Adaptively Secure Efficient Lattice (H)IBE in Standard Model with Short Public Parameters

Kunwar Singh[1], C. Pandurangan[2], and A.K. Banerjee[1]

[1] Computer Science and Engineering Department
National Institute of Technology, Tiruchirappalli, India
{kunwar,banerjee}@nitt.edu
[2] Computer Science and Engineering Department
IIT Madras, Chennai, India
rangan@cse.iitm.ac.in

Abstract. Independent work by Chatterjee and Sarkar [9] and Naccache [16] provided a variant of Waters' IBE to reduce public parameters. The idea is to divide an l-bit identity into l' blocks of l/l' so that size of the vector $\overrightarrow{V}$ can be reduced from l elements of G to l' elements of G. We name this technique as blocking technique. This leads to some associated degradation in security reduction. In this paper our contribution is two fold: First we apply Waters' [21] idea to convert Agrawal et al. [1] selective-ID secure lattice HIBE to adaptive-ID secure HIBE then using blocking technique we reduce the public parameters. Second we present efficient lattice identity based encryption in standard model with smaller public key size which is variant of [1]. Using blocking technique our scheme reduces public key size by a factor of β at the cost of increasing $(\beta - lg(\beta))^2$ number of bits in q where q is size of field Z_q. There is an interesting trade-off between reducing the public parameter size and increase in the computational cost. For 160-bit identities we show that compared to scheme [1] the public parameter size can be reduced by almost 90% while increasing the computation cost by only 8.71% for appropriate choice of β.

Keywords: Lattice, Hierarchical Identity Base Encryption, Standard model, Learning with error(LWE).

1 Introduction

The concept of identity-based cryptosystem was introduced by Adi Shamir in 1984 [20]. In this new paradigm users' public key can be any string which uniquely identifies the user. For example email or phone number can be public key. As a result, it significantly reduces system complexity and cost of establishing public key infrastructure. Although Shamir constructed an identity-based signature scheme using RSA function but he could not construct an identity-based encryption and this became a long-lasting open problem. Only in 2001, Shamir's open problem was independently solved by Boneh and Franklin [6] and Cocks [11].

A. Bogdanov and S. Sanadhya (Eds.): SPACE 2012, LNCS 7644, pp. 153–172, 2012.

First Canetti et al. [7] presented identity-based encryption in standard model. They proved the security of scheme in selective-ID model. In the selective-ID model the adversary must first declare which identity it wishes to be challenged before the global parameters are generated. Boneh and Boyen [4] then provided an efficient secure scheme in selective-ID model. Boneh and Boyen [5] describe a scheme that is fully secure in standard model, but their scheme is too inefficient to practical use. Finally, the first practical and fully secure IBE scheme was proposed by Waters [21] in the standard model under the Decisional Bilinear Diffie-Hellman assumption. However, one drawback was that the public parameters is very large: namely, the public parameters contain $l + 4$ group elements, where l is the size of the bit-string representing identities. In that scheme, if the identities are n-bit string then one needs $\overrightarrow{V}$ consists of n group elements. Independent work by Chatterjee and Sarkar [9] and Naccache [16] provided a variant of Waters' IBE. The idea is to divide an l-bit identity into l' blocks of l/l' so that size of the vector $\overrightarrow{V}$ can be reduced from l elements of G to l' elements of G. We name this technique as blocking technique. This leads to some associated degradation in security reduction.

The task of Public Key Generator (PKG) in IBE is to authenticate identity of the entity, generate the private key corresponding to identity of the entity and finally transmit the private key securely to the entity. In large network PKG has a burdensome job. So the notion of Hierarchical IBE (HIBE) was introduced in [13,14,5] to distribute the workload by delegating private key generation and identity authentication to lower-level PKGs. However, lower level PKGs do not have their own public parameters. Only root PKG has some set of public parameters.

Lattice based cryptogrphy have arisen in recent years. Lattice based cryptography are attractive due to their worst case hardness assumption and their potential resistance to quantum computers. Recently Regev [19] defined the learning with errors (LWE) problem and proved that it enjoys similar worst-case hardness properties, under a quantum reduction.

Based on LWE problem, Gentry et al. [18] constructed lattice based IBE scheme in random oracle model. Recently Cash et al. [8], Peikert [17], and Agarwal et al. [2] have constructed secure IBE in the standard model from LWE problem. Their construction view an identity as a sequence of bits and then assign a matrix to each bit, which resulted into less efficient scheme compared to Gentry et al.[18]. Recently Agarwal et al.[1] constructed a efficient lattice based selective-ID secure IBE scheme in standard model. They have considered identities as one chunk rather than bit-by-bit. As Water modified Boneh Boyen selective-ID secure IBE scheme (BB-IBE1)[5] to obtain an adaptive-ID (full model) secure IBE scheme [21], similarly Agarwal et al.[1] in their full version paper constructed an adaptive secure IBE using LWE problem. Similar to Waters [21], it has large public parameters of size l $n \times m$ matrices, where l is the size of the bit-string representing identities.

Recently Cash et al. [8] and Peikert [17] have constructed secure HIBE in the standard model using basis delegation technique. Their construction view an identity as a sequence of bits and then assign a matrix to each bit, which resulted into less efficient scheme. Recently Agarwal et al. [1] constructed a efficient lattice based secure HIBE scheme in standard model in weaker security notion i.e. selective-ID. They have considered identities as one chunk rather than bit-by-bit.

Our Contributions. Our contribution is two fold: First we apply Waters' [21] idea to convert Agrawal et al. [1] selective-ID secure lattice HIBE to adaptive-ID secure HIBE. Then using blocking technique we reduce the public parameters. Second one drawback of Agarwal et al. adaptive secure IBE scheme[1] was that the public parameters is very large: namely, the public parameters contain $l + 1$ $n \times m$ matrices, where l is the size of the bit-string representing identities. Using blocking technique we can reduce $n \times m$ matrices by factor β or public parameters is reduced by around factor β; encryption and decryption are almost as efficient as in [1]. This is associated with increase in size of q by 2^{β} where q is a prime number and size of field Z_q. We show that compared to scheme [1] the public parameter size can be reduced by almost 90% while increasing the computation cost by only 8.71% for appropriate choice of β.

2 Preliminaries

2.1 Hierarchical IBE and IBE

Here definitions and security model of HIBE and IBE are similar to [13,14,5,1]. User at depth l is defined by its tuple of ids : $(id/id_l) = (id_1, ..., id_l)$. The user's ancestors are the root PKG and the prefix of id tuples (users/lower level PKGs).

HIBE consists of four algorithms.

Setup$(d, \lambda$:) On input a security parameter d(maximum depth of hierarchy tree) and λ, it outputs the public parameters and master key of root PKG.

Derive(PP,$(id/id_l), SK_{(id/id_l)}$): On input public parameters PP, an identity $(id/id_l) = (id_1, ..., id_l)$ at depth l and the private key $SK_{(id/id_{l-1})}$ corresponding to parent identity $(id/id_{l-1}) = (id_1, ..., id_{l-1})$ at depth $l - 1 \geq 0$ the algorithm outputs private key for the identity (id/id_l) at depth l.

If $l = 1$ then $SK_{(id/id_0)}$ is defined to be master key of root PKG.

The private key corresponding to an identity $(id/id_l) = (id_1, ..., id_l)$ at depth l can be generated by PKG or any ancestor (prefix) of an identity (id/id_l).

Encrypt(PP,(id/id_l),M): On input public parameters PP, an identity (id/id_l), and a message M outputs ciphertext C.

Decrypt(PP,$SK_{(id/id_l)}$,C): On input public parameters PP, a private key $SK_{(id/id_l)}$, and a ciphertext C outputs message M.

Identity Based Encryption. IBE is special case of HIBE when depth of hierarchy tree is one.

2.2 Adaptive-ID (Full) Security Model of HIBE and IBE

We define adaptive-ID security model using a game that the challenge ciphertext is indistinguisable from a random element in the ciphertext space. This property implies both semantic security and recipient anonymity. The game proceeds as follows.

Setup: The challenger runs Setup($1^\lambda, 1^d$) and gives the public parameters PP to adversary and keeps master key MK to itself.

Phase 1: The adversary issues a query for a private key for identity $(id/id_k) = (id_1, ..., id_k)$, $k \leq d$. Adversary can repeat this multiple times for different identities adaptivly.

Challenge: The adversary submits identity id* and message M. Identity id* and prefix of id* should not be one of the identity query in phase 1. The challenger picks a random bit $r \in \{0, 1\}$ and a random ciphertext C. If $r = 0$ it sets the challenge ciphertext to $C^* := Encrypt(PP, id^*, M)$. If $r = 1$ it sets the challenge ciphertext to $C^* := C$. It sends C^* as challenge to the adversary.

Phase 2: Phase 1 is repeated with the restriction that the adversary can not query for id^* and prefix of id^*.

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

We refer an adversary A as an IND-ID-CPA adversary. We define the advantage of the adversary A in attacking an IBE scheme ξ as

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

Definition 1. We say that depth d HIBE scheme ξ is adaptive-ID, indistinguishable from random if for all IND-ID-CPA PPT adversaries A we have $Adv_{d,\xi,A}(\lambda)$ is a negligible function.

Full Security Model of IBE. Security model of IBE is same as security model of HIBE with depth of hierarchy tree is one.

2.3 Integer Lattices

A lattice is defined as the set of all integer combinations

$$L(b_1, ..., b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in Z \text{ for } 1 \leq i \leq n \right\}$$

of n linearly independent vectors $b_1, ..., b_n \in R^n$. The set of vectors $\{b_1, ..., b_n\}$ is called a basis for the lattice. A basis can be represented by the matrix $B = [b_1, ..., b_n] \in R^{n \times n}$ having the basis vectors as columns. Using matrix notation, the lattice generated by a matrix $B \in R^{n \times n}$ can be defined as $L(B) = \{Bx : x \in Z^n\}$, where Bx is the usual matrix-vector multiplication. The determinant of a lattice is the absolute value of the determinant of the basis matrix $det(L(B)) = |det(B)|$.

Definition 2. For q prime, $A \in Z_q^{n \times m}$ and $u \in Z_q^n$, define:

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

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

2.4 The Gram-Schmidt Norm of a Basis

Let S be a set of vectors S $= \{s_1, ..., s_k\}$ in R^m. We use the following notation:

- $|S|$ denotes the L_2 length of the longest vector in S, i.e. $\|S\| := max_i |s_i|$ for $1 \leq i \leq k$.
- $\widetilde{S} := \{\widetilde{s_1}, ..., \widetilde{s_k}\} \subset R^m$ denotes the Gram-Schmidt orthogonalization of the vector $s_1, ..., s_k$ taken in that order.

We refer to $\widetilde{\|S\|}$ as the Gram-Schmidt norm of S.

Lemma 1([15, Lemma 7.1]). Let Λ be an m-dimensional lattice. There is a deterministic polynomial-time algorithm that, given an arbitrary basis of Λ and a full-rank set $S = \{s_1, ..., s_m\}$ in Λ, returns a basis T of Λ satisfying

$$\|\widetilde{T}\| \leq \|\widetilde{S}\| \ \ and \ \ \|T\| \leq \|S\|\sqrt{m}/2$$

Theorem 1([3, Theorem 3.2]). Let $q \geq 3$ be odd and $m := \lceil 6n\log q \rceil$.

There is probabilistic polynomial-time algorithm TrapGen(q, n) that outputs a pair $(A \in Z_q^{n \times m}, S \in Z^{n \times m})$ such that A is statistically close to a uniform matrix in $Z_q^{n \times m}$ and S is a basis for $\Lambda_q^{\perp}(A)$ satisfying

$$\|\widetilde{S}\| \leq O(\sqrt{n \log q}) \ \ and \ \ \|S\| \leq O(n \log q)$$

with all but negligible probability in n.

Theorem 2([17]). For $i = 1, 2, 3$ let A_i be a matrix in $Z_q^{n \times m_i}$ and $A = (A_1|A_2|A_3)$. Let T_2 be a basis of $\Lambda_q^{\perp}(A_2)$. There is deterministic polynomial time algorithm ExtendBasis(A_1, A_2, A_3, T_2) that outputs a basis T for $\Lambda_q^{\perp}(A)$ such that $\|\widetilde{T}\| = \|\widetilde{T_2}\|$.

2.5 Discrete Gaussians

Let L be a subset of Z^m. For any vector $c \in R^m$ and any positive parameter $\sigma \in R > 0$, define:

$\rho_{\sigma,c}(x) = exp(-\pi \frac{\|x-c\|}{\sigma^2})$: a Gaussian-shaped function on R^m with center c and parameter σ,
$\rho_{\sigma,c}(L) = \sum_{x \in L} \rho_{\sigma,c}(x)$: the (always converging) $\rho_{\sigma,c}$ over L,
$D_{L,\sigma,c}$: the discrete Gaussian distribution over L with parameters σ and c,

$$\forall y \in L \ , \ D_{L,\sigma,c} = \frac{\rho_{\sigma,c}(y)}{\rho_{\sigma,c}(L)}$$

we abbreviate $\rho_{\sigma,0}$ and $D_{L,\sigma,c}$ will most often be defined over the Lattice $L = \Lambda_q^{\perp}$ for a matrix $A \in Z_q^{n\times m}$ or over a coset $L = t + \Lambda_q^{\perp}(A)$ where $t \in Z^m$.

Lemma 2 ([17, Lemma 2.4]). Let $q \geq 2$ and let A be a matrix in $Z_q^{n\times m}$ with $m > n$. Let T_A be a basis for $\Lambda_q^{\perp}(A)$ and $\sigma \geq \|\widetilde{T_A}\|\omega(\sqrt{(\log m)})$. Then for $c \in R^m$ and $u \in Z_q^n$:

1. There is a PPT algorithm SampleGaussian (A, T_A, σ, c) that returns $x \in \Lambda_q^{\perp}(A)$ drawn from a distribution statistically close to $D_{\Lambda,\sigma,c}$.
2. There is a PPT algorithm SamplePre (A, T_A, u, σ) that returns $x \in \Lambda_q^u(A)$ sampled from a distribution statistically close to $D_{\Lambda_q^u,\sigma}$.

2.6 The LWE Hardness Assumption

The LWE (learning with error) hardness assumption is defined by Regev[19].

Definition 3. Consider a prime q, a positive integer n, and a distibution χ over Z_q, typically taken to be normal distribution. The input is a pair (A, v) from an unspecified challenge oracle $\bigcirc$,where $A \in Z_q^{m\times n}$ is chosen uniformly. v is chosen uniformly from Z_q^m or chosen to be $As + e$ for a uniformly chosen $s \in Z_q^n$ and a vector $e \in Z_q^m$. When v is chosen to be $As+e$ for a uniformly chosen $s \in Z_q^n$ and a vector $e \in Z_q^m$ an unspecified challenge oracle $\bigcirc$ is a noisy pseudo-random sampler $\bigcirc_s$. When v is chosen uniformly an unspecified challenge oracle $\bigcirc$ is a truly random sampler $\bigcirc_\$$.

Goal of the adversary is to distinguish with some non-negligible probability between these two cases.

Or we say that an algorithm A decides the (Z_q, n, χ)-LWE problem if $|Pr[A^{\bigcirc_s} = 1] - Pr[A^{\bigcirc_\$} = 1]|$ is non-negligible for a random $s \in Z_q^n$.

Definition 4. Consider a real parameter $\alpha = \alpha(n) \in \{0, 1\}$ and a prime q. Denote by $T = R/Z$ the group of reals [0,1) with addition modulo 1. Denote by ψ_α the distribution over T of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$ then reduced modulo 1. Denote by $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor$ the nearest integer to the real $x \in R$. We denote by $\overline{\psi}_\alpha$ the discrete distribution over Z_q of the random variable $\lfloor qX \rceil \bmod q$ where the random variable $X \in T$ has distribution ψ_α.

Lemma 3([3]). Suppose that $m > (n+1)\log_2 q + w(\log n)$ and that q is prime. Let A,B be matrices chosen uniformly in $Z_q^{n\times m}$ and let R be an $m \times m$ matrix chosen uniformly in $\{1, -1\}^{m\times m}$ mod q. Then, for all vectors w in Z_q^m, the distribution $(A, AR, R^T w)$ is statistically close to the distribution $(A, B, R^T w)$.

3 Sampling Algorithms

Let A and B be matrices in $Z_q^{n\times m}$ and let R be a matix in $\{-1, 1\}^{m\times m}$. Our construction makes use of matrices of the form $F = (AR + B) \in Z_q^{n\times 2m}$ and we will need to sample short vectors in $\Lambda_q^u(F)$ for some u in Z_q^n. This can be done either a SampleLeft or SampleRight algorithm.

3.1 SampleLeft Algorithm ([1,Theorem 17])

SampleLeft Algorithm(A, M_1, T_A, u, σ):

Inputs:

a rank n matrix A in $Z_q^{n\times m}$ and a matrix M_1 in $Z_q^{n\times m_1}$.

a "short" basis T_A of $\Lambda_q^{\perp}(A)$ and a vector $u \in Z_q^n$.

a gaussian parameter $\sigma > \|\widetilde{T_A}\|\omega(\sqrt{(\log(m+m_1))})$.

Output: Let $F_1 := (A|M_1)$. The algorithm outputs a vector $e \in Z^{m+m_1}$ sampled from a distribution statistically close to $D_{\Lambda_q^u(F_1),\sigma}$.

3.2 SampleRight Algorithm ([1,Theorem 18])

SampleRight Algorithm$(A, B, R, T_B, u, \sigma)$:

Inputs:

matrices A in $Z_q^{n\times k}$ and B in $Z_q^{n\times m}$ where B is rank n,

a matrix R in $Z_q^{k\times m}$, let $s_R := \|R\|$.

a basis T_B of $\Lambda_q^{\perp}(B)$ and a vector $u \in Z_q^n$,

a gaussian parameter $\sigma > \|\widetilde{T_B}\| s_R \omega(\sqrt{\log(m)})$.

Output: Let $F_2 := (A|AR+B)$. The algorithm outputs a vector $e \in Z^{m+k}$ sampled from a distribution statistically close to $D_{\Lambda_q^u(F_2),\sigma}$.

4 Adaptively Secure HIBE Scheme in Standard Model

The new scheme is a variant of Agarwal et al. HIBE [1], but with short public parameter. In our scheme, identity id/id_l is represented as $id/id_l = (id_1, ..., id_l)$ $= ((b_{1,1}||...||b_{1,l''}), ..., (b_{l,1}||...||b_{l,l''}))$ where id_i is l' bit string and $b_{i,j}$ is $l'/l'' = \beta$ bit string. We apply Waters'[21] idea to convert Agrawal et al. [1] selective-ID secure lattice HIBE to adaptive-ID secure HIBE. Then using blocking technique we reduce the public parameters.

4.1 The HIBE Construction

Now we describe our adaptive secure HIBE scheme as follows.

Setup(d, λ). On input a security parameter λ and a maximum hierarchy depth d, set the parameters $q, n, m, \overline{\sigma}, \overline{\alpha}$ as specified in section 4.2 below. The vectors $\overline{\sigma}$ and $\overline{\alpha}$ live in R^d and we use σ_l and α_l to refer to their t-th coordinate. Next do following.

1. Use algorithm TrapGen(q, n) to generate a matrix $A_0 \in Z_q^{n\times m}$ and a short basis T_{A_0} for $\Lambda_q^{\perp}(A_0)$ such that $\|\widetilde{T_{A_0}}\| \leq \mathrm{O}(\sqrt{n\log q})$.
2. Select $l''d+1$ uniformly random $n\times m$ matrices $A_{1,1}, ..., A_{1,l''}, ..., A_{d,1}, ...,$ $A_{d,l''}$ and $B \in Z_q^{n\times m}$.
3. Select a uniformly random n - vector $u \in Z_q^n$.
4. Output the public parameters and master key,
   PP $= A_{1,1}, ..., A_{1,l''}, ..., A_{d,1}, ..., A_{d,l''}$ *and* B, MK $= (T_{A_0})$.

Derive(PP,$(\boldsymbol{id/id_l})$, $\mathbf{SK}_{(\boldsymbol{id/id_{(l-1)}})}$). On input public parameters PP, a private key $\mathrm{SK}_{(id/id_{l-1})}$ corresponding to an identity (id/id_{l-1}) at depth $l-1$ the algorithm outputs a private key for the identity (id/id_l) at depth l. From equation (1),

$$F_{id/id_l} = (A_0|\sum_{i=1}^{l''} A_{1,i}b_{1,i} + B|...|\sum_{i=1}^{l''} A_{l,i}b_{l,i} + B) \tag{1}$$

Or $F_{id/id_l} = (F_{id/id_{l-1}}|\sum_{i=1}^{l''} A_{l,i}b_{l,i} + B)$ Given short basis $SK_{(id/id_{(l-1)})}$ for $\Lambda_q^{\perp}(F_{id/id_{l-1}})$ and F_{id/id_l} as defined in (1), we can construct short basis $SK_{(id/id_l)}$ for $\Lambda_q^{\perp}(F_{id/id_l})$ by invoking

$$S \longleftarrow \mathrm{SampleLeft}(F_{id/id_{l-1}}, \sum_{i=1}^{l''} A_{l,i}b_{l,i} + B, SK_{(Id/id_{(l-1)})}, 0, \sigma_l)$$

and output $SK_{(id/id_l)} \longleftarrow S$.

The private key corresponding to an identity $(id/id_l) = (id_1, ..., id_l)$ at depth l can be generated by PKG or any ancestor (prefix) of an identity (id/id_l) by repeatedly calling SampleLeft algorithm.

Encrypt(PP,Id,b). On input public parameters PP, an identity (id/id_l) of depth l and a message $b \in \{0, 1\}$,do following:

1. Build encryption matrix

$$F_{id/id_l} = (A_0|\sum_{i=1}^{l''} A_{1,i}b_{1,i} + B||...||\sum_{i=1}^{l''} A_{l,i}b_{l,i} + B) \in Z_q^{n\times(l+1)m}.$$

2. Choose a uniformly random vector $s \xleftarrow{R} Z_q^n$.
3. Choose ll'' uniformly random matrices $R_{i,j} \xleftarrow{R} \{-1, 1\}^{m\times m}$ for $i = 1, ..., l$ and $j = 1, ..., l''$. Define R_{id}[1] $= \sum_{i=1}^{l''} b_{1,i}R_{1,i}||...||\sum_{i=1}^{l''} b_{l,i}R_{l,i} \in Z^{m\times ll''m}$
4. Choose noise vector $x \xleftarrow{\overline{\psi}_{\alpha_l}} Z_q$, $y \xleftarrow{\overline{\psi}^m_{\alpha_l}} Z_q^m$ and $z \longleftarrow R_{id}^T y \in Z_q^{lm}$,
5. Output the ciphertext,

$$CT = \left(C_0 = u_0^T s + x + b\lfloor\frac{q}{2}\rfloor, C_1 = F_{id}^T s + \begin{bmatrix} y \\ z \end{bmatrix}\right) \quad \in Z_q \times Z_q^{(l+1)m}$$

Decrypt(PP,$\mathbf{SK}_{(\boldsymbol{id/id_l})}$,CT). On input public parameters PP, a private key SK_{id/id_l}, and a ciphertext CT $= (C_0, C_1)$, do following.

[1] In security proof, R_{id} is used to answer adversary's secret key query and also for valid challenge ciphertext, error vector has to be $\left[\begin{smallmatrix} y \\ R_{id}^T y \end{smallmatrix}\right]$.

1. Set $\tau_l = \sigma_l \sqrt{m(l+1)} w(\sqrt{log(lm)})$. Then $\tau_l \geq \|\widetilde{SK}\| w(\sqrt{log(lm)})$.
2. $e_{id} \longleftarrow SamplePre(F_{id/id_l}, SK_{(Id/id_l)}, u, \tau_l)$ Then $F_{id}e_{id} = u$ and $\|e_{id}\| \leq \tau_l \sqrt{m(l+1)}$
3. Compute $C_0 - e_{id}^T C_1 \quad \in Z_q$.
4. compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in Z. If they are close, i.e., if $|w - \lfloor \frac{q}{2} \rfloor| < \frac{q}{4}$ in Z, output 1 otherwise output 0.

During Decryption:
$w_0 = C_0 - e_{id}^T C_1 = b\lfloor \frac{q}{2} \rfloor + x - e_{id}^T \left[\begin{smallmatrix} y \\ z \end{smallmatrix}\right]$.

4.2 Parameters and Correctness

We have during decryption, $w = C_0 - e_{id}^T c_1 = b\lfloor \frac{q}{2} \rfloor + x - e_{id}^T \left[\begin{smallmatrix} y \\ z \end{smallmatrix}\right]$.

And $x - e_{id}^T \left[\begin{smallmatrix} y \\ z \end{smallmatrix}\right]$ is called error term.

Lemma 4. Norm of the error term is bounded by $[q2^{\beta}l''l^2\sigma_l m\alpha_l \omega(\sqrt{\log m}) + O(2^{\beta}l''l^2\sigma_l m^{3/2})]$.

Proof: Lemma is essentially same as lemma 32 of [1] except now R_{id} is uniformly random matrix in $\{-2^{\beta}l'', 2^{\beta}l''\}^{m \times lm}$. So now $|R_{id}|$ will be equal to $2^{\beta}l''R_{id}$. Hence error term will have extra factor $2^{\beta}l''$.

Now, for the system to work correctly we need to ensure that:

- the error term is less than $q/5$ i.e. $\alpha_l < [2^{\beta}l''l^2\sigma_l m\omega(\sqrt{\log m})]^{-1}$ and $q = \Omega(2^{\beta}l''l^2\sigma_l m^{3/2})$.
- that TrapGen can operate (i.e $m > 6n\log q$).
- That σ_l is sufficiently large for SimpleLeft and SimpleRight (i.e. $\sigma_l > \|\widetilde{T_B}\| s_R \omega(\sqrt{\log m})\ ) = 2^{\beta}l''\sqrt{l}m\omega(\sqrt{\log m})$.
- that Regev's reduction applies (i.e. $(q2^{\beta})^l > 2Q$, where Q is the number of identity queries from the adversary)

To satisfy these requirements we set the parameters $(q, m, \sigma_l, \alpha_l)$ as follows, taking n to be the security parameter:

$$m = 6n^{1+\delta}, \qquad \sigma_l = l''\sqrt{l}m\omega(\sqrt{\log n})$$

$$q = max((2Q/2^{\beta})^{1/l}, (2^{\beta}l'')^2 l^{2.5} m^{2.5}\omega(\sqrt{\log n})), \alpha_l = [(2^{\beta}l'')^2 l^{2.5} m^2 \omega(\sqrt{\log m})]^{-1} \tag{2}$$

From above requirements, we need $q = (2^{\beta}l'')^2 l^{2.5} m^{2.5}\omega(\sqrt{\log n})$.

4.3 Security Proof

Our proof of theorem will require an abort-resistant hash functions defined as follows.

Abort-Resistant Hash Functions

Definition 5. Let $H = \{\hbar : X \longrightarrow Y\}$ be family of hash functions from X to Y where $0 \in Y$. For a set of $Q+1$ inputs $\overline{x} = (x_0, x_1, ..., x_Q) \in X^{Q+1}$, define the non-abort probability of $\overline{x}$ as the quantity

$$\alpha(\overline{x}) = Pr\left[\hbar[x_0] = 0 \ \wedge \ \hbar[x_1] \neq 0 \ \wedge ... \wedge \ \hbar[x_Q] \neq 0\right]$$

where the probability is over the random choice of $\hbar$ in H.

We say that H is $(Q, \alpha_{min}, \alpha_{max})$ abort-resistance if for all $\overline{x} = (x_0, x_1, ..., x_Q) \in X^{Q+1}$ with $x_0 \notin \{x_1, ..., x_Q\}$ we have $\alpha(\overline{x}) \in [\alpha_{min}, \alpha_{max}]$.

we use the following abort-resistant hash family very similar to [1].
For a prime q let $(Z_q^{l''})^* = Z_q^{l''}$-$\{0^l\}$ and define the family

$$H : \{\hbar : ((Z_q^{l''})^*|...|(Z_q^{l''})^*) \longrightarrow (Z_q|...|Z_q)\}$$

$$\hbar(id) = \hbar(id_1|...|id_l) = (1 + \sum_{i=1}^{l''} h_{1,i}b_{1,i})|...|(1 + \sum_{i=1}^{l''} h_{l,i}b_{l,i}) \tag{3}$$

where $h_{k,i}$ and $b_{k,i}$ are defined in section 4.1.

Lemma 5. let q be a prime and $0 < Q < q$. Then the hash family H defined in (4) is $(Q, \frac{1}{q^l}(1 - \frac{Q}{q^l}), \frac{1}{q^l})$ abort-resistant.

Proof: The proof is samilar to [1]. Consider a set of $\overline{id}$ of $Q+1$ inputs $id^0, ..., id^Q$ in $(Z_q^{ll''})^*$ where $id^0 \notin \{id^1, ..., id^Q\}$ and $id^i = \{id_1, ..., id_l\}$. Since number of functions in $H = (q2^\beta)^{l''l}$ and for $i = 0, ..., Q+1$ let S_i be the set of functions $\hbar$ in H such that $\hbar(id^i) = 0$. Hence number of such functions $= |S_i| = \frac{(q2^\beta)^{l''l}}{q^l}$.

And $\frac{|S_0 \wedge S_j| \leq (q2^\beta)^{l''l}}{q^{2l}}$ for every $j > 0$. Number of functions in H such that $\hbar(id^0) = (0|...|0)$ but $\hbar(id^i) \neq 0$ for $i = 1, ..., Q.$ $= |S|$ and

$$|S| = |S_0 - (S_1 \vee ...S_Q)| \geq |S_0| - \sum_{i=1}^{Q} |S_0 \wedge S_i|$$

$$\geq \frac{(q2^\beta)^{l''l}}{q^l} - Q\frac{(q2^\beta)^{l''l}}{q^{2l}}$$

Therefore the no-abort probability of identities is atleast equal to $\frac{\frac{(q2^\beta)^{l''l}}{q^l} - \frac{Q(q2^\beta)^{l''l}}{q^{2l}}}{(q2^\beta)^{l''l}} = \frac{1}{q^l}(1 - \frac{Q}{q^l})$ Since $|S| \leq |S_0|$, so the no-abort probability is atmost $\frac{|S_0|}{(q2^\beta)^{l''l}} = \frac{1}{q^l}$.

Now we show that our lattice-based IBE construction is indistinguishable from random under a adaptive identity attack (IND-ID-CPA).

Theorem 3. The Full-HIBE scheme with parameters$(q, n, m, \overline{\sigma}, \overline{\alpha})$ as in (3) is IND-ID-CPA secure provided that the $(Z_q, n, \bar{\psi}_{\alpha_d})$-LWE assumptions holds.

Or Suppose there exists a probabilistic algorithm A (Adversary) that wins the IND-ID-CPA game with advantage ϵ, making no more than $Q \leq q^l/2$ adaptive chosen-identity queries. Then there is a probabilistic algorithm B that solves the $(Z_q, n, \overline{\psi}_\alpha)$-LWE problem in about the same time as A and with $\epsilon' \geq \epsilon/4q^l$.

Proof. Here proof is very similar to proof of theorem 25 and theorem 33 of [1]. We assume that W_i denote the event that the adversary correctly guessed the challenge bit, namely that $r = r'$ at the end of Game i. The adversary's advantage in Game i is $|Pr[W_i] - \frac{1}{2}|$. We proceed the proof in a sequence of games.

Game 0. Game 0 is the IND-ID-CPA game between an attacker against our scheme and IND-ID-CPA challenger.

Game 1. In Game 0 the challenger generates public parameters PP by choosing $ll'' + 2$ uniformly random matrices $A_0, A_{1,1}, ..., A_{l,l''}, B$ in $Z_q^{n \times m}$. In Game 1, challenger generates uniformly random matrices A_0, B same as Game 0. But challenger generates matrices $A_{k,j}, k \in [1, l]$ *and* $i \in [1, l'']$ in slightly different way. The Game 1 challenger choose $R^*_{k,d}, k \in [1, l], i \in [1, l'']$ at the set up phase and chooses ll'' random scalars $h_{k,i} \in Z_q$ for $k \in [1, l], i \in [1, l'']$. Next it constructs the matrices $A_{k,i}$ as

$$A_{k,i} \longleftarrow A_0 R_{k,i} + h_{k,i} B$$

By lemma 3, the distribution $(A_0, A_0 R^*, (R^*)^T y)$ and $(A_0, (A'_{1,1}|, ..., |A'_{l,l''}),$ $(R^*)^T y)$ are statistically close, where $R^* = (R'_{1,1}|...|R'_{l,l''}) \in Z_q^{m \times lm}$ and $A'_{k,i}, i \in [1, l''], k \in [1, l]$ are uniformly independent matrices in $Z_q^{n \times m}$. It follows that with $z = (R^*_{id})^T y$ the distributions $A_0, A_0 R^*_{1,1}, ..., A_0 R^*_{l,l''}$ and $A_0, A^*_{1,1}, ..., A^*_{l,l''}$ are statistically close. So in the attacker'view, Game 0 is same as Game 1. This shows that

$$Pr[W_0] = Pr[W_1] \tag{4}$$

Game 2. We introduce an abort event that is independent of the adversary's view and rest is same as Game 1. We will see in later part of the proof that abort event is directly related to abort-resistant family of hash functions H introduced in Lemma(6). From Lemma(6) H is a $\{Q, \alpha_{min}.\alpha_{max}\}$ abort-resistant family, where $\alpha_{min} = \frac{1}{q^l}(1 - \frac{Q}{q^l})$. For $\alpha_{min} \geq 0$ we must have $q^l > Q$. We assume $q^l \geq 2Q$ so $\alpha_{min} \geq \frac{1}{2q^l}$. For a $(Q+1)$-tuple of identities I $= (id^*, id^1, ..., id^Q)$, Game 2 challenger behaves as follows:

- The setup phase is identical to Game 1 except that the challenger also chooses a random hash function $\hbar \in H$ and keeps it to itself.
- The challenger responds to identity queries and issues the challenge ciphertext exactly as in Game 1.

- In the final guess phase, the challenger now does following:
  1. **Abort check:** The challenger checks if $H(id^*) = 0$ and $H(id^i) \neq 0$ for $i = 1, ..., Q$, where identity id^* is the challenge identity and $id^* \notin \{id^1, ..., id^Q\}$. If not, it returns random bit from $\{0, 1\}$ and game is aborted. Adversary does not know about abort condition i.e. H.
  2. **Artificial abort:** This technique was introduced by Waters [21]. Abort condition could be correlated with the adversary's query. The goal of the artificial abort step is to make the probability of abort "independent" of the adversaries queries by ensuring that in all cases its probability of abort is the maximum possible. Function $\gamma(id^*, id^1, ..., id^Q)$ *or* $\gamma(I)$ is defined in such a way that when there is no artificial abort $\gamma(I)$ is zero else $\gamma(I) = 1$. When $\gamma(I) = 1$, challenger returns random bit from $\{0, 1\}$ and game is aborted.

  If game is not aborted, the attacker outputs its guess $r' \in \{0, 1\}$ for r.

Let $\epsilon(I)$ be the probability that an abort (either real or artificial) does not happen when the adversary makes these queries. Let ϵ_{max} and ϵ_{min} be scalars such that $\epsilon(I) \in [\epsilon_{min}, \epsilon_{max}]$ for all $(Q + 1)$ tuples of identities I.

Lemma 6 (Lemma 28 of [1]). For $i = 1, 2$ let W_i be the event that $r = r'$ at the end of Game i. Then

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

Obviously $[\epsilon_{max} - \epsilon_{min}] = [\alpha_{max} - \alpha_{min}]$, when there was no artificial abort. With artificial abort, $(\epsilon_{min} - \epsilon_{max})$ is less than $\alpha_{min}|Pr[W_1] - \frac{1}{2}|$ and therefore

$$\left|Pr[W_2] - \frac{1}{2}\right| \geq 1/2.\alpha_{min}\left|Pr[W_1] - \frac{1}{2}\right| \geq (1/4q^l)\left|Pr[W_1] - \frac{1}{2}\right|. \tag{5}$$

Game 3. Game 3 differs from Game 2 how A_0 and B are chosen. In Game 3, A_0 is generated as a random matrix in $Z_q^{n\times m}$. Matrix B is generated by using algorithm TrapGen, which returns random matrix B in $Z_q^{n\times m}$ and a Trapdoor T_B for $\Lambda_q^{\perp}(B)$. From adversary's point of view, Game 2 and Game 3 are identical, hence adversary's advantage against Game 2 and Game 3 will be same. So

$$Pr[W_2] = Pr[W_3] \tag{6}$$

Game 4. In Game 4 the challenge ciphertext (C_0^*, C_1^*) is always chosen as a random independent element in $Z_q \times Z_q^{2m}$. Rest is same as Game 3. Since ciphertext is random element, hence Adversary's advantage against Game 4 is zero.

Now we have to show that Game 3 and Game 4 are computationally indistinguishable. We can show it in following way.

Suppose there exist an Adversary who can distinguish Games 3 and 4 with non-negligible then simulator can construct an algorithm which can solve LWE hard problem.

Reduction from LWE. In instance of LWE a sampling oracle $\bigcirc$ is provided. Sampling oracle $\bigcirc$ can be either truly random $\bigcirc_\$$ or a noisy pseudorandom $\bigcirc_S$ for some secret random $s \in Z_q^n$.

Instance. Simulator request from $\bigcirc$ and receives a fresh pair $(u_i, v_i) \in Z_q^n \times Z_q$ for each $i = 0, ..., m$.

Setup. B constructs the system's public parameters PP as follows:

1. The random matrix $A_0 \in Z_q^{n\times m}$ is constructed by assembling LWE sample u_i for all $i = 1, ..., m$, where i^{th} column of A is u_i.
2. Public random n-vector u_0 is the zeroth LWE sample.
3. Rest of public parameters are constructed as in Game 3.

Queries. Matrices A_0 and B are generated as in Game 3. Since B was generated using KeyGen algorithm so challenger knows trapdoor T_B. Matrices $A_{k,i}$ are constructed as in Game 1.

$$A_{k,i} = A_0 R_{k,i} + h_{k,i}B \quad for\ k = 1, ..., l \text{ and } i = 1, ..., l''. \tag{7}$$

where all the matrices $R_{k,i}$ are random in $\{1, -1\}^{m\times m}$ and $h_{k,i}$ is a random scalar coefficient in Z_q. Encryption matrix to encrypt to an identity $id = (id_1, ..., id_l)$ at depth $l \leq d$ is

$$F_{id/id_l} = (A_0|\sum_{i=1}^{l''} A_{1,i}b_{1,i} + B||...||\sum_{i=1}^{l''} A_{d,i}b_{d,i} + B) \tag{8}$$

Substituting the value of matrices $A_{k,i}$ from equation(7)

$$F_{id/id_l} = (A_0|A_0(\sum_{i=1}^{l''} R_{1,i}b_{1,i}) + B(1 + \sum_{i=1}^{l''} h_{1,i}b_{1,i})||...$$

$$||A_0(\sum_{i=1}^{l''} R_{l,i}b_{l,i}) + B(1 + \sum_{i=1}^{l''} h_{l,i}b_{l,i}))$$

Or $\quad F_{id} = (A_0|A_0R_{id} + Bh_{id}) \quad$ where $R_{id} = \sum_{i=1}^{l''} R_{1,i}b_{1,i}||...||\sum_{i=1}^{l''} R_{l,i}b_{l,i}$ and $B_{id} = B\hbar_{id} = (1 + \sum_{i=1}^{l''} h_{1,i}b_{1,i})||...||(1 + \sum_{i=1}^{l''} h_{l,i}b_{l,i})$
If h_{id} is not equal to zero then challenger responds the private key query of $id = (id^1, id^2, ..., id^l)$ by running

$$SK_{id} \longleftarrow \text{SampleRight}(A_0, B_{id}, R_{id}, T_B, 0, \sigma_l)$$

and sending SK_{id} to A. h_{id} is equal to zero will be part of abort resistant hash function.

Challenge. Adversary declares target identity $id^* = (id_1, id_2, ..., id_l)$ and message bit $b^* \in \{0, 1\}$. Simulator B creates challenge ciphertext for the target identity as follows:

1. Let $v_0, ..., v_m$ be entries from LWE instance. Set
$$v^* = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_m \end{pmatrix} \in Z_q^m$$
2. Blind the message bit by letting
$$C_0^* = v_0 + b^* \lfloor \frac{q}{2} \rceil \in Z_q$$
3. Let
$$R_{id^*} = (R_1^* | ... | R_l^*)$$
where
$$R_{j^*} = \sum_{j=1}^{l''} R_{i,j} b_{i,j}$$
and set
$$C_1^* = \begin{pmatrix} v^* \\ (R_{id^*})^T v^* \end{pmatrix} \in Z_q^{m+lm}$$
4. Choose a random bit $r \leftarrow \{0, 1\}$. If $r = 0$ send $CT^* = (C_0^*, C_1^*)$ to the adversary. If $r = 1$ choose a random $(C_0, C_1) \in Z_q \times Z_q^{m+lm}$ and send (C_0, C_1) to the adversary.

When the LWE oracle is pseudorandom then $F_{id^*} = (A_0 | A_0 \overline{R}_{id^*})$ since $h_{id^*} = 0$ and
$$v^* = A_0^T s + y$$
for some random noise vector $y \in Z_q^m$ distributed as $\bar{\psi}_\alpha^m$. Therefore
$$C_1^* = \begin{pmatrix} A_0^T s + y \\ (A_0 R_{id^*})^T s + (R_{id^*})^T y \end{pmatrix} = (F_{id^*})^T s + \begin{pmatrix} y \\ (R_{id^*})^T y \end{pmatrix}$$
Above C_1^* is a valid C_1 part of challenge ciphertext. Again $C_0^* = u_0^T + x + b^* \lfloor \frac{q}{2} \rceil$ is also a valid C_0 part of challenge ciphertext. Therefore (C_0^*, C_1^*) is valid challenge ciphertext as in Game 3.

When LWE oracle is random oracle, v_0 is uniform in Z_q and v^* is uniform in Z_q^m. Therefore challenge ciphertext is always uniform in $Z_q \times Z_q^{2m}$ as in Game 4.

Guess. Adversary is again allowed to make private key extraction query as in Game 3 except prefix of id^*. Then Adversary guess if it is valid ciphertext (Game 3) or random string (Game 4). Hence simulator's advantage in solving LWE hard problem is same as Adversary's advantage in distinguishing valid ciphertext (Game 3) and random string (Game 4). Since $Pr[W_4] = 1/2$, So
$$|Pr[W_3] - Pr[W_4]| = |Pr[W_3] - \frac{1}{2}| \leq \text{LWE-adv(B)} \tag{9}$$
Combining equation (4),(5),(6) and (9), we get
$$|Pr[W_0] - \frac{1}{2}| \leq 4q^l \text{ LWE-adv(B)}$$

5 New Full-IBE Scheme in Standard Model

The new scheme is a variant of Agarwal et al. IBE [1], but with short public parameter. In Agrawal et al. IBE scheme identities are represented as l-bit string. Because of this representation, scheme requires l $n \times m$ matrices. In our scheme, identity id is represented as id $= (b_1, ..., b_{l'})$, where each b_i is an $l/l' = \beta$ bit string.

5.1 The New Full-IBE Construction

Now we describe our new Full-IBE Scheme in the standard model as follows.

Setup(λ). On input a security parameter λ, set the parameters q, n, m, σ, α as specified in section 5.2 below. Next do following.

1. Use algorithm TrapGen(q, n) to generate a matrix $A_0 \in Z_q^{n\times m}$ and a short basis T_{A_0} for $\Lambda_q^{\perp}(A_0)$ such that $\|\widetilde{T_{A_0}}\| \leq \mathrm{O}(\sqrt{n \log q})$.
2. Select $l' + 1$ uniformly random $n \times m$ matrices $A_1, A_2, ..., A_{l'}, B \in Z_q^{n\times m}$.
3. Select a uniformly random n - vector $u \in Z_q^n$.
4. Output the public parameters and master key, PP $= (A_1, A_2, ..., A_{l'}, B, u)$, MK $= (T_{A_0})$.

Extract(PP,MK,Id). On input public parameters PP, a master secret key MK, and an identity id $= (b_1, ..., b_{l'})$, where each b_i is an $l/l' = \beta$ bit string.

1. Let $A_{id} = B + \sum_{i=1}^{l'} b_i A_i \quad \in Z_q^{n\times m}$.
2. Sample $e \in Z_q^{2m}$ as $e \longleftarrow SampleLeft(A_0, A_{id}, T_{A_0}, u, \sigma)$.
3. Output $SK_{id} = e \quad \in Z^{2m}$.

Let $F_{id} = (A_0|A_{id})$, then $F_{id}.e = u$ in Z_q and e is distributed as $D_{\Lambda_q^u(F_{id}),\sigma}$ by lemma 2.

Encrypt(PP,Id,b). On input public parameters PP, an identity id, and a message $b \in \{0, 1\}$,do following:

1. Let $A_{id} = B + \sum_{i=1}^{l'} b_i A_i \quad \in Z_q^{n\times m}$ and $F_{id} = (A_0|A_{id}) \ \in Z_q^{n\times 2m}$.
2. Choose a uniformly random $s \xleftarrow{R} Z_q^n$.
3. Choose l' uniformly random matrices $R_i \xleftarrow{R} \{-1, 1\}^{m\times m}$ for $i = 1, ..., l'$ and define R_{id}[2] $= \sum_{i=1}^{l'} b_i R_i \quad \in \{-l'(2^\beta - 1), ..., l'(2^\beta - 1)\}$.
4. Choose noise vectors $x \xleftarrow{\overline{\Psi}_\alpha} Z_q$, $y \xleftarrow{\overline{\Psi}_\alpha^m} Z_q^m$ and $z \longleftarrow R_{id}^T y \in Z_q^m$,
5. Set $C_0 \longleftarrow u^T s + x + b\lfloor \frac{q}{2} \rfloor \in Z_q$ and $C_1 \longleftarrow F_{id}^T s + \left[\begin{smallmatrix} y \\ z \end{smallmatrix}\right] \quad \in Z_q^{2m}$ and .
6. Output the ciphertext CT $= (C_0, C_1) \ \in Z_q \times Z_q^{2m}$.

[2] In security proof, R_{id} is used to answer adversary's secret key query and also for valid challenge ciphertext, error vector has to be $\left[\begin{smallmatrix} y \\ R_{id}^T y \end{smallmatrix}\right]$.

Decrypt(PP,SK$_{id}$,CT). On input public parameters PP, a private key $SK_{id} = e_{id}$, and a ciphertext CT $= (C_0, C_1)$, do following.

1. Compute $w \longleftarrow C_0 - e_{id}^T C_1 \quad \in Z_q$. If they are close, i.e., if $|w - \lfloor \frac{q}{2} \rfloor| < q/4$ in Z, output 1 otherwise output 0.

During Decryption:
$w_0 = C_0 - e_{id}^T C_1 = b\lfloor \frac{q}{2} \rfloor + x - e_{id}^T \left[\begin{smallmatrix} y \\ z \end{smallmatrix} \right]$.

5.2 Parameters and Correctness

We have during decryption, $w = C_0 - e_{id}^T c_1 = b\lfloor \frac{q}{2} \rfloor + x - e_{id}^T \left[\begin{smallmatrix} y \\ z \end{smallmatrix} \right]$.

And $x - e_{id}^T \left[\begin{smallmatrix} y \\ z \end{smallmatrix} \right]$ is called error term.

Lemma 7. For an l-bit identity id $= (b_1, ..., b_{l'})$, where each b_i is an $l/l' = \beta$ bit string. Norm of the error term is bounded by $q\sigma 2^\beta l' m \alpha \omega(\sqrt{\log m}) + O(\sigma 2^\beta l' m^{3/2})$.

Proof: The proof is identical to the proof of Lemma 22 in [1] except that matrix R is replaced by $R_{id} = \sum_{i=1}^{l'} b_i A_i$. Since $\|R_{id}\| \leq \sum_{i=1}^{l'} \|b_i\| \|A_i\|$ and by [1,theorem 15],$\|R\| \leq O(\sqrt{m})$.

So $\|R_{id}\| \leq O(2^\beta l' \sqrt{m})$. This leads to the extra factor $2^\beta l'$ in the error bound. Now, for the system to work correctly we need to ensure that:

- the error term is less than $q/5$ i.e. $\alpha < [\sigma 2^\beta l' m \alpha \omega(\sqrt{\log m})]^{-1}$ and $q = \Omega(\sigma 2^\beta l' m^{3/2})$.
- that TrapGen can operate (i.e $m > 6n \log q$).
- That σ is sufficiently large for SimpleLeft and SimpleRight (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 Regev's reduction applies (i.e. $q > 2Q$, where Q is the number of identity queries from the adversary)

To satisfy these requirements we set the parameters (q, m, σ, α) as follows, taking n to be the security parameter:

$$m = 6n^{1+\delta}, \qquad \sigma = 2^\beta l' \sqrt{m} \omega(\sqrt{\log n})$$

$$q = max(2Q, m^{2.5}(2^\beta l')^2 \omega(\sqrt{\log n})), \qquad \alpha = [2^\beta l' m \omega(\sqrt{\log m})]^{-1}. \tag{10}$$

From above requirements, we need $q = m^{2.5}(2^\beta l')^2 \omega(\sqrt{\log n})$. But in [1], required value of $q = m^{2.5} l^2 \omega(\sqrt{\log n})$. In this scheme value of q is increased by $(2^\beta \frac{l'}{l})^2 = (\frac{2^\beta}{\beta})^2$. This means that when public parameters are reduced by factor β, the value of q is increased by $(\frac{2^\beta}{\beta})^2$ or number of bits in q is increased by $(\beta - lg(\beta))^2$.

5.3 Efficiency

Here efficiency analysis is similar to [9]. Difference between our scheme and scheme [1] is computation of A_{id} and R_{id}. Rest of the algorithm for key generation, encryption and decryption algorithms etc are same. Let $|Z_q|$ be the size of the representation of an element of Z_q. We assume that cost of adding two $n \times m$ matrices is approximately equal to $nm|Z_q|$. Cost of computing A_{id} is adding two $n \times m$ matrices and l' multiplication where each multiplication is multiplication of (l/l')-bit string and $n \times m$ matrix. On an average, cost of each such multiplication will be $l/2l'$ addition and $(l/l'-1)$ doubling. Hence, the total cost of computing A_{id} is l/l' addition and $(l-l')$ doubling. This cost is equal to $cnm(\frac{3l}{2} - l')|Z_q|$ for some constant c. This cost is minimum when $l' = l$ (as in [1]). Minimum value is $cnm\frac{l}{2}|Z_q|$. Maximum value is less than $cnm\frac{3l}{2}|Z_q|$. Cost of computing $F_{id}^T s$ is equal to $dnmn|Z_q|$ for some constant d. Cost of encryption is equal to $cnm(\frac{3l}{2}-l')|Z_q|+dnmn|Z_q|$. Cost of encryption in IBE [1] is equal to $cnm(\frac{l}{2})|Z_q| + dnmn|Z_q|$. Value of q is less than poly(n) assume n^5. If q is more than 512 bit then value of n is atleast 2^{100}, which is much greater than size of identity $l(160)$. So cost of encryption is $enmn|Z_q|$ for some constant e, which does not depend on l'. There is no effect of l' on computation of A_{id}. Similarly There is no effect of l' on computation of R_{id}. Hence there is no effect of l' on cost of key generation and encryption. Decryption algorithm is same in both scheme. Computational cost increases because of increase in value of q or size of $|Z_q|$.

5.4 Space/Time Trade-Off

Our scheme reduces public size by a factor β. The relative decrease in amount of space (expressed in percentage) required to store the public parameter in case of our scheme with respect to scheme [1] is equal to $\frac{l-l'}{l}$. Our scheme reduces public size by a factor of β at the cost of increasing value of q by a factor of $(\frac{2^\beta}{\beta})^2$ with same security as [1]. By making same security as [1], new q or q' is $q(\frac{2^\beta}{\beta})^2$. Size of $Z_{q'} = |Z_{q'}| = |Z_q| + (\beta - lg(\beta))^2$. Relative increase in encryption cost in case of our scheme with respect to [1] is $\frac{|Z_{q'} - Z_q|}{|Z_q|} = \frac{(\beta - lg(\beta))^2}{|Z_q|}$.

In table 1, we give the results for $l = 160$ and $|Z_q| = 512$ for different values of l' ranging from 8 to 64. Overall, we suggest $l' = 16$ to be good choice for implementing the protocol.

Table 1. Relative decrease in space and relative increase in time for different values of l'

l'	Relative decrease in space	Relative increase in time
8	95	48
16	90	8.71
32	80	1.40
64	60	0.27

5.5 Security Proof

Our proof of theorem will require an abort-resistant hash functions defined as follows.

Abort-Resistant Hash Functions([1])

Definition 6. Let $H = \{\hbar : X \longrightarrow Y\}$ be family of hash functions from X to Y where $0 \in Y$. For a set of $Q+1$ inputs $\overline{x} = (x_0, x_1, ..., x_Q) \in X^{Q+1}$, define the non-abort probability of $\overline{x}$ as the quantity

$$\alpha(\overline{x}) = Pr\left[\hbar[x_0] = 0 \ \wedge \ \hbar[x_1] \neq 0 \ \wedge ... \wedge \ \hbar[x_Q] \neq 0\right]$$

where the probability is over the random choice of $\hbar$ in H.

We say that H is $(Q, \alpha_{min}, \alpha_{max})$ abort-resistance if for all $\overline{x} = (x_0, x_1, ..., x_Q) \in X^{Q+1}$ with $x_0 \notin \{x_1, ..., x_Q\}$ we have $\alpha(\overline{x}) \in [\alpha_{min}, \alpha_{max}]$.

we use the following abort-resistant hash family very similar to [1]. For a prime q let $(Z_q^{l'})^* = Z_q^{l'} - \{0^l\}$ and define the family

$$H : \{\hbar : ((Z_q^{l'})^*) \longrightarrow (Z_q)\}$$

$$\hbar(id) = (1 + \sum_{i=1}^{l'} h_i b_i) \in Z_q \tag{11}$$

where h_i and b_i are defined in section 4.1.

Lemma 8. Let q be a prime and $0 < Q < q$. Then the hash family H defined in (4) is $(Q, \frac{1}{q}(1 - \frac{Q}{q}), \frac{1}{q})$ abort-resistant.

Proof: The proof is very similar to [1]. Consider a set of $\overline{id}$ of $Q+1$ inputs $id_0, ..., id_Q$ in $(Z_q^{l'})^*$ where $id_0 \notin \{id_1, ..., id_Q\}$. For $i = 0, ..., Q+1$ let S_i be the set of functions $\hbar$ in H such that $\hbar(id_i) = 0$. We know that number of such functions $= |S_i| = \frac{(q2^\beta)^{l'}}{q}$.

And $|S_0 \wedge S_j| \leq \frac{(q2^\beta)^{l'}}{q^2}$ for every $j > 0$. Number of functions in H such that $\hbar(id_0) = 0$ but $\hbar(id_i) \neq 0$ for $i = 1, ..., Q$. $= |S|$ and

$$|S| = |S_0 - (S_1 \vee, ..., S_Q)| \geq |S_0| - \sum_{i=1}^{Q} |S_0 \wedge S_i|$$

$$\geq \frac{(q2^\beta)^{l'}}{q} - Q\frac{(q2^\beta)^{l'}}{q^2}$$

Since number of functions in $H = (q2^\beta)^{l'}$, therefore the no-abort probability of identities is atleast equal to $\frac{\frac{(q2^\beta)^{l'}}{q} - Q\frac{(q2^\beta)^{l'}}{q^2}}{(q2^\beta)^{l'}} = \frac{1}{q}(1 - \frac{Q}{q})$ Since $|S| \leq |S_0|$, so the no-abort probability is atmost $\frac{|S_0|}{(q2^\beta)^{l'}} = \frac{1}{q}$.

Now we show that our lattice-based IBE construction is indistinguishable from random under a adaptive identity attack (IND-ID-CPA).

Theorem 4. The Full-HIBE Scheme with parameters$(q, n, m, \overline{\sigma}, \overline{\alpha})$ as in (10) is IND-ID-CPA secure provided that the $(Z_q, n, \bar{\psi}_{\alpha_d})$-LWE assumptions hold.

Or Suppose there exists a probabilistic algorithm A (Adversary) that wins the IND-ID-CPA game with advantage ϵ, making no more than $Q \leq q/2$ adaptive chosen-identity queries, then there is a probabilistic algorithm B that solves the $(Z_q, n, \overline{\psi}_\alpha)$-LWE problem in about the same time as A and with $\epsilon' \geq \epsilon/(4q)$.

Proof. Since limits of no-abort probability (Lemma 5) of identity is same as lemma 27 of [1] so security proof will be same as security proof of [1,theorem 25].

6 Conclusion

We have shown that by converting selective-ID HIBE to adaptive-ID HIBE security degradation is exponential in number of levels. In our efficient lattice based IBE scheme we have also shown that there is an interesting trade-off between reducing the public parameter size and increase in the value of q (computational cost). The main open problem in the construction of lattice based IBE protocols is to reduce the public parameter size without increasing the value of q(computational cost).

Acknowledgments. We would like to thank anonymous reviewer and PC chairs for their useful comments.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Agrawal, S., Boyen, X.: Identity-based encryption from lattices in the standard model (2009) (manuscript), `http://www.cs.stanford.edu/xb/ab09/`
3. Alwen, J., Peikert, C.: Generating Shorter Bases for Hard Random Lattices. In: STACS 2009, pp. 75–86 (2009)
4. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
5. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
6. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–219. Springer, Heidelberg (2001)
7. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
8. Cash, D., Hofheinz, D., Kiltz, E.: How to Delegate a Lattice Basis. IACR Cryptology ePrint Archive 2009, p. 351 (2009)

9. Chatterjee, S., Sarkar, P.: Trading Time for Space: Towards an Efficient IBE Scheme with Short(er) Public Parameters in the Standard Model. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 424–440. Springer, Heidelberg (2006)
10. Chatterjee, S., Sarkar, P.: HIBE With Short Public Parameters Without Random Oracle. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 145–160. Springer, Heidelberg (2006)
11. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: IMA Int. Conf. 2001, pp. 360–363 (2001)
12. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM J. Comput. 38(1), 97–139 (2008)
13. Gentry, C., Silverberg, A.: Hierarchical ID-Based Cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
14. Horwitz, J., Lynn, B.: Toward Hierarchical Identity-Based Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
15. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptographic Perspective. In: Engineering and Computer Science. The Kluwer International Series, vol. 671. Kluwer Academic Publishers, Boston (2002)
16. Naccache, D.: Secure and Practical Identity-Based Encryption. IACR Cryptology ePrint Archive 2005, p. 369 (2005)
17. Peikert, C.: Bonsai trees (or, arboriculture in lattice-based cryptography). Cryptology ePrint Archive, Report 2009/359 (2009)
18. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197–206 (2008)
19. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93 (2005)
20. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
21. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)