$ and $h_{id} = 1 + \Sigma_{i=1}^{l'} b_i \cdot h_i \in Z_q$. If $h_{id} = 0$, the challenger abort the game as in Game 2. Otherwise, the challenger gets $\boldsymbol{s} \leftarrow SampleRight(\boldsymbol{a}_0, h_{id} \cdot \boldsymbol{b}, \boldsymbol{R}_{id}, T_{\boldsymbol{b}}, u, \sigma) \in R_q^{2m}$. Then, it sends $SK_{id} = \boldsymbol{s}$ to adversary $\mathcal{A}$.

In adversary's view, Game 2 and Game 3 are indistinguishable. Therefore,

$$Pr[W_2] = Pr[W_3] \tag{9}$$

Game 4. The challenge ciphertext $(c_0^*, \boldsymbol{c}_1^*)$ is randomly selected in $R_q \times R_q^{2m}$ and the rest is the same as in Game 3, so the advantage of $\mathcal{A}$ is 0 in Game 4. Then, we need to prove that Game 3 and Game 4 are computationally indistinguishable.

Suppose there is an adversary $\mathcal{A}$ who has non-negligible probability in distinguishing Game 3 and Game 4. Then, we constructs an RLWE algorithm $\mathcal{B}$.

An instance of RLWE problem is provided as a sample oracle $\mathcal{O}$. We use $\mathcal{O}_\$$ to denote a truly random oracle. For a random $s \in R_q$, we use $\mathcal{O}_s$ to denote a noisy pseudo-random oracle.

Instance. For $i = 0, \cdots, m$, $\mathcal{B}$ requests from $\mathcal{O}$ and gets RLWE samples $(u_i, v_i) \in R_q \times R_q$.

Setup. $\mathcal{B}$ generates the public parameters:

1. Construct random vector $\boldsymbol{a}_0 \in R_q^m$ with RLWE samples. For $i = 1, \cdots, m$, the i-th column of $\boldsymbol{a}_0$ is u_i.
2. Let the random polynomial $u_0 \in R_q$ be the 0-th RLWE sample.
3. Construct vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}$ as in Game 3.
4. Send public parameters $PP = (\boldsymbol{a}_0, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_{l'}, \boldsymbol{b}, u_0)$ to adversary $\mathcal{A}$.

Phase 1 and Phase 2. $\mathcal{B}$ answers private key queries as in Game 3.

Challenge. $\mathcal{A}$ submits a target identity $id^* = (b_1, \cdots, b_{l'})$ and a message $\mu^* \in \{0,1\}^n$. $\mathcal{B}$ prepares a challenge ciphertext for the target identity as follows:

1. Set $v_* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in R_q^m$ with the RLWE instance.
2. Let $c_0^* = v_0 + \mu^* \cdot \lfloor q/2 \rceil \in R_q$ to blind the message bit.
3. Set $\boldsymbol{R}_{id}^* = \Sigma_{i=1}^{l'} b_i \cdot \boldsymbol{R}_i^* \in R_q^{m \times m}$ and $\boldsymbol{c}_1^* = \begin{bmatrix} v^* \\ (\boldsymbol{R}_{id^*})^{\top} \cdot v^* \end{bmatrix} \in R_q^{2m}$.
4. Choose a random bit $r \in \{0,1\}$. If $r = 0$, set $CT^* = (c_0^*, \boldsymbol{c}_1^*)$. Otherwise, select a random element $CT^* = (c_0, \boldsymbol{c}_1)$ in $R_q \times R_q^{2m}$. Then, send challenge ciphertext CT^* to adversary.

Guess. Finally, the adversary $\mathcal{A}$ returns a guess r'. The simulator $\mathcal{B}$ outputs 1 if $r' = r$, otherwise 0.

Analysis. According to [18], the challenge ciphertext is the same as valid ciphertext in game 3 if sampling oracle $\mathcal{O}$ is pseudo-random $\mathcal{O}_s$, and the challenge ciphertext is the same as random ciphertext in game 4 if oracle $\mathcal{O}$ is truly random $\mathcal{O}_\$$. The simulator's advantage in solving RLWE problem is equal to $\mathcal{A}$'s advantage in distinguishing valid ciphertext and random ciphertext. For $Pr[W_4] = \frac{1}{2}$, we get

$$|Pr[W_3] - \frac{1}{2}| = |Pr[W_3] - Pr[W_4]| \leq Adv_{\mathcal{B}}^{RLWE} \tag{10}$$

Then, we have

$$|Pr[W_0] - \frac{1}{2}| \leq 4q \cdot Adv_{\mathcal{B}}^{RLWE} \tag{11}$$

□

4. Adaptively Secure HIBE

We extend our IBE scheme to a hierarchical IBE scheme. Similar to our IBE scheme above, we also use the blocking technique to reduce the size of public parameters.

4.1. The HIBE Construction

The identity $id|id_l$ is composed of l identities id_i at different depth, and it is represented as $id|id_l = (id_1, \cdots, id_l)$ where id_i is a l' bit string. We divide the identity id_i at depth i into l'' segments $(b_{i,1}, \cdots, b_{i,l''})$ where $b_{i,j}$ is a $\beta = l'/l''$ bits string.

Then, we describe our HIBE construction as follows.

Setup(d, λ): On input a security parameter λ, a maximum depth d and other parameters q, n, m, σ, α, do:

1. Run $(\boldsymbol{a}_0, \boldsymbol{T}_{\boldsymbol{a}_0}) \leftarrow TrapGen(q, n)$, where $\boldsymbol{a}_0$ is a vector in R_q^m with a trapdoor $\boldsymbol{T}_{\boldsymbol{a}_0} \in R_q^{(m-k)\times k}$;
2. Choose $l''d + 1$ random vectors $\boldsymbol{a}_{1,1}, \cdots, \boldsymbol{a}_{1,l''}, \cdots, \boldsymbol{a}_{d,1}, \cdots, \boldsymbol{a}_{d,l''}, \boldsymbol{b} \in R_q^m$, and these vectors are used to form the public parameters;
3. Choose a uniformly random polynomial $u \in R_q$;
4. Output the public parameters $PP = (\boldsymbol{a}_0, \boldsymbol{a}_{1,1}, \cdots, \boldsymbol{a}_{1,l''}, \cdots, \boldsymbol{a}_{d,1}, \cdots, \boldsymbol{a}_{d,l''}, \boldsymbol{b}, u)$ and master key $MK = (\boldsymbol{T}_{\boldsymbol{a}_0})$.

Derive$(PP, id|id_l, SK_{id|id_{l-1}})$: On input public parameters PP, an identity $id|id_l$ and a private key $SK_{id|id_{l-1}}$ at depth $l-1$, do:

1. Set $\boldsymbol{f}_{id|id_l} = \begin{pmatrix} \boldsymbol{f}_{id|id_{l-1}} \\ \sum_{i=1}^{l''} \boldsymbol{a}_{l,i}\boldsymbol{b}_{l,i} + \boldsymbol{b} \end{pmatrix} \in R_q^{(l+1)m}$, and it is used to generate the private key;
2. Run $\boldsymbol{s} \leftarrow SampleLeft(\boldsymbol{f}_{id|id_{l-1}}, \sum_{i=1}^{l''} \boldsymbol{a}_{l,i}\boldsymbol{b}_{l,i} + \boldsymbol{b}, SK_{id|id_{l-1}}, \sigma_l)$, where $\boldsymbol{s}$ is a vector in R_q^{2m};
3. Output the private key $SK_{id|id_l} = \boldsymbol{s} \in R_q^{2m}$.

Encrypt(PP, id, m): On input public parameters PP, an identity $id|id_l$ at depth l and a message $\mu \in \{0,1\}^n$, do:

1. Set $\boldsymbol{f}_{id|id_l} = \begin{pmatrix} \boldsymbol{f}_{id|id_{l-1}} \\ \sum_{i=1}^{l''} \boldsymbol{a}_{l,i}\boldsymbol{b}_{l,i} + \boldsymbol{b} \end{pmatrix} \in R_q^{(l+1)m}$, and it is used to generate the ciphertext;
2. Choose a uniformly random polynomial $t \in R_q$;
3. Choose ll'' matrices $\boldsymbol{R}_{i,j} \in R^{m\times m}$ for $i = 1, \cdots, l$ and $j = 1, \cdots, l''$, which consist of random polynomials with coefficient $\{1, -1\}$. Define $\boldsymbol{R}_{id} = \sum_{i=1}^{l''} b_{1,i}\boldsymbol{R}_{1,i}|| \cdots ||\sum_{i=1}^{l''} b_{l,i}\boldsymbol{R}_{l,i} \in R^{m\times lm}$;
4. Choose noise polynomial $x \leftarrow D_{R_q,\sigma}$, noise vector $\boldsymbol{y} \leftarrow D_{R_q^m,\sigma}$, and set $\boldsymbol{z} \leftarrow \boldsymbol{R}_{id}^\top \cdot \boldsymbol{y} \in R_q^{lm}$;

5. Set $c_0 = u \cdot t + x + \mu \cdot \lfloor q/2 \rfloor \in R_q$, and $\boldsymbol{c}_1 = \boldsymbol{f} \cdot t + \begin{bmatrix} y \\ z \end{bmatrix} \in R_q^{(l+1)m}$;
6. Output the ciphertext $CT = (c_0, \boldsymbol{c}_1) \in R_q \times R_q^{(l+1)m}$.

Decrypt$(PP, SK_{id|id_l}, CT)$: On input public parameters PP, a private key $SK_{id|id_l}$ at depth l and a ciphertext $CT = (c_0, \boldsymbol{c}_1)$, do:

1. Set $\tau_l := \sigma_l \sqrt{m(l+1)} w \sqrt{\log(lm)}$;
2. Sample $\boldsymbol{s}_{id} \leftarrow SamplePre(\boldsymbol{f}_{id|id_l}, SK_{id|id_l}, u, \tau_l)$ such that $\boldsymbol{f}_{id} \cdot \boldsymbol{s}_{id} = u$;
3. Compute $w = c_0 - \boldsymbol{s}_{id}^\top \cdot \boldsymbol{c}_1 \in R_q$, w_i denotes the coefficient of w;
4. Compare w_i and $\lfloor q/2 \rfloor$ treating them as integer in Z, if $|w_i - \lfloor q/2 \rfloor| < \lfloor q/4 \rfloor$, output 1, otherwise output 0.

4.2. Parameters and Correctness

In this section, we prove the correctness of the above HIBE scheme. During decryption, we have

$$\begin{aligned} w &= c_0 - \boldsymbol{s}_{id}^\top \cdot \boldsymbol{c}_1 \\ &= u \cdot t + x + \mu \cdot \lfloor q/2 \rfloor - \boldsymbol{s}_{id}^\top (\boldsymbol{f} \cdot t + \begin{bmatrix} y \\ z \end{bmatrix}) \\ &= \mu \cdot \lfloor q/2 \rfloor + \underbrace{x - \boldsymbol{s}_{id}^\top \begin{bmatrix} y \\ z \end{bmatrix}}_{error\ term} \end{aligned} \tag{12}$$

In order to decrypt correctly, the error term $x - \boldsymbol{s}_{id}^\top \begin{bmatrix} y \\ z \end{bmatrix}$ should be bounded by $\lfloor q/4 \rfloor$. Similar to our IBE scheme, the following proof also needs Lemmas 1 and 2 to analyze the error rate of decryption.

Theorem 4. *Let* $q \leq 4[l''(2^\beta - 1)\sqrt{lmn} + 1]\delta c\sigma\sqrt{mn/2\pi}, c \geq 1, t > 15$, *the above HIBE scheme decrypts correctly with overwhelming probability.*

Proof of Theorem 4. Letting $\boldsymbol{s}_{id} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$ with $\boldsymbol{s}_1, \boldsymbol{s}_2 \in R^m$ we have $\boldsymbol{s}_{id}^\top \begin{bmatrix} y \\ z \end{bmatrix} = \boldsymbol{s}_1^\top \cdot \boldsymbol{y} + \boldsymbol{s}_2^\top \cdot \boldsymbol{z}$. Since $\boldsymbol{z} = \boldsymbol{R}_{id}^\top \cdot \boldsymbol{y}$, we have $\|\boldsymbol{z}\| = \|\boldsymbol{R}_{id} \cdot \boldsymbol{y}\| \leq \|\boldsymbol{R}_{id}\| \cdot \|\boldsymbol{y}\| = l''(2^\beta - 1)\sqrt{lmn}\|\boldsymbol{y}\|$.

Then, we compute the decryption error rate with Lemma 2 as

$$\begin{aligned} &Pr\left[(l''(2^\beta - 1)\sqrt{lmn} + 1)\sqrt{m}| < x, y > | \geq q/4\right] \\ &= Pr\left[| < x, y > | \geq q/(4(l''(2^\beta - 1)\sqrt{lmn} + 1)\sqrt{m})\right] \\ &= Pr\left[| < x, y > | \geq T\delta\|x\|\right] \\ &< 2exp(-\pi T^2) \end{aligned} \tag{13}$$

For $c \geq 1$, we have $\|x\| \leq c\sigma\sqrt{n/2\pi}$ with Lemma 1. Then

$$T = \frac{q}{4[l''(2^\beta - 1)\sqrt{lmn} + 1]\sqrt{m}\delta\|x\|} \geq \frac{q}{4[l''(2^\beta - 1)\sqrt{lmn} + 1]\delta c\sigma\sqrt{mn/2\pi}} \tag{14}$$

When T gets large enough, the decryption error rate $2exp(-\pi T^2)$ is negligible, and we can decrypt correctly with overwhelming probability. □

Similar to [18,19,37], we need to set the parameters as follows:

- the error term is less than $q/4$ (*i.e.*, $q \leq 4[l''(2^\beta - 1)\sqrt{lmn} + 1]\delta c\sigma\sqrt{mn/2\pi}$),
- that algorithm TrapGen can operate (*i.e.*, $m = O(n \log q)$),
- that σ is sufficiently large for sampling algorithm (i.e., $\sigma > \|\widetilde{T_B}\| 2^\beta l''\sqrt{lm}\omega\sqrt{\log m} = 2^\beta l''\sqrt{lm}\omega\sqrt{\log m}$),
- that reduction applies (i.e., the number of private key queries $Q \leq q^l/2$).

4.3. Security Proof

In this section, we give the security proof of our HIBE scheme. We describe the definition of abort-resistant hash functions in Definition 2.

Lemma 5. *Let q be a prime and $0 < Q < q$; the hash family $\mathcal{H}_{Wat}$ is $(Q, \frac{1}{q^l}(1-\frac{Q}{q^l}), \frac{1}{q^l})$ abort-resistant.*

Proof of Lemma 5. Let $\bar{id}$ be a set of $(id_0, id_1, \cdots, id_Q)$ where $id_0 \notin \{id_1, \cdots, id_Q\}$. For $i = 0, \cdots, Q+1$, we have $|S_i| = q^{l(l''-1)}$ and $|S_0 \cap S_j| \leq q^{l(l''-2)}$ for $j > 0$. Then,

$$|S| = |S_0 \setminus (S_1 \cup \cdots \cup S_Q)| \geq |S_0| - \sum_{i=1}^{Q} |S_0 \cap S_i| \geq q^{l(l''-1)} - Qq^{l(l''-2)} \tag{15}$$

The non-abort probability of $\bar{id}$ is $|S|/q^{ll''} \geq \frac{1}{q^l}(1-\frac{Q}{q^l})$. Since $|S| \leq |S_0|$, the non-abort probability is $|S|/q^{ll''} \leq |S_0|/q^{ll''} \leq \frac{1}{q^l}$ at most. □

Theorem 5. *The HIBE system with parameters (n, m, q, σ) is IND-ID-CPA secure for depth d in the standard model under the hardness of RLWE.*

Proof of Theorem 5. The proof proceeds in a sequence of games, and the first game is the same as the security game in Definition 1. In game i, we use W_i to denote that adversary guess the challenge message correctly. The advantage of adversary in game i is $|Pr[W_i] - \frac{1}{2}|$.

Game 0. The original IND-ID-CPA game between an adversary $\mathcal{A}$ and a challenger.

Game 1. The challenger builds the public parameters $PP = (\boldsymbol{a}_0, \boldsymbol{a}_{1,1}, \cdots, \boldsymbol{a}_{1,l''}, \cdots, \boldsymbol{a}_{d,l''}, \boldsymbol{b}, u)$ in the original game. These vectors $\boldsymbol{a}_{1,1}, \cdots, \boldsymbol{a}_{1,l''}, \cdots, \boldsymbol{a}_{d,l''}, \boldsymbol{b}$ are chosen uniformly random from R_q^m.

The Game 1 challenger chooses ll'' random matrices $\boldsymbol{R}_{k,i}^* \in R^{m \times m}$ and polynomials $h_{k,i} \in R_q$ for $k \in [1, l], i \in [1, l'']$. Matrix $\boldsymbol{R}_{k,i}^*$ consists of uniformly random polynomials with coefficients $\{-1.1\}$. Then, the challenger generates vectors $\boldsymbol{a}_0$ and $\boldsymbol{b}$ as in original game, and constructs vector $\boldsymbol{a}_{k,i}$ as

$$\boldsymbol{a}_{k,i} \leftarrow (\boldsymbol{R}_{k,i})^\top \cdot \boldsymbol{a}_0 - h_{k,i} \cdot \boldsymbol{b} \in R_q^m, k \in [1, l], i \in [1, l''] \tag{16}$$

In the adversary's view, the distribution $\boldsymbol{a}_0^\top \cdot \boldsymbol{R}_{\boldsymbol{k,i}}^*$ is statistically close to uniform $(\boldsymbol{a}'_{k,i})^\top$ and independent of vector $\boldsymbol{z}$. Therefore, in adversary's view, vecors $\boldsymbol{a}_{k,i}$ are uniformly random elements as in Game 0. This shows that

$$Pr[W_0] = Pr[W_1] \tag{17}$$

Game 2. In Game 2, we add an abort event which is similar to the abort event in Section 3.3. The rest is the same as Game 1. We use the abort-resistant $\mathcal{H}_{Wat}$ introduced in Lemma 5.

According to [18], they show that $\epsilon_{max} - \epsilon_{min}$ is less than $\epsilon_{\min} \left|\Pr[W_1] - \frac{1}{2}\right|$. Since $q^l \geq 2Q$, we have $\epsilon_{min} = \frac{1}{q^l}(1-\frac{Q}{q^l}) \geq \frac{1}{2q^l}$. By Lemma 4, we have

$$\left|\Pr[W_2] - \frac{1}{2}\right| \geq \frac{1}{2}\epsilon_{\min} \left|\Pr[W_1] - \frac{1}{2}\right| \geq \frac{1}{4q^l} \left|\Pr[W_1] - \frac{1}{2}\right| \tag{18}$$

Game 3. In Game 3, we change the method of generating $\boldsymbol{a}_0$ and $\boldsymbol{b}$ in PP. Vector $\boldsymbol{a}_0$ is generated as a random vector in R_q^m and vector $\boldsymbol{b}$ is generated by algorithm TrapGen. The challenger also gets a trapdoor $T_{\boldsymbol{b}}$ of $\Lambda_q^\perp(\boldsymbol{b})$. The construction $\boldsymbol{a}_{k,i} \leftarrow (\boldsymbol{R}_{k,i}^*)^\top \cdot \boldsymbol{a}_0 - h_{k,i} \cdot \boldsymbol{b} \in R_q^m$ is the same as in Game 2. To answer the private key query of $id = (id_1, id_2, \cdots, id_l)$, the challenger generates the corresponding private key $SK_{id} = \boldsymbol{s}$ from $\Lambda_q^u(\boldsymbol{f}_{id})$. Let

$$\boldsymbol{f}_{id|id_l} := \begin{pmatrix} \boldsymbol{a}_0 \\ \sum_{i=1}^{l''} \boldsymbol{a}_{1,i} b_{1,i} + \boldsymbol{b} \\ \vdots \\ \sum_{i=1}^{l''} \boldsymbol{a}_{l,i} b_{l,i} + \boldsymbol{b} \end{pmatrix} \text{ or } \boldsymbol{f}_{id} = \begin{pmatrix} \boldsymbol{a}_0 \\ (\boldsymbol{R}_{id})^\top \cdot \boldsymbol{a}_0 - h_{id} \cdot \boldsymbol{b} \end{pmatrix} \tag{19}$$

where

$$\boldsymbol{R}_{id} := \sum_{i=1}^{l''} b_{1,i} \boldsymbol{R}^*_{1,i} || \cdots || \sum_{i=1}^{l''} b_{l,i} \boldsymbol{R}^*_{l,i} \in R^{m \times lm} \tag{20}$$

and

$$h_{id} = (1 + \sum_{i=2}^{l''} b_{1,i} \cdot h_{1,i}) || \cdots || (1 + \sum_{i=1}^{l''} b_{l,i} \cdot h_{l,i}) \tag{21}$$

If $h_{id} = 0$, the challenger aborts the game as in Game 2. Otherwise, the challenger gets private key $\boldsymbol{s} \leftarrow SampleRight(\boldsymbol{a}_0, h_{id} \cdot \boldsymbol{b}, \boldsymbol{R}_{id}, T_{\boldsymbol{b}}, u, \sigma) \in R_q^{2m}$. Then, it sends $SK_{id} = \boldsymbol{s}$ to the adversary $\mathcal{A}$. In the adversary's view, Game 2 and Game 3 are indistinguishable. Therefore,

$$Pr[W_2] = Pr[W_3] \tag{22}$$

Game 4. The challenge ciphertext $(c_0^*, \boldsymbol{c}_1^*)$ is randomly selected in $R_q \times R_q^{2m}$ and the rest is the same as in Game 3, so the advantage of $\mathcal{A}$ is 0 in Game 4. Similar to Section 3.3, we need to prove that Game 3 and Game 4 are computationally indistinguishable.

Instance. For $i = 0, \cdots, m$, $\mathcal{B}$ receives RLWE samples $(u_i, v_i) \in R_q \times R_q$.

Setup. $\mathcal{B}$ generates the public parameters:

1. Construct random vector $\boldsymbol{a}_0 \in R_q^m$ with RLWE samples. For $i = 1, \cdots, m$, the i-th column of $\boldsymbol{a}_0$ is u_i.
2. Let a random polynomial $u_0 \in R_q$ be the 0-th RLWE sample.
3. Construct $\boldsymbol{a}_{k,i}$ and $\boldsymbol{b}$ as in Game 3.
4. Send public parameters $PP = (\boldsymbol{a}_0, \boldsymbol{a}_{1,1}, \cdots, \boldsymbol{a}_{1,l''}, \cdots, \boldsymbol{a}_{d,l''}, \boldsymbol{b}, u)$ to adversary $\mathcal{A}$.

Phase 1 and Phase 2. $\mathcal{B}$ answers private key queries as in Game 3.

Challenge. $\mathcal{A}$ submits a target identity $id^* = (id_1^*, \cdots, id_l^*)$ and a message $\mu^* \in \{0,1\}^n$. $\mathcal{B}$ returns a challenge ciphertext as follows:

1. Set $\boldsymbol{v}^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in R_q^m$ with the RLWE instance.
2. Set $c_0^* = v_0 + \mu^* \cdot \lfloor q/2 \rceil \in R_q$ to blind the message bit.
3. Set $\boldsymbol{R}_{id^*} := \sum_{i=1}^{l''} b_{1,i} \boldsymbol{R}^*_{1,i} || \cdots || \sum_{i=1}^{l''} b_{l,i} \boldsymbol{R}^*_{l,i}$ and $\boldsymbol{c}_1^* = \begin{bmatrix} \boldsymbol{v}^* \\ (\boldsymbol{R}^*_{id^*})^\top \cdot \boldsymbol{v}^* \end{bmatrix}$.
4. Choose a random bit $r \in \{0,1\}$. If $r = 0$ set $CT^* = (c_0^*, \boldsymbol{c}_1^*)$, otherwise, select a random $CT^* = (c_0, \boldsymbol{c}_1)$ in $R_q \times R_q^{2m}$. Then, send the challenge ciphertext CT^* to adversary.

Guess. Finally, the adversary $\mathcal{A}$ returns a guess r'. The simulator $\mathcal{B}$ outputs 1 if $r' = r$ otherwise 0.

Analysis. According to [18], the challenge ciphertext is the same as valid ciphertext in game 3 if sampling oracle $\mathcal{O}$ is pseudo-random $\mathcal{O}_s$, and the challenge ciphertext is the same as random ciphertext

in game 4 if oracle $\mathcal{O}$ is truly random $\mathcal{O}_\$$. The simulator's advantage in solving RLWE problem is equal to $\mathcal{A}$'s advantage in distinguishing valid ciphertext and random ciphertext. For $Pr[W_4] = \frac{1}{2}$, we get

$$|Pr[W_3] - \frac{1}{2}| = |Pr[W_3] - Pr[W_4]| \leq Adv_{\mathcal{B}}^{RLWE} \tag{23}$$

Then

$$|Pr[W_0] - \frac{1}{2}| \leq 4q^l \cdot Adv_{\mathcal{B}}^{RLWE} \tag{24}$$

□

5. Efficiency

Trade-off. We make a trade-off between the decrease in the size of public parameters and the increase in the computation cost. Using the blocking technique, we divide an identity into l' segments, and the number of elements in public parameters is reduced from $l+2$ to $l/\beta+2$ where β is a flexible constant. Therefore, the percentage of decrease in public parameter space is $\frac{l-l'}{l+2}$ and it is shown as the thin blue line in Figure 1 with $l = 160$. According to the analysis of Singh [19], there is no effect of l' on cost of key generation, encryption and decription. However, we need to increase the value of lattice modulo q for maintaining the same security level, and it will increase the computation cost. According to Chatterjee 's work [20], the number of bits in q is increased by $\Delta = \beta - \log_2 \beta$. We use $|q|$ to denote the bit length of q and then $|q'| = |q| + 2\Delta = |q| + 2(\beta - \log_2 \beta)$. The percentage of increase in computation cost is $\frac{|q'|-|q|}{|q|} = \frac{2(\beta-\log_2 \beta)}{|q|}$ and it is shown as the thick red line in Figure 1 with $|q| = 256$. In Figure 1, the x-axis represents the value of β, and the y-axis represents the percentage of increase or decrease. For $l = 160$ and $|q| = 256$, the size of public parameters is reduced by 89.7% while the cost of computation is merely increased by 5.2% when $l' = 16$ or $\beta = 10$. If we set $l' = 8$ or $\beta = 20$, the size of public parameters is reduced by 93.8% while the computational cost is merely increased by 12.25%.

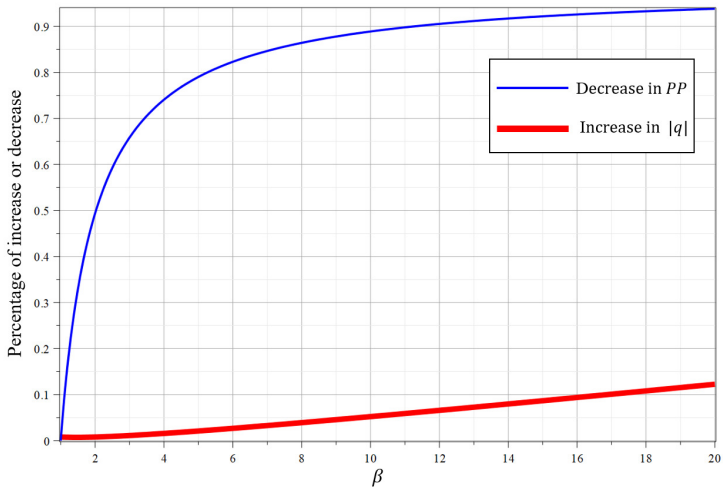

Figure 1. Relative decrease in PP and relative increase in $|q|$

Comparisons. We propose an adaptively secure IBE scheme in Section 3. Table 1 shows the comparison of storage space between different IBE schemes in the standard model. In this table, PP, SK, l denote the public parameters, private keys and length of user's identity.

Table 1. Comparison of storage space.

Schemes	*PP* Size	*SK* Size	Ciphertext Size	Security	Assumption
[18]	$(l+2)mn\log q$	$2m\log q$	$(2m+1)\log q$	Adaptive-CPA	LWE
[23]	$(\log l+2)mn\log q$	$mn\log q$	$(m+n)\log q$	Adaptive-CPA	LWE
[22] *	$(d\lceil l^{1/d}\rceil+2)mn\log q$	$2m\log q$	$(2m+1)\log q$	Adaptive-CPA	LWE
[36] *	$(d\lceil l^{1/d}\rceil+2)mn\log q$	$2mn\log q$	$(2m+1)n\log q$	Adaptive-CPA	RLWE †
[24]	$(\log^2 l+2)mn\log q$	$2m\log q$	$(2m+1)\log q$	Adaptive-CPA	LWE
Ours **	$(l/\beta+2)mn\log q$	$2mn\log q$	$(2m+1)n\log q$	Adaptive-CPA	RLWE †

* In [22] and [36], they use an injective map which maps an identity $id\in\{0,1\}^l$ to a subset of $[1,\lceil l^{1/d}\rceil]^d$, where the element d is a flexible constant. The choice of d will affect the reduction cost; ** In our construction, the element β is a flexible constant. The choice of β will affect the size of modulus q and we make a trade-off in the previous part; † Our scheme and [36] only work over the rings R_q; thus, the basic elements in the public parameters are polynomial vectors rather than matrices.

Since the public parameters are composed of multiple matrices, its size will directly affect the communication overhead in actual applications. As shown in this table, the public parameter in Agrawal's construction [18] contains $l+2$ matrices. Zhang's construction [23] achieves shorter public parameter at the cost of weaker security guarantees. In Yamada's construction [22], the public parameter consists of $d\lceil l^{1/d}\rceil+2$ matrices, where d is a constant. In Katsumata's scheme [36], the public parameter consists of $d\lceil l^{1/d}\rceil+2$ vectors because of ring setting. The relationship between the size of public parameters and constant d is shown in Figure 2. For $l=160$, the minimum size of public parameters is 17 vectors when we set $d=5$. Moreover, we need to set d very small (e.g., $d=2$ or 3) because of the reduction cost. If we set $d=2$ (resp. 3), the public parameters have 28 (resp. 20) vectors. In [24], the public parameter consists of $\log^2 l+2$ matrices via new partitioning functions. In our construction, the public parameters only contain $l'+2$ vectors, where $l'=l/\beta$. We have analyzed the choice of β or l' in the previous part. For $l=160$, the public parameter only contains 10 (resp. 18) vectors if we choose $\beta=20$ (resp. 10).

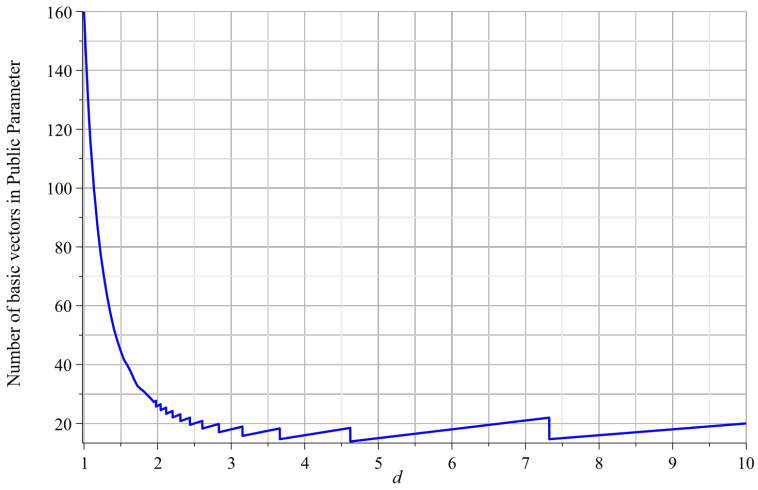

Figure 2. The relationship between the size of public parameters and constant d.

The comparison of public parameter size is shown in the Figure 3. It involves four IBE schemes with short public parameters, including Yam17 [24] , KY16 [36] ($d=3$), ZCZ16 [23] and ours ($\beta=20$). The x-axis represents the length of user's identity, and the y-axis represents the number of basic matrices (or vectors) in the public parameters of each scheme. Obviously, the public parameters in our

scheme are shorter than [24] and [36]. Moreover, it can be shorter than [23] if the identity length l is small (e.g., less than 140).

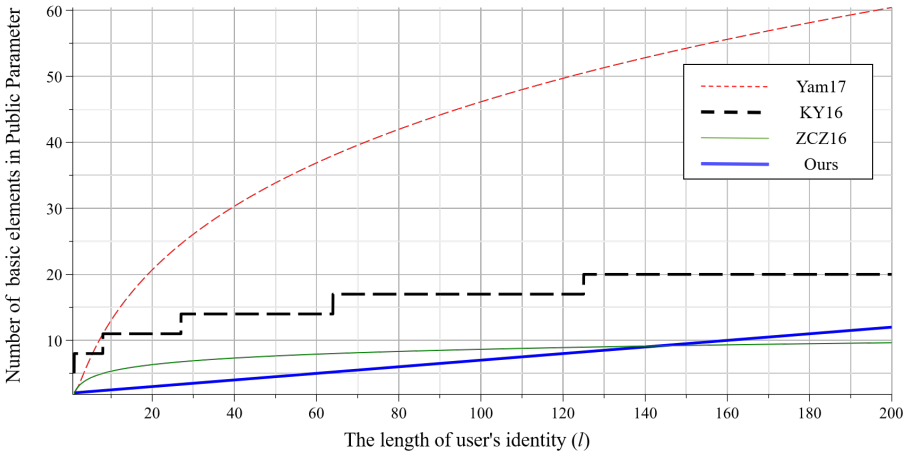

Figure 3. Comparison of public parameter size in different schemes.

Compared with the LWE-based scheme, the RLWE-based scheme contains a lot of polynomial operations instead of matrix operations. To compare more fair, we only compare the computational efficiency between the schemes under RLWE assumption. Since the scheme by [36] also has short public parameters and ring setting, we only compare the calculation efficiency between [36] and our scheme. Table 2 shows the comparison of computational efficiency. In this table, *KeyGen*, *Enc*, *Dec* denote the key generation, encryption and decryption.

Table 2. Comparison of computational efficiency.

Schemes	*KeyGen*	*Enc*	*Dec*
[36]	dm^2n^2	$dm^2n^2 + n^2 + 2mn$	$2mn^2$
Ours	$l'mn$	$l'mn + n^2 + 2mn$	$2mn^2$

The difference between these two schemes is the calculation of $H(id)$ and $\boldsymbol{a}_{id}$. In Katsumata's construction [36], $H(id) = \boldsymbol{b} + \Sigma_{j_1,\cdots,j_d} PubEval_d(\boldsymbol{b}_{1,j_1}, \boldsymbol{b}_{2,j_2}, \cdots, \boldsymbol{b}_{d,j_d})$ and it is used to generate private keys. They use the homomorphic function $PubEval_d : (R_q^m)^d \rightarrow R_q^m$ as in [22], which maps vectors $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_d$ to a vector in R_q^m. The function $PubEval$ needs dm^2n^2 multiplications and $d-1$ inversions. In our construction, $\boldsymbol{a}_{id} = \boldsymbol{b} + \Sigma_{i=1}^{l'} b_i \cdot \boldsymbol{a}_i$ and it is also used as the input of the sampling algorithm to generate private keys. However, it only needs $l'mn$ multiplication operations which is obviously less than [36].

In Section 4, we also extend our IBE scheme to an adaptively secure HIBE scheme. Using Waters' technology, we can convert the selectively secure HIBE scheme to adaptive security. Howerve, the size of the public parameter increases from $d+2$ matrices to $dl'+2$ matrices. In our HIBE construction, the public parameter is reduced from $dl'+2$ matrices to $dl''+2$ vectors where $l'' = l'/\beta$. In particular, it can be further reduced to $l''+2$ thanks to the method of Chatterjee [11,43]. Finally, both of our constructions support multi-bit encryption because of ring setting.

6. Conclusions

In this paper, we propose an identity-based encryption scheme and a hierarchical identity-based encryption scheme over ideal lattice. The new schemes have short public parameters, and achieve IND-ID-CPA security in the standard model. In addition, we use the trapdoor of Micciancio to further

improve the efficiency of our scheme. However, there are still many problems to be solved, such as how to reduce the size of ciphertext and how to implement these schemes.

Author Contributions: Conceptualization, S.Z.; Investigation, Y.G. and S.Z.; Methodology, Y.Z., Y.L., Y.G. and L.W.; Validation, Y.Z. and Y.L.; Writing—original draft, Y.Z.; Writing—review & editing, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shandong Provincial Key Research and Development 512 Program of China: 2018CXGC0701; National Natural Science Foundation of China (NSFC): No. 61972050; BUPT Excellent Ph.D. Students Foundation: No. CX2019119 and in part by the 111 Project: No. B08004.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 1984; pp. 47–53._5. [CrossRef]
2. Boneh, D.; Franklin, M.K. Identity-Based Encryption from the Weil Pairing. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberger, Germany, 2001; pp. 213–229._13. [CrossRef]
3. Canetti, R.; Halevi, S.; Katz, J. A Forward-Secure Public-Key Encryption Scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2003; pp. 255–271. [CrossRef]
4. Boneh, D.; Boyen, X. Secure Identity Based Encryption Without Random Oracles. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberger, Germany, 2004; pp. 443–459. [CrossRef]
5. Waters, B. Efficient Identity-Based Encryption Without Random Oracles. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2005; pp. 114–127. [CrossRef]
6. Cocks, C.C. An Identity Based Encryption Scheme Based on Quadratic Residues. In *IMA International Conference on Cryptography and Coding*; Springer: Berlin/Heidelberger, Germany, 2001; pp. 360–363. [CrossRef]
7. Gentry, C.; Silverberg, A. Hierarchical ID-Based Cryptography. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberger, Germany, 2002; pp. 548–566. [CrossRef]
8. Horwitz, J.; Lynn, B. Toward Hierarchical Identity-Based Encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2002; pp. 466–481. [CrossRef]
9. Boneh, D.; Boyen, X. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2004; pp. 223–238. [CrossRef]
10. Gentry, C. Practical Identity-Based Encryption Without Random Oracles. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2006; pp. 445–464. [CrossRef]
11. Chatterjee, S.; Sarkar, P. HIBE With Short Public Parameters without Random Oracle. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberger, Germany, 2006; pp. 145–160. [CrossRef]
12. Canetti, R.; Halevi, S.; Katz, J. A Forward-Secure Public-Key Encryption Scheme. *J. Cryptol.* **2007**, *20*, 265–294. [CrossRef]
13. Waters, B. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberger, Germany, 2009; pp. 619–636. [CrossRef]
14. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* **2005**, *56*, 1–40. [CrossRef]
15. Stehlé, D.; Steinfeld, R.; Tanaka, K.; Xagawa, K. Efficient Public Key Encryption Based on Ideal Lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberger, Germany, 2009; pp. 617–635. [CrossRef]

16. Lyubashevsky, V.; Peikert, C.; Regev, O. On Ideal Lattices and Learning with Errors over Rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2010; pp. 1–23. [CrossRef]
17. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, BC, Canada, 17–20 May, 2008; pp. 197–206. [CrossRef]
18. Agrawal, S.; Boneh, D.; Boyen, X. Efficient Lattice (H)IBE in the Standard Model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2010; pp. 553–572. [CrossRef]
19. Singh, K.; Rangan, C.P.; Banerjee, A.K. Adaptively Secure Efficient Lattice (H)IBE in Standard Model with Short Public Parameters. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*; Springer: Berlin/Heidelberger, Germany, 2012; pp. 153–172. [CrossRef]
20. Chatterjee, S.; Sarkar, P. Trading Time for Space: Towards an Efficient IBE Scheme with Short(er) Public Parameters in the Standard Model. In *International Conference on Information Security and Cryptology*; Springer: Berlin/Heidelberger, Germany, 2005; pp. 424–440. [CrossRef]
21. Naccache, D. Secure and practical identity-based encryption. *IET Inf. Secur.* **2005**, *1*, 59–64. [CrossRef]
22. Yamada, S. Adaptively Secure Identity-Based Encryption from Lattices with Asymptotically Shorter Public Parameters. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2016; pp. 32–62. [CrossRef]
23. Zhang, J.; Chen, Y.; Zhang, Z. Programmable Hash Functions from Lattices: Short Signatures and IBEs with Small Key Sizes. In *Annual international cryptology conference*; Springer: Berlin/Heidelberger, Germany, 2016; pp. 303–332. [CrossRef]
24. Yamada, S. Asymptotically Compact Adaptively Secure Lattice IBEs and Verifiable Random Functions via Generalized Partitioning Techniques. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberger, Germany, 2017; pp. 161–193. [CrossRef]
25. Agrawal, S.; Boyen, X. Identity-Based Encryption from Lattices in the Standard Model. 2009. Available online: http://www.cs.stanford.edu/~xb/ab09/ (accessed on 20 October 2020).
26. Cash, D.; Hofheinz, D.; Kiltz, E. How to Delegate a Lattice Basis. *IACR Cryptol. ePrint Arch.* **2009**, *2009*, 351.
27. Cash, D.; Hofheinz, D.; Kiltz, E.; Peikert, C. Bonsai Trees, or How to Delegate a Lattice Basis. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2010; pp. 523–552. [CrossRef]
28. Agrawal, S.; Boneh, D.; Boyen, X. Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In *Annual Cryptology Conference*; Springer: Berlin/Heidelberger, Germany, 2010; pp. 98–115. [CrossRef]
29. Wang, F.; Wang, C.; Liu, Z.H. Efficient hierarchical identity based encryption scheme in the standard model over lattices. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 781–791. [CrossRef]
30. Apon, D.; Fan, X.; Liu, F. Compact identity based encryption from LWE. *Cryptol. ePrint Arch.* **2016**, *2016*.
31. Boyen, X.; Li, Q. Towards tightly secure lattice short signature and id-based encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberger, Germany, 2016; pp. 404–434.
32. Zhang, L.; Wu, Q. Adaptively Secure Hierarchical Identity-Based Encryption over Lattice. In *International Conference on Network and System Security*; Springer: Berlin/Heidelberger, Germany, 2017; pp. 46–58. [CrossRef]
33. Yang, X.; Wu, L.; Zhang, M.; Chen, X. An efficient CCA-secure cryptosystem over ideal lattices from identity-based encryption. *Comput. Math. Appl.* **2013**, *65*, 1254–1263. [CrossRef]
34. Ducas, L.; Lyubashevsky, V.; Prest, T. Efficient Identity-Based Encryption over NTRU Lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberger, Germany, 2014; pp. 22–41. [CrossRef]
35. Hoffstein, J., Pipher, J., Silverman, J.H. NTRU: A Ring-Based Public Key Cryptosystem. In *ANTS-III*; Springer: Berlin/Heidelberger, Germany, 1998; pp. 267–288. [CrossRef]
36. Katsumata, S.; Yamada, S. Partitioning via Non-linear Polynomial Functions: More Compact IBEs from Ideal Lattices and Bilinear Maps. In *International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberger, Germany, 2016; pp. 682–712. [CrossRef]

37. Bert, P.; Fouque, P.; Roux-Langlois, A.; Sabt, M. Practical Implementation of Ring-SIS/LWE Based Signature and IBE. In *International Conference on Post-Quantum Cryptography*; Springer: Berlin/Heidelberger, Germany, 2018; pp. 271–291. [CrossRef]
38. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberger, Germany, 2012; pp. 700–718. [CrossRef]
39. Peikert, C. Bonsai Trees (or, Arboriculture in Lattice-Based Cryptography). *IACR Cryptol. ePrint Arch.* **2009**, *2009*, 359.
40. Ajtai, M. Generating Hard Instances of Lattice Problems (Extended Abstract). In the Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 99–108. [CrossRef]
41. Banaszczyk, W. New bounds in some transference theorems in the geometry of numbers. *Math. Ann.* **1993**, *296*, 625–635. [CrossRef]
42. Banaszczyk, W. Inequalites for Convex Bodies and Polar Reciprocal Lattices in R^n. *Discret. Comput. Geom.* **1995**, *13*, 217–231. [CrossRef]
43. Singh, K.; Rangan, C.P.; Banerjee, A.K. Efficient Lattice HIBE in the Standard Model with Shorter Public Parameters. In *Information and Communication Technology-EurAsia Conference*; Springer: Berlin/Heidelberger, Germany, 2014; pp. 542–553. [CrossRef]