Efficient Lattice (H)IBE in the Standard Model*

Shweta Agrawal
University of Texas, Austin

Dan Boneh †
Stanford University

Xavier Boyen
Université de Liège, Belgium

April 26, 2010

Abstract

We construct an efficient identity based encryption system based on the standard learning with errors (LWE) problem. Our security proof holds in the standard model. The key step in the construction is a family of lattices for which there are two distinct trapdoors for finding short vectors. One trapdoor enables the real system to generate short vectors in all lattices in the family. The other trapdoor enables the simulator to generate short vectors for all lattices in the family except for one. We extend this basic technique to an adaptively-secure IBE and a Hierarchical IBE.

1 Introduction

Identity-Based Encryption (IBE) provides a public-key encryption mechanism where a public key is an arbitrary string such as an email address or a telephone number. The corresponding private key can only be generated by a Private-Key Generator (PKG) who has knowledge of a master secret. Identity-based encryption was first proposed by Shamir [35], however, it is only recently that practical implementations were proposed. Boneh and Franklin [11] define a security model for identity-based encryption and give a construction based on the Bilinear Diffie-Hellman (BDH) problem. Cocks [19] describes a construction using quadratic residues modulo a composite (see also [12]) and Gentry et al. [22] give a construction using lattices. The security of all these systems requires cryptographic hash functions that are modeled as random oracles.

For pairing-based systems, the structure of pairing groups enabled several secure IBE systems in the standard model [16, 8, 9, 38, 23, 39]. For systems based on quadratic residuosity it is still not known how to build a secure IBE in the standard model.

In this paper we focus on lattice-based IBE. Cash et al. [18, 17, 31], and Agrawal et al. [3] recently showed how to construct secure IBE in the standard model from the learning with errors (LWE) problem [34]. Their constructions view an identity as a sequence of bits and then assign a matrix to each bit. The resulting systems, while quite elegant, are considerably less efficient than the underlying random-oracle system of [22] on which they are built.

*This paper combines preliminary results that appeared in Eurocrypt'10 [1] and PKC'10 [14].

†Supported by NSF and the Packard Foundation.

1.1 Our Results

We construct a lattice-based IBE in the standard model whose performance is comparable to the performance of the random-oracle system from [22]. In particular, we process identities as one chunk rather than bit-by-bit resulting in lattices whose dimension is similar to those in the random oracle system. This construction also gives an efficient chosen ciphertext secure lattice-based public-key encryption (PKE) system via a generic selective-IBE to CCA-PKE transformation [15, 13, 10].

Lattices in our system are built from two parts called "right" and "left" lattices. A trapdoor for the left lattice is used as the master secret in the real system and enables one to generate private keys for all identities. A trapdoor for the right lattice is only used in the proof of selective security and enables the simulator to generate private keys for all identities except for one. We use a "low norm" randomization matrix R to ensure that an attacker cannot distinguish between the real world and a simulation.

In pairing-based IBE systems one uses large groups G and therefore identities can be encoded as integers in the range $1 \dots |G|$. In contrast, lattice systems are typically defined over a relatively small field $\mathbb{Z}_q$ and consequently encoding identities as integers in $1 \dots q$ would result in too few identities for the system. Instead, we represent identities as matrices in $\mathbb{Z}_q^{n \times n}$ for some n. More precisely, we represent identities as elements in $\mathbb{Z}_q^n$ (for a total of q^n identities) and then use an encoding function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ to map identities to matrices. Our security proof requires that for all $\mathsf{id}_1 \neq \mathsf{id}_2$ the matrix $H(\mathsf{id}_1) - H(\mathsf{id}_2) \in \mathbb{Z}_q^{n \times n}$ is invertible. We present an encoding function H that has this property and expect this encoding to be useful in other lattice-based constructions. A similar function H was developed by Cramer and Damgard [20] in an entirely different context.

Full IBE. In Section 7 we show that our base construction extends to an adaptively-secure IBE using a lattice analog of the Waters IBE [38]. Our base construction requires that the underlying field $\mathbb{Z}_q$ satisfy $q > Q$ where Q is the number of private key queries issued by the adversary. This requirement can be relaxed using the framework of Boyen [14].

Hierarchical IBE (HIBE). In Section 8 we show how to extend our base IBE to an HIBE using the basis delegation technique from [17, 31]. The construction assigns a matrix to each level of the hierarchy and the resulting lattice dimension is linear in the recipient identity's depth. Since we do not process identities bit-by-bit we obtain an efficient HIBE where the lattice dimension is much smaller than in [17, 31]. We note that a recent result of [2] uses a different basis delegation mechanism to construct an improved HIBE where the lattice dimension is fixed for the entire hierarchy.

2 Preliminaries

Notation. Throughout the paper we say that a function $\epsilon : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is negligible if $\epsilon(n)$ is smaller than all polynomial fractions for sufficiently large n. We say that an event happens with overwhelming probability if it happens with probability at least $1 - \epsilon(n)$ for some negligible function ϵ. We say that integer vectors $v_1, \dots, v_n \in \mathbb{Z}^m$ are $\mathbb{Z}_q$-linearly independent if they are linearly independent when reduced modulo q.

2.1 IBE and Hierarchical IBE

Recall that an Identity-Based Encryption system (IBE) consists of four algorithms [35, 11]: Setup, $\mathsf{Extract}$, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$. The Setup algorithm generates system parameters, denoted by PP, and a master key MK. The $\mathsf{Extract}$ algorithm uses the master key to extract a private key corresponding to a given identity. The encryption algorithm encrypts messages for a given identity (using the system parameters) and the decryption algorithm decrypts ciphertexts using the private key.

In a Hierarchical IBE [27, 24], identities are vectors, and there is a fifth algorithm called Derive. A vector of dimension ℓ represents an identity at depth ℓ. Algorithm Derive takes as input an identity $\mathsf{id} = (\mathsf{I}_1, \ldots, \mathsf{I}_\ell)$ at depth ℓ and the private key $\mathsf{SK}_{\mathsf{id}|\ell-1}$ of the parent identity $\mathsf{id}_{|\ell-1} = (\mathsf{I}_1, \ldots, \mathsf{I}_{\ell-1})$ at depth $\ell - 1 \geq 0$. It outputs the private key $\mathsf{SK}_{\mathsf{id}}$ for identity id. We sometimes refer to the master key as the private key at depth 0, given which the algorithm Derive performs the same function as $\mathsf{Extract}$. The Setup algorithm in an HIBE scheme takes the maximum depth of the hierarchy as input.

Selective and Adaptive ID Security. The standard IBE security model of [11] defines the indistinguishability of ciphertexts under an adaptive chosen-ciphertext and chosen-identity attack (IND-ID-CCA2). A weaker notion of IBE security given by Canetti, Halevi, and Katz [16] forces the adversary to announce ahead of time the public key it will target, which is known as a selective-identity attack (IND-sID-CCA2).

As with regular public-key encryption, we can deny the adversary the ability to ask decryption queries (for the target identity), which leads to the weaker notions of indistinguishability of ciphertexts under an adaptive chosen-identity chosen-plaintext attack (IND-ID-CPA) and under a selective-identity chosen-plaintext attack (IND-sID-CPA) respectively.

Security Game. We define IBE and HIBE selective security using a game that captures a strong privacy property called *indistinguishable from random* which means that the challenge ciphertext is indistinguishable from a random element in the ciphertext space. This property implies both semantic security and recipient anonymity, and also implies that the ciphertext hides the public parameters (PP) used to create it. This can make the IBE more resistant to subpoenas since an observer cannot tell from the ciphertext which authority holds the corresponding master secret. For a security parameter λ, we let $\mathcal{M}_\lambda$ denote the message space and let $\mathcal{C}_\lambda$ denote the ciphertext space. The game, for a hierarchy of maximum depth d, proceeds as follows.

Init: The adversary is given the maximum depth of the hierarchy d and outputs a target identity $\mathsf{id}^* = (\mathsf{I}_1^*, \ldots, \mathsf{I}_k^*), k \leq d$.

Setup: The challenger runs $\mathsf{Setup}(1^\lambda, 1^d)$ (where $d = 1$ for IBE) and gives the adversary the resulting system parameters PP. It keeps the master key MK to itself.

Phase 1: The adversary issues queries $q_1, \ldots, q_m$ where the i-th query q_i is a query on id_i, where $\mathsf{id}_i = (\mathsf{I}_1, \ldots, \mathsf{I}_u)$ for some $u \leq d$. We require that id_i is not a prefix of id^*, (i.e., it is not the case that $u \leq k$ and $\mathsf{I}_i = \mathsf{I}_i^*$ for all $i = 1, \ldots, u$). The challenger responds by running algorithm $\mathsf{Extract}$ to obtain a private key d_i for the public key id_i. It sends d_i to the adversary.

All queries may be made adaptively, that is, the adversary may ask q_i with knowledge of the challenger's responses to $q_1, \ldots, q_{i-1}$.

Challenge: Once the adversary decides that Phase 1 is over it outputs a plaintext $M \in \mathcal{M}_\lambda$ on which it wishes to be challenged. The challenger picks a random bit $r \in \{0,1\}$ and a random ciphertext $C \in \mathcal{C}_\lambda$. If $r = 0$ it sets the challenge ciphertext to $C^* := \mathsf{Encrypt}(\mathsf{PP}, \mathsf{id}^*, M)$. If $r = 1$ it sets the challenge ciphertext to $C^* := C$. It sends C^* as the challenge to the adversary.

Phase 2: The adversary issues additional adaptive queries $q_{m+1}, \ldots, q_n$ where q_i is a private-key extraction query on id_i, where id_i is not a prefix of id^*. The challenger responds as in Phase 1.

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

We refer to such an adversary $\mathcal{A}$ as an INDr–sID-CPA adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking an IBE or HIBE scheme $\mathcal{E}$ as

$$\mathrm{Adv}_{d,\mathcal{E},\mathcal{A}}(\lambda) = \left| \Pr[r = r'] - 1/2 \right|$$

The probability is over the random bits used by the challenger and the adversary.

Definition 1. We say that an IBE or a depth d HIBE system $\mathcal{E}$ is selective-identity, indistinguishable from random if for all INDr–sID-CPA PPT adversaries $\mathcal{A}$ we have that $\mathrm{Adv}_{d,\mathcal{E},\mathcal{A}}(\lambda)$ is a negligible function. We abbreviate this by saying that $\mathcal{E}$ is INDr–sID-CPA secure for depth d.

Finally, we define the adaptive-identity counterparts to the above notions by removing the Init phase from the attack game, and allowing the adversary to wait until the Challenge phase to announce the identity id^* it wishes to attack. The adversary is allowed to make arbitrary private-key queries in Phase 1 and then choose an arbitrary target id^*. The only restriction is that he did not issue a private-key query for id^* or a prefix of id^* during phase 1. The resulting security notion is defined using the modified game as in Definition 1, and is denoted INDr–ID-CPA.

2.2 Statistical Distance

Let X and Y be two random variables taking values in some finite set Ω. Define the *statistical distance*, denoted $\Delta(X;Y)$, as

$$\Delta(X;Y) := \frac{1}{2} \sum_{s \in \Omega} \left| \Pr[X = s] - \Pr[Y = s] \right|$$

We say that X is δ-uniform over Ω if $\Delta(X; U_\Omega) \le \delta$ where U_Ω is a uniform random variable over Ω.

Let $X(\lambda)$ and $Y(\lambda)$ be ensembles of random variables. We say that X and Y are statistically close if $d(\lambda) := \Delta(X(\lambda); Y(\lambda))$ is a negligible function of λ.

2.3 Integer Lattices

We will be using integer lattices, namely discrete subgroups of $\mathbb{Z}^m$. The specific lattices we use contain $q\mathbb{Z}^m$ as a sub-lattice for some prime q that is much smaller than the determinant of the lattice.

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

$$\begin{array}{rcl} \Lambda_q(A) & := & \{\ e \in \mathbb{Z}^m \quad \text{s.t.} \quad \exists s \in \mathbb{Z}_q^n \text{ where } A^\top s = e \pmod q\ \} \\ \Lambda_q^\perp(A) & := & \{\ e \in \mathbb{Z}^m \quad \text{s.t.} \quad A\, e = 0 \pmod q\ \} \\ \Lambda_q^u(A) & := & \{\ e \in \mathbb{Z}^m \quad \text{s.t.} \quad A\, e = u \pmod q\ \} \end{array}$$

Observe that if $t \in \Lambda_q^u(A)$ then $\Lambda_q^u(A) = \Lambda_q^\perp(A) + t$ and hence $\Lambda_q^u(A)$ is a shift of $\Lambda_q^\perp(A)$.

2.4 The Gram-Schmidt Norm of a Basis

Let S be a set of vectors $S = \{s_1, \ldots, s_k\}$ in $\mathbb{R}^m$. 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 $1 \le i \le k$.
- $\tilde{S} := \{\tilde{s}_1, \ldots, \tilde{s}_k\} \subset \mathbb{R}^m$ denotes the Gram-Schmidt orthogonalization of the vectors $s_1, \ldots, s_k$ taken in that order.

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

Micciancio and Goldwassser [29] showed that a full-rank set S in a lattice Λ can be converted into a basis T for Λ with an equally low Gram-Schmidt norm.

Lemma 3 ([29, 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 $S = \{s_1, \ldots, s_m\}$ in Λ, returns a basis T of Λ satisfying*

$$\|\widetilde{T}\| \le \|\widetilde{S}\| \quad \textit{and} \quad \|T\| \le \|S\|\sqrt{m}/2$$

Ajtai [4] showed how to sample an essentially uniform matrix $A \in \mathbb{Z}_q^{n\times m}$ with an associated basis S_A of $\Lambda_q^\perp(A)$ with low Gram-Schmidt norm. We use an improved sampling algorithm from Alwen and Peikert [6]. The following follows from Theorem 3.2 of [6] taking $\delta := 1/3$.

Theorem 4. *Let $q \ge 3$ be odd and $m := \lceil 6n \log q \rceil$.*
There is a probabilistic polynomial-time algorithm $\mathsf{TrapGen}(q,n)$ that outputs a pair $(A \in \mathbb{Z}_q^{n\times m},\ S \in \mathbb{Z}^{m\times m})$ such that A is statistically close to a uniform matrix in $\mathbb{Z}_q^{n\times m}$ and S is a basis for $\Lambda_q^\perp(A)$ satisfying

$$\|\widetilde{S}\| \le O(\sqrt{n \log q}\,) \quad \textit{and} \quad \|S\| \le O(n \log q)$$

with all but negligible probability in n.

Notation: We let $\sigma_{\mathrm{TG}} := O(\sqrt{n \log q}\,)$ denote the maximum (w.h.p) Gram-Schmidt norm of a basis produced by $\mathsf{TrapGen}(q,n)$.

Peikert [31, Lemma 3.2] shows how to construct a basis for $\Lambda_q^\perp(A|B|C)$ from a basis for $\Lambda_q^\perp(B)$.

Theorem 5. *For $i = 1,2,3$ let A_i be a matrix in $\mathbb{Z}_q^{n\times m_i}$ and let $A := (A_1|A_2|A_3)$. Let T_2 be a basis of $\Lambda_q^\perp(A_2)$. There is deterministic polynomial time algorithm $\mathsf{ExtendBasis}(A_1, A_2, A_3, T_2)$ that outputs a basis T for $\Lambda_q^\perp(A)$ such that $\|\widetilde{T}\| = \|\widetilde{T_2}\|$.*

We will also need the following simple lemma about the effect of matrix multiplication on the Gram-Schmidt norm.

Lemma 6. *Let R be a matrix in $\mathbb{R}^{\ell\times m}$ and $S=\{s_1,\ldots,s_k\}\subset\mathbb{R}^m$ a linearly independent set. Let $S_R:=\{Rs_1,\ldots,Rs_k\}$. Then*

$$\|\widetilde{S_R}\| \leq \max_{1\leq i\leq k} \|R\tilde{s}_i\|$$

Proof. We show that for all $i=1,\ldots,k$ the i-th Gram-Schmidt vector of S_R has L_2 norm less than $\|R\tilde{s}_i\|$. This will prove the lemma.

For $i\in\{1,\ldots,k\}$ let $V:=\mathrm{span}_{\mathbb{R}}(Rs_1,\ldots,Rs_{i-1})$. Set $v:=s_i-\tilde{s}_i$. Then $v\in\mathrm{span}_{\mathbb{R}}(s_1,\ldots,s_{i-1})$ and therefore $Rv\in V$. Let u be the projection of $R\tilde{s}_i$ on V and let $z:=R\tilde{s}_i-u$. Then z is orthogonal to V and

$$Rs_i = Rv + R\tilde{s}_i = Rv + u + z = (Rv+u) + z\ .$$

By construction, $Rv+u\in V$ and hence, since z is orthogonal to V, this z must be the i-th Gram-Schmidt vector of S_R. Since z is the projection of $R\tilde{s}_i$ on $V^\perp$ we obtain that $\|z\|\leq\|R\tilde{s}_i\|$. Hence, for all $i=1,\ldots,k$ the i-th Gram-Schmidt vector of S_R has L_2 norm less than $\|R\tilde{s}_i\|$ which proves the lemma. $\square$

2.5 Discrete Gaussians

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

$\rho_{\sigma,c}(x) \ = \exp\left(-\pi\frac{\|x-c\|^2}{\sigma^2}\right)$: a Gaussian-shaped function on $\mathbb{R}^m$ with center c and parameter σ,

$\rho_{\sigma,c}(L) \ = \sum_{x\in L}\rho_{\sigma,c}(x)$: the (always converging) sum of $\rho_{\sigma,c}$ over L,

$\mathcal{D}_{L,\sigma,c}$: the discrete Gaussian distribution over L with parameters σ and c,

$$\forall y\in L \quad , \quad \mathcal{D}_{L,\sigma,c}(y) = \frac{\rho_{\sigma,c}(y)}{\rho_{\sigma,c}(L)}$$

We abbreviate $\rho_{\sigma,0}$ and $\mathcal{D}_{L,\sigma,0}$ as ρ_σ and $\mathcal{D}_{L,\sigma}$. We write ρ to denote ρ_1. The distribution $\mathcal{D}_{L,\sigma,c}$ will most often be defined over the lattice $L=\Lambda_q^\perp(A)$ for a matrix $A\in\mathbb{Z}_q^{n\times m}$ or over a coset $L=t+\Lambda_q^\perp(A)$ where $t\in\mathbb{Z}^m$.

Properties. The following lemma from [31] captures standard properties of these distributions. The first two properties follow from Lemma 4.4 of [30] and Corollary 3.16 of [34] respectively (using Lemma 3.1 from [22] to bound the smoothing parameter). We state in property (2) a stronger version of Regev's Corollary 3.16 found in [2]. The last two properties are algorithms from [22].

Lemma 7. *Let $q\geq 2$ and let A be a matrix in $\mathbb{Z}_q^{n\times m}$ 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 $c\in\mathbb{R}^m$ and $u\in\mathbb{Z}_q^n$:*

1. $\Pr\left[\ x\sim\mathcal{D}_{\Lambda_q^u(A),\sigma}\ :\ \|x\|>\sqrt{m}\,\sigma\ \right]\ \leq\ \mathrm{negl}(n)$.
2. *A set of $O(m\log m)$ samples from $\mathcal{D}_{\Lambda_q^\perp(A),\sigma}$ contains a full rank set in $\mathbb{Z}^m$, except with negligible probability.*
3. *There is a PPT algorithm $\mathsf{SampleGaussian}(A,T_A,\sigma,c)$ that returns $x\in\Lambda_q^\perp(A)$ drawn from a distribution statistically close to $\mathcal{D}_{\Lambda,\sigma,c}$.*

4. *There is a PPT algorithm* $\mathsf{SamplePre}(A, T_A, u, \sigma)$ *that returns* $x \in \Lambda_q^u(A)$ *sampled from a distribution statistically close to* $\mathcal{D}_{\Lambda_q^u(A),\sigma}$*, whenever* $\Lambda_q^u(A)$*is not empty.*

Recall that when $\Lambda_q^u(A)$ is not empty then $\Lambda_q^u(A) = t + \Lambda_q^\perp(A)$ for some $t \in \Lambda_q^u(A)$. Algorithm $\mathsf{SamplePre}(A, T_A, u, \sigma)$ works by calling $\mathsf{SampleGaussian}(A, T_A, \sigma, t)$ and subtracts t from the result.

2.6 The LWE Hardness Assumption

Security of all our constructions reduces to the LWE (learning with errors) problem, a classic hard problem on lattices defined by Regev [34].

Definition 8. Consider a prime q, a positive integer n, and a distribution χ over $\mathbb{Z}_q$, all public. An $(\mathbb{Z}_q, n, \chi)$-LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being, either, a noisy pseudo-random sampler $\mathcal{O}_s$ carrying some constant random secret key $s \in \mathbb{Z}_q^n$, or, a truly random sampler $\mathcal{O}_\$$, whose behaviors are respectively as follows:

$\mathcal{O}_s$**:** outputs samples of the form $(u_i, v_i) = \big(u_i,\ u_i^T\, s + x_i\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where, $s \in \mathbb{Z}_q^n$ is a uniformly distributed persistent value invariant across invocations, $x_i \in \mathbb{Z}_q$ is a fresh sample from χ, and u_i is uniform in $\mathbb{Z}_q^n$.

$\mathcal{O}_\$$**:** outputs truly uniform random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

The $(\mathbb{Z}_q, n, \chi)$-LWE problem allows repeated queries to the challenge oracle $\mathcal{O}$. We say that an algorithm $\mathcal{A}$ decides the $(\mathbb{Z}_q, n, \chi)$-LWE problem if

$$\mathsf{LWE\text{-}adv}[\mathcal{A}] := \big|\Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_\$} = 1]\big|$$

is non-negligible for a random $s \in \mathbb{Z}_q^n$.

Regev [34] shows that for certain noise distributions χ, denoted $\overline{\Psi}_\alpha$, the LWE problem is as hard as the worst-case SIVP and GapSVP under a quantum reduction (see also [32]).

Definition 9. Consider a real parameter $\alpha = \alpha(n) \in (0, 1)$ and a prime q. Denote by $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ the group of reals $[0, 1)$ with addition modulo 1. Denote by Ψ_α the distribution over $\mathbb{T}$ 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 $x \in \mathbb{R}$. We denote by $\overline{\Psi}_\alpha$ the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q\, X \rceil \mod q$ where the random variable $X \in \mathbb{T}$ has distribution Ψ_α.

Theorem 10 ([34])**.** *If there exists an efficient, possibly quantum, algorithm for deciding the* $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$*-LWE problem for* $q > 2\sqrt{n}/\alpha$ *then there exists an efficient quantum algorithm for approximating the SIVP and GapSVP problems, to within* $\tilde{O}(n/\alpha)$ *factors in the* ℓ_2 *norm, in the worst case.*

If we assume the hardness of approximating the SIVP or GapSVP problems in lattices of dimension n to within approximation factors that are polynomial in n, then it follows from Lemma 9 that deciding the LWE problem is hard when n/α is polynomial in n.

The following lemma about the distribution $\overline{\Psi}_\alpha$ will be needed to show that decryption works correctly. The proof is implicit in [22, Lemma 8.2].

Lemma 11. *Let e be some vector in $\mathbb{Z}^m$ and let $y \xleftarrow{R} \overline{\Psi}_\alpha^m$. Then the quanity $|e^\top y|$ treated as an integer in $[0, q-1]$ satisfies*

$$|e^\top y| \leq \|e\|\, q\alpha\, \omega(\sqrt{\log m}\,) + \|e\|\sqrt{m}/2$$

with all but negligible probability in m.

Proof. By definition of $\overline{\Psi}_\alpha$, we have $y_i = \lfloor q \cdot w_i \rceil \mod q$, where the w_i are independent Gaussian variables of mean 0 and variance $\alpha^2/(2\pi)$. Let $w := (w_1, \ldots, w_m)$ then since the rounding error per component is at most $1/2$, we have $\|y - qw\| \leq \frac{\sqrt{m}}{2}$. Moreover, the random variable $|e^\top w|$ is Gaussian with mean 0 and variance $\|e\|^2\alpha^2/(2\pi)$, and therefore by a standard Gaussian tail bound, $Pr\big(|e^\top w| > \|e\|\alpha\, \omega(\sqrt{\log m})\big) \leq \text{negl}(m)$. By triangle inequality and the Cauchy-Schwarz inequality, w.h.p

$$|e^\top y| \leq |e^\top(y - qw)| + |e^\top(qw)| \leq \|e\|\sqrt{m}/2 + \|e\|\, q\alpha\, \omega(\sqrt{\log m}\,)$$

which proves the lemma. □

As a special case, Lemma 10 shows that if $x \xleftarrow{R} \overline{\Psi}_\alpha$ is treated as an integer in $[0, q-1]$ then $|x| < q\alpha\, \omega(\sqrt{\log m}) + 1/2$ with all but negligible probability in m.

3 Randomness Extraction

We will need the following lemma which follows directly from a generalization of the leftover hash lemma due to Dodis et al. [21].

Lemma 12. *Suppose that $m > (n+1)\log_2 q + \omega(\log n)$ and that $q > 2$ is prime. Let R be an $m \times k$ matrix chosen uniformly in $\{1, -1\}^{m \times k} \mod q$ where $k = k(n)$ is polynomial in n. Let A and B be matrices chosen uniformly in $\mathbb{Z}_q^{n \times m}$ and $\mathbb{Z}_q^{n \times k}$ respectively. Then, for all vectors w in $\mathbb{Z}_q^m$, the distribution $(A,\ AR,\ R^\top w)$ is statistically close to the distribution $(A,\ B,\ R^\top w)$.*

To prove the lemma recall that for a prime q the family of hash functions $h_A : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ for $A \in \mathbb{Z}_q^{n \times m}$ defined by $h_A(x) = A\,x$ is universal. Therefore, when the k columns of R are sampled independently and have sufficient entropy, the leftover hash lemma (e.g. as stated in [36, Theorem 8.38]) shows that the distributions $(A,\ AR)$ and $(A,\ B)$ are statistically close. A generalization by Dodis et al. [21] (Lemma 2.2b and 2.4) shows that the same holds even if some small amount of information about R is leaked. In our case $R^\top w$ is leaked which is precisely the settings of Dodis et al. The details follow (the reader can safely skip the remainder of this section on a first reading).

Let T be a random variable taking values in some set X. Recall that the guessing probability of T is defined as $\gamma(T) = \max_t \Pr[T = t]$. Also, recall that a family of hash functions $\mathcal{H} = \{h : X \to Y\}_{h \in \mathcal{H}}$ is universal if for all $x_1 \neq x_2 \in X$ we have that $\Pr_{h \in \mathcal{H}}[h(x_1) = h(x_2)] = 1/|Y|$. Let U_Y denote a uniform independent random variable in Y. The "classic" left-over-hash-lemma states that when h is uniform in $\mathcal{H}$ and independent of T, the distribution $(h, h(T))$ is statically close to (h, U_Y), assuming $\gamma(T)$ is sufficiently small [25] (see also [36, Theorem 8.37]). The following lemma shows that about the same holds, even if a few bits of T are "leaked."

Lemma 13 (Generalized left-over hash lemma). *Let $\mathcal{H} = \{h : X \to Y\}_{h \in \mathcal{H}}$ be a universal hash family. Let $f : X \to Z$ be some function. Then for any random variable T taking values in X we have*

$$\Delta\left(\ \big(h, h(T), f(T)\big) \ , \ \big(h, U_Y, f(T)\big) \ \right) \le \frac{1}{2} \cdot \sqrt{\gamma(T)\, |Y|\, |Z|} \tag{1}$$

More generally, let $T_1, \ldots, T_k$ be independent random variables taking values in X. Let $\gamma := \max_{i=1,\ldots,k} \gamma(T_i)$. Then

$$\Delta\Big(\ \big(h, \ h(T_1), f(T_1), \ \ldots, \ h(T_k), f(T_k)\,\big) \ , \ \big(h, \ U_Y^{(1)}, f(T_1), \ \ldots, \ U_Y^{(k)}, f(T_k)\,\big) \ \Big) \le \frac{k}{2} \cdot \sqrt{\gamma\, |Y|\, |Z|} \tag{2}$$

Proof. The proof of (1) follows directly from Lemma 2.2b and Lemma 2.4 in [21]. Equation (2) follows from (1) by a hybrid argument identical to the one given in the proof of Theorem 8.38 in [36]. □

Proof of Lemma 11. Define the family of hash functions $\mathcal{H} = \{h_A : \mathbb{Z}_q^m \to \mathbb{Z}_q^n\}$where $h_A(r) = Ar$ and $A \in \mathbb{Z}_q^{n\times m}$. Since q is prime we have that for all $r_1 \neq r_2 \in \mathbb{Z}_q^m$ there are exactly $q^{n(m-1)}$ matrices $A \in \mathbb{Z}_q^{n\times m}$ such that $Ar_1 = Ar_2$. Hence, $\mathcal{H}$ is universal. For a vector $w \in \mathbb{Z}_q^m$, let $f : \mathbb{Z}_q^m \to \mathbb{Z}_q$ be the function $f(r) = r^\top \cdot w$. Observe that for a matrix $R \in \mathbb{Z}_q^{m\times k}$ whose columns are $r_1, \ldots, r_k \in \mathbb{Z}_q^m$ we have that $R^\top w = (f(r_1), \ldots, f(r_k)) \in \mathbb{Z}_q^k$. Similarly, the columns of the matrix $A \cdot R$ are the k columns vectors $h_A(r_1), \ldots, h_A(r_k)$.

Now, using the notation of Lemma 11, observe that the k columns of R are independent vectors uniform in $\{1, -1\}^m$. Therefore, letting $T_1, \ldots, T_m$ be the m columns of R and setting $X = \mathbb{Z}_q^m$, $Y = \mathbb{Z}_q^n$ and $Z = \mathbb{Z}_q$, we obtain from (2) that

$$\Delta\left(\ \big(A, \ AR, \ R^\top w\big) \ , \ \big(A, \ B, \ R^\top w\big) \ \right) \le \frac{k}{2} \cdot \sqrt{2^{-m} \cdot q^n \cdot q} = \frac{k}{2} \cdot \sqrt{2^{-m+(n+1)\log q}} \ . \tag{3}$$

When $m > (n+1)\log_2 q + \omega(\log n)$ and k is polynomial in n, the quantity on the right is at most $k\sqrt{2^{-\omega(\log n)}}/2$ which is $\mathrm{negl}(n)$, as required. □

3.1 The Norm of a Random Matrix

Recall that the norm of a matrix $R \in \mathbb{R}^{k\times m}$ is defined as $\|R\| := \sup_{\|u\|=1} \|Ru\|$. We will need the following lemma from Litvak et al. [5] to bound the norm of a random matrix in $\{-1, 1\}^{m\times m}$. A similar lemma appears in [6, Lemma 2.2].

Lemma 14. *Let R be an $m \times m$ matrix chosen at random from $\{-1, 1\}^{m\times m}$. Then for all vectors $u \in \mathbb{R}^m$ we have*

$$\Pr\left[\ \|R\| > C\sqrt{m}\right] < e^{-m}$$

for some universal constant C (taking $C = 16$ is sufficient).

Proof. The proof follows from Litvak et al. [5] Fact 2.4 with $a_2 = 1$. Their proof shows that taking $C = 16$ is sufficient. □

4 Sampling Algorithms

Let A and B be matrices in $\mathbb{Z}_q^{n\times m}$ and let R be a matrix in $\{-1,1\}^{m\times m}$. Our construction makes use of matrices of the form $F = (A \mid AR + B) \in \mathbb{Z}_q^{n\times 2m}$ and we will need to sample short vectors in $\Lambda_q^u(F)$ for some u in $\mathbb{Z}_q^n$. We show that this can be done using either a trapdoor for $\Lambda_q^\perp(A)$ or a trapdoor $\Lambda_q^\perp(B)$. More precisely, we define two algorithms:

1. $\mathsf{SampleLeft}$ takes a basis for $\Lambda_q^\perp(A)$ (the left side of F) and outputs a short vector $e \in \Lambda_q^u(F)$.
2. $\mathsf{SampleRight}$ takes a basis for $\Lambda_q^\perp(B)$ (the right side of F) and outputs a short vector $e \in \Lambda_q^u(F)$.

We will show that, with appropriate parameters, the distributions on e produced by these two algorithms are statistically indistinguishable.

4.1 Algorithm SampleLeft

Algorithm $\mathsf{SampleLeft}(A, M_1, T_A, u, \sigma)$:
Inputs:

$$\begin{array}{l} \text{a rank } n \text{ matrix } A \text{ in } \mathbb{Z}_q^{n\times m} \text{ and a matrix } M_1 \text{ in } \mathbb{Z}_q^{n\times m_1}, \\ \text{a “short” basis } T_A \text{ of } \Lambda_q^\perp(A) \text{ and a vector } u \in \mathbb{Z}_q^n, \\ \text{a gaussian parameter } \sigma > \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log(m+m_1)}). \end{array} \tag{4}$$

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

The algorithm appears in Theorem 3.4 in [17] and also in the signing algorithm in [31]. For completeness, we briefly review the algorithm.

1. sample a random vector $e_2 \in \mathbb{Z}^{m_1}$ distributed statistically close to $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma}$,
2. run $e_1 \xleftarrow{R} \mathsf{SamplePre}(A, T_A, y, \sigma)$ where $y = u - (M_1 \cdot e_2) \in \mathbb{Z}_q^n$,
   note that $\Lambda_q^y(A)$ is not empty since A is rank n,
3. output $e \leftarrow (e_1, e_2) \in \mathbb{Z}^{m+m_1}$

Clearly $(A \,|\, M_1) \cdot e = u \bmod q$ and hence $e \in \Lambda_q^u(F_1)$. Theorem 3.4 in [17] shows that the vector e is sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$.

Peikert's basis extension method [31] gives an alternate way to view this. Theorem **??** shows how to construct a basis T_{F_1} of $\Lambda_q^\perp(F_1)$ from a basis T_A of $\Lambda_q^\perp(A)$ such that $\|\widetilde{T_{F_1}}\| = \|\widetilde{T_A}\|$. Then calling $\mathsf{SamplePre}(F_1, T_{F_1}, u, \sigma)$ generates a vector e sampled from a distribution close to $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$. We summarize this in the following theorem.

Theorem 15. *Let $q > 2$, $m > n$ and $\sigma > \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log(m+m_1)})$. Then $\mathsf{SampleLeft}(A, M_1, T_A, u, \sigma)$ taking inputs as in (4), outputs a vector $e \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$ where $F_1 := (A \mid M_1)$.*

4.2 Algorithm SampleRight

Algorithm $\mathsf{SampleRight}(A, B, R, T_B, u, \sigma)$.
Inputs:

$$\begin{aligned}
&\text{matrices } A \text{ in } \mathbb{Z}_q^{n\times k} \text{ and } B \text{ in } \mathbb{Z}_q^{n\times m} \text{ where } B \text{ is rank } n,\\
&\text{a matrix } R \in \mathbb{Z}^{k\times m}, \quad \text{let } s_R := \|R\| = \sup_{\|x\|=1} \|Rx\|,\\
&\text{a basis } T_B \text{ of } \Lambda_q^\perp(B) \text{ and a vector } u \in \mathbb{Z}_q^n,\\
&\text{a parameter } \sigma > \|\widetilde{T_B}\| \cdot s_R\, \omega(\sqrt{\log m}).
\end{aligned} \tag{5}$$

Often the matrix R given to the algorithm as input will be a a random matrix in $\{1, -1\}^{m\times m}$. Then Lemma 13 shows that $s_R < O(\sqrt{m})$ w.h.p.

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

The algorithm uses the basis growth method of Peikert [31, Sec. 3.3] and works in three steps:

1. First, it constructs a set T_{F_2} of $(m + k)$ linearly independent vectors in $\Lambda_q^\perp(F_2)$ such that
$$\|\widetilde{T_{F_2}}\| < \|\widetilde{T_B}\|\, (s_R + 1) \qquad \big(< \sigma/\omega(\sqrt{\log m})\ \big)$$
2. Next, if needed it uses Lemma 3 to convert T_{F_2} into a basis T'_{F_2} of $\Lambda_q^\perp(F_2)$ with the same Gram-Schmidt norm as T_{F_2}.
3. Finally, it invokes $\mathsf{SamplePre}(F_2, T'_{F_2}, u, \sigma)$ to generate a vector $e \in \Lambda_q^u(F_2)$.
Since $\sigma > \|\widetilde{T_{F_2}}\|\, \omega(\sqrt{\log m})$ this e is distributed close to $\mathcal{D}_{\Lambda_q^u(F_2),\sigma}$, as required.

The shifted lattice $\Lambda_q^u(F_2)$ used in step 3 is not empty. To see why, choose an arbitrary $x \in \mathbb{Z}^m$ satisfying $Bx = u \bmod q$ and observe that $(-Rx \,|\, x) \in \mathbb{Z}^{m+k}$ is in $\Lambda_q^u(F_2)$. This x must exist since B is rank n. Thus, $\Lambda_q^u(F_2)$ is not empty and therefore e is distributed close to $\mathcal{D}_{\Lambda_q^u(F_2),\sigma}$ as stated.

Step 1 is the only step that needs explaining. Let $T_B = \{b_1, \ldots, b_m\} \in \mathbb{Z}^{m\times m}$ be the given basis of $\Lambda_q^\perp(B)$. We construct $(m + k)$ linearly independent vectors $t_1, \ldots, t_{m+k}$ in $\Lambda_q^\perp(F_2)$ as follows:

1. for $i = 1, \ldots, m$ set $t_i := (-Rb_i \mid b_i) \in \mathbb{Z}^{m+k}$ and view it as a column vector; then clearly $F_2 \cdot t_i = B\, b_i = 0 \bmod q$ and therefore t_i is in $\Lambda_q^\perp(F_2)$.
2. for $i = 1, \ldots, k$ let w_i be the i-th column of the identity matrix I_k. Let u_i be an arbitrary vector in $\mathbb{Z}^m$ satisfying $Aw_i + Bu_i = 0 \bmod q$. This u_i exists since B is rank n. Set t_{i+m} to be
$$t_{i+m} := \begin{bmatrix} w_i - Ru_i \\ u_i \end{bmatrix} \in \mathbb{Z}^{m+k}$$
Then $F_2 \cdot t_{i+m} = Aw_i + Bu_i = 0 \bmod q$ and hence, $t_{i+m} \in \Lambda_q^\perp(F_2)$.

Lemma 16. *The vectors $T_{F_2} := \{t_1, \ldots, t_{m+k}\}$ are linearly independent in $\mathbb{Z}^{m+k}$ and satisfy $\|\widetilde{T_{F_2}}\| \le \|\widetilde{T_B}\| \cdot (s_R + 1)$.*

Proof. Observe that the first m vectors are linearly independent and span the linear space V of vectors of the form $(-Rx \mid x)$ where $x \in \mathbb{Z}_q^m$. For all $i > m$, the vector t_i is the sum of the unit vector $(w_i \mid 0^m)$ plus a vector in V. It follows that T_{F_2} is a linearly independent set. This also means that for $i > m$ the i-th Gram-Schmidt vector of T_{F_2} cannot be longer than $(w_i \mid 0^m)$ and therefore has norm at most 1. Hence, to bound $\|\widetilde{T_{F_2}}\|$ it suffices to bound the Gram-Schmidt norm of the first m vectors $\{t_1, \dots, t_m\}$.

Let $W \in \mathbb{Z}^{(m+k)\times m}$ be the matrix $(-R^\top \mid I_m)^\top$ and observe that $t_i = Wb_i$ for $i = 1, \dots, m$. Since $\|R\| \le s_R$ we obtain that for all $x \in \mathbb{R}^m$

$$\|W\,x\| \le \|R\,x\| + \|x\| \le \|x\|\,s_R + \|x\| \le \|x\|\,(s_R + 1)$$

Now, since $t_i = Wb_i$ for $i = 1, \dots, m$, applying Lemma 5 to the matrix W gives a bound on the Gram-Schmidt norm of $\{t_1, \dots, t_m\}$ (and hence also on $\|\widetilde{T_{F_2}}\|$):

$$\|\widetilde{T_{F_2}}\| \le \max_{1\le i\le m} \|W\,\tilde{b}_i\| \le \max_{1\le i\le m} \|\tilde{b}_i\| \cdot (s_R + 1) \le \|\widetilde{T_B}\| \cdot (s_R + 1)$$

as required. □

Thus, we built $m + k$ linearly independent vectors in $\Lambda_q^\perp(F_2)$ that have a short Gram-Schmidt norm as required for Step 1. This completes the description of algorithm SampleRight. We summarize this in the following theorem.

Theorem 17. *Let $q > 2, m > n$ and $\sigma > \|\widetilde{T_B}\| \cdot s_R\,\omega(\sqrt{\log m})$. Then SampleRight$(A, B, R, T_B, u, \sigma)$ taking inputs as in (5) outputs a vector $e \in \mathbb{Z}^{m+k}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^u(F_2),\sigma}$ where $F_2 := (A \mid AR + B)$.*

5 Encoding Identities as Matrices

Our construction uses an encoding function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ to map identities in $\mathbb{Z}_q^n$ to matrices in $\mathbb{Z}_q^{n\times n}$. Our proof of security requires that the map H satisfy a strong notion of injectivity, namely that, for any two distinct inputs id_1 and id_2, the difference between the outputs $H(\mathsf{id}_1)$ and $H(\mathsf{id}_2)$ is never singular, i.e., $\det(H(\mathsf{id}_1) - H(\mathsf{id}_2)) \neq 0$.

Definition 18. Let q be a prime and n a positive integer. We say that a function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is an **encoding with full-rank differences** (FRD) if:

1. for all distinct $u, v \in \mathbb{Z}_q^n$, the matrix $H(u) - H(v) \in \mathbb{Z}_q^{n\times n}$ is full rank; and
2. H is computable in polynomial time (in $n \log q$).

Clearly the function H must be injective since otherwise, if $u \neq v$ satisfies $H(u) = H(v)$, then $H(u) - H(v)$ is not full-rank and hence H cannot be FRD.

The function H in Definition 17 has domain of size q^n which is the largest possible for a function satisfying condition 1 of Definition 17. Indeed, if H had domain larger than q^n then its image is also larger than q^n. But then, by pigeonhole, there are two distinct inputs u, v such that the matrices $H(u)$ and $H(v)$ have the same first row and therefore $H(u) - H(v)$ is not full rank. It follows that our definition of FRD, which has domain of size of q^n, is the largest possible.

An Explicit FRD Construction. We construct an injective FRD encoding for the exponential-size domain $\mathsf{id} \in \mathbb{Z}_q^n$. A similar construction is described in [20]. Our strategy is to construct an additive subgroup $\mathbb{G}$ of $\mathbb{Z}_q^{n\times n}$ of size q^n such that all non-zero matrices in $\mathbb{G}$ are full-rank. Since for all distinct $A, B \in \mathbb{G}$ the difference $A - B$ is also in $\mathbb{G}$, it follows that $A - B$ is full-rank.

While our primary interest is the finite field $\mathbb{Z}_q$ we describe the construction for an arbitrary field $\mathbb{F}$. For a polynomial $g \in \mathbb{F}[X]$ of degree less than n define $\text{coeffs}(g) \in \mathbb{F}^n$ to be the n-vector of coefficients of g (written as a row-vector). If g is of degree less than $n-1$ we pad the coefficients vector with zeroes on the right to make it an n-vector. For example, for $n = 6$ we have $\text{coeffs}(x^3 + 2x + 3) = (3, 2, 0, 1, 0, 0) \in \mathbb{F}^6$. Let f be some polynomial of degree n in $\mathbb{F}[X]$ that is irreducible. Recall that for a polynomial $g \in \mathbb{F}[X]$ the polynomial $g \bmod f$ has degree less than n and therefore $\text{coeffs}(g \bmod f)$ is a vector in $\mathbb{F}^n$.

Now, for an input $u = (u_0, u_1, \ldots, u_{n-1}) \in \mathbb{F}^n$ define the polynomial $g_u(X) = \sum_{i=0}^{n-1} u_i x^i \in \mathbb{F}[X]$. Define $H(u)$ as

$$H(u) := \begin{pmatrix} \text{coeffs}(\ g_u\ ) \\ \text{coeffs}(\ X \cdot g_u \bmod f\ ) \\ \text{coeffs}(\ X^2 \cdot g_u \bmod f\ ) \\ \vdots \\ \text{coeffs}(\ X^{n-1} \cdot g_u \bmod f\ ) \end{pmatrix} \in \mathbb{F}^{n\times n} \tag{6}$$

This completes the construction. Since for all primes q and integers $n > 1$ there are (many) irreducible polynomials in $\mathbb{Z}_q[X]$ of degree n, the construction can accommodate any pair of q and n.

The following theorem proves that the function H in (6) is an FRD. The proof, given in [20], is based on the observation that the matrix $H(u)^\top$ corresponds to multiplication by a constant in the number field $K = \mathbb{F}[X]/(f)$ and is therefore invertible when the matrix is non-zero. We note that similar matrix encodings of ring multiplication were previously used in [33, 28].

Theorem 19 ([20])**.** *Let $\mathbb{F}$ be a field and f a polynomial in $\mathbb{F}[X]$. If f is irreducible in $\mathbb{F}[X]$ then the function H defined in (6) is an encoding with full-rank differences (or FRD encoding).*

An example. Let $n = 4$ and $f(X) = x^4 + x - 1$. The function H works as follows:

$$H\big(u = (u_0, u_1, u_2, u_3)\big) \ := \begin{pmatrix} u_0 & u_1 & u_2 & u_3 \\ u_3 & u_0 - u_3 & u_1 & u_2 \\ u_2 & u_3 - u_2 & u_0 - u_3 & u_1 \\ u_1 & u_2 - u_1 & u_3 - u_2 & u_0 - u_3 \end{pmatrix}$$

Theorem 18 shows that the map H is FRD for all primes q where $x^4 + x - 1$ is irreducible in $\mathbb{Z}_q[X]$ (e.g. $q = 19, 31, 43, 47$).

6 The Main Construction: an Efficient IBE

The system uses parameters q, n, m, σ, α specified in Section 6.3. Throughout the section, the function H refers to the FRD map $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ defined in Section 5. We assume identities are elements in $\mathbb{Z}_q^n$. The set of identities can be expanded to $\{0,1\}^*$ by hashing identities into $\mathbb{Z}_q^n$ using a collision resistant hash.

6.1 Intuition

The public parameters in our system consist of three random $n \times m$ matrices over $\mathbb{Z}_q$ denoted by A_0, A_1 and B as well as a vector $u \in \mathbb{Z}_q^n$. The master secret is a trapdoor T_{A_0} (i.e. a basis with a low Gram-Schmidt norm) for the lattice $\Lambda_q^\perp(A_0)$.

The secret key for an identity id is a short vector $e \in \mathbb{Z}^{2m}$ satisfying $F_{\mathsf{id}} \cdot e = u$ in $\mathbb{Z}_q$ where

$$F_{\mathsf{id}} := (A_0 \mid A_1 + H(\mathsf{id})\, B) \quad \in \mathbb{Z}_q^{n \times 2m}$$

The vector e is generated using algorithm $\mathsf{SampleLeft}$ (Theorem 14) and the trapdoor T_{A_0}.

In a selective IBE security game the attacker announces an identity id^* that it plans to attack. We need a simulator that can respond to private key queries for $\mathsf{id} \neq \mathsf{id}^*$, but knows nothing about the private key for id^*. We do so by choosing the public parameters A_0 and B at random as before, but choosing A_1 as

$$A_1 := A_0\, R - H(\mathsf{id}^*)\, B$$

where R is a random matrix in $\{1, -1\}^{m \times m}$. We show that $A_0\, R$ is uniform and independent in $\mathbb{Z}_q^{n \times m}$ so that A_1 is distributed as required. We provide the simulator with a trapdoor T_B for $\Lambda_q^\perp(B)$, but no trapdoor for $\Lambda_q^\perp(A_0)$.

Now, to respond to a private key query for an identity id, the simulator must produce a short vector e satisfying $F_{\mathsf{id}} \cdot e = u$ in $\mathbb{Z}_q$ where

$$F_{\mathsf{id}} := \big(A_0 \mid A_0 \cdot R + B'\big) \in \mathbb{Z}_q^{n \times 2m} \quad \text{and} \quad B' := \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big) \cdot B\ .$$

When $\mathsf{id} \neq \mathsf{id}^*$ we know that $H(\mathsf{id}) - H(\mathsf{id}^*)$ is full rank by construction and therefore T_B is also a trapdoor for the lattice $\Lambda_q^\perp(B')$. The simulator can now generate e using algorithm $\mathsf{SampleRight}$ and the basis T_B.

When $\mathsf{id} = \mathsf{id}^*$ the matrix F_{id} no longer depends on B and the simulator's trapdoor disappears. Consequently, the simulator can generate private keys for all identities other than id^*. As we will see, for id^* the simulator can produce a challenge ciphertext that helps it solve the given LWE challenge.

6.2 The Basic IBE Construction

Setup(λ)**:** On input a security parameter λ, set the parameters q, n, m, σ, α as specified in Section 6.3 below. Next do:

1. Use algorithm $\mathsf{TrapGen}(q, n)$ to select a uniformly random $n \times m$-matrix $A_0 \in \mathbb{Z}_q^{n \times m}$ with a basis T_{A_0} for $\Lambda_q^\perp(A_0)$ such that $\|\widetilde{T_{A_0}}\| \leq O(\sqrt{n \log q})$
2. Select two uniformly random $n \times m$ matrices A_1 and B in $\mathbb{Z}_q^{n \times m}$.
3. Select a uniformly random n-vector $u \xleftarrow{R} \mathbb{Z}_q^n$.
4. Output the public parameters and master key,

$$\mathsf{PP} = \Big(\ A_0, A_1, B, u\ \Big) \qquad ; \qquad \mathsf{MK} = \Big(\ T_{A_0}\ \Big) \quad \in \mathbb{Z}^{m \times m}$$

Extract($\mathsf{PP}, \mathsf{MK}, \mathsf{id}$)**:** On input public parameters PP, a master key MK, and an identity $\mathsf{id} \in \mathbb{Z}_q^n$, do:

1. Sample $e \in \mathbb{Z}^{2m}$ as $e \leftarrow \mathsf{SampleLeft}(A_0, \quad A_1 + H(\mathsf{id})\, B, \quad T_{A_0}, \quad u, \quad \sigma)$ where H is an FRD map as defined in Section 5.
   Note that A_0 is rank n w.h.p as explained in Section 6.3.
2. Output $\mathsf{SK}_{\mathsf{id}} := e \ \in \mathbb{Z}^{2m}$

Let $F_{\mathsf{id}} := \big(A_0 \mid A_1 + H(\mathsf{id})\, B\big)$, then $F_{\mathsf{id}} \cdot e = u$ in $\mathbb{Z}_q$ and e is distributed as $D_{\Lambda_q^u(F_{\mathsf{id}}),\sigma}$ by Theorem 14.

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

1. Set $F_{\mathsf{id}} \leftarrow \big(A_0 \mid A_1 + H(\mathsf{id}) \cdot B\big) \in \mathbb{Z}_q^{n \times 2m}$
2. Choose a uniformly random $s \xleftarrow{R} \mathbb{Z}_q^n$
3. Choose a uniformly random $m \times m$ matrix $R \xleftarrow{R} \{-1,1\}^{m \times m}$
4. Choose noise vectors $x \xleftarrow{\bar{\Psi}_\alpha} \mathbb{Z}_q$ and $y \xleftarrow{\bar{\Psi}_\alpha^m} \mathbb{Z}_q^m$, and set $z \leftarrow R^\top y \in \mathbb{Z}_q^m$ (the distribution $\bar{\Psi}_\alpha$ is as in Definition 8),
5. Set $c_0 \leftarrow u^\top s + x + b \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$ and $c_1 \leftarrow F_{\mathsf{id}}^\top s + \begin{bmatrix} y \\ z \end{bmatrix} \in \mathbb{Z}_q^{2m}$
6. Output the ciphertext $\mathsf{CT} := (c_0, c_1) \ \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$.

$\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}_{\mathsf{id}}, \mathsf{CT})$: On input public parameters PP, a private key $\mathsf{SK}_{\mathsf{id}} := e_{\mathsf{id}}$, and a ciphertext $\mathsf{CT} = (c_0, c_1)$, do:

1. Compute $w \leftarrow c_0 - e_{\mathsf{id}}^\top c_1 \in \mathbb{Z}_q$.
2. Compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in $\mathbb{Z}$. If they are close, i.e., if $\left| w - \lfloor \frac{q}{2} \rfloor \right| < \lfloor \frac{q}{4} \rfloor$ in $\mathbb{Z}$, output 1, otherwise output 0.

The matrix R. The matrix R used in encryption plays an important role in the security proof. Note that the matrix is only used as a tool to sample the noise vector (y, z) from a specific distribution needed in the simulation.

6.3 Parameters and Correctness

When the cryptosystem is operated as specified, we have,

$$w \ = c_0 - e_{\mathsf{id}}^\top c_1 = b \lfloor \frac{q}{2} \rfloor + \underbrace{x - e_{\mathsf{id}}^\top \begin{bmatrix} y \\ z \end{bmatrix}}_{\text{error term}}$$

Lemma 20. *The norm of the error term is bounded by $[q\sigma m\alpha\, \omega(\sqrt{\log m}) + O(\sigma m^{3/2})]$ w.h.p.*

Proof. Letting $e_{\mathsf{id}} = (e_1|e_2)$ with $e_1, e_2 \in \mathbb{Z}^m$ the error term is

$$x - e_1^\top y - e_2^\top z = x - e_1^\top y - e_2^\top R^\top y = x - (e_1 - Re_2)^\top y$$

By Lemma 6 (part 1) we have $\|e_{\mathsf{id}}\| \leq \sigma\sqrt{2m}$ w.h.p.
Hence, by Lemma 13, $\|e_1 - Re_2\| \leq \|e_1\| + \|Re_2\| \leq O(\sigma m)$.
Then, by Lemma 10 the error term is bounded by

$$\left| x - e_{\mathsf{id}}^\top \begin{bmatrix} y \\ z \end{bmatrix} \right| \leq |x| + |(e_1 - Re_2)^\top y| \leq q\sigma m\alpha\, \omega(\sqrt{\log m}) + O(\sigma m^{3/2})$$

as required. □

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

- the error term is less than $q/5$ w.h.p (i.e. $\alpha < [\sigma m \omega(\sqrt{\log m})]^{-1}$ and $q = \Omega(\sigma m^{3/2})$),
- that TrapGen can operate (i.e. $m > 6n \log q$),
- that σ is sufficiently large for SampleLeft and SampleRight (i.e. $\sigma > \sigma_{\mathrm{TG}} \sqrt{m}\, \omega(\sqrt{\log m}\,) = m\, \omega(\sqrt{\log m}\,)$), and
- that Regev's reduction applies (i.e. $q > 2\sqrt{n}/\alpha$)

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

$$\begin{aligned} m &= 6\, n^{1+\delta} \quad , \quad q = m^{2.5} \cdot \omega(\sqrt{\log n}\,) \\ \sigma &= m \cdot \omega(\sqrt{\log n}\,) \quad , \quad \alpha = [m^2 \cdot \omega(\sqrt{\log n}\,)]^{-1} \end{aligned} \tag{7}$$

and round up m to the nearest larger integer and q to the nearest larger prime. Here we assume that δ is such that $n^\delta > \lceil \log q \rceil = O(\log n)$.

Since the matrices A_0, B are random in $\mathbb{Z}_q^{n \times m}$ and $m > n \log q$, with overwhelming probability both matrices will have rank n. Hence, calling SampleLeft in algorithm Extract succeeds w.h.p.

6.4 Security Reduction

We show that the basic IBE construction is indistinguishable from random under a selective identity attack as in Definition 1. Recall that indistinguishable from random means that the challenge ciphertext is indistinguishable from a random element in the ciphertext space. This property implies both semantic security and recipient anonymity.

Theorem 21. *The basic IBE system with parameters $(q, n, m, \sigma, \alpha)$ as in (7) is INDr–sID-CPA secure provided that the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE assumption holds.*

Proof. The proof proceeds in a sequence of games where the first game is identical to the INDr–sID-CPA game from Definition 1. In the last game in the sequence the adversary has advantage zero. We show that a PPT adversary cannot distinguish between the games which will prove that the adversary has negligible advantage in winning the original INDr–sID-CPA game. The LWE problem is used in proving that Games 2 and 3 are indistinguishable.

Game 0. This is the original INDr–sID-CPA game from Definition 1 between an attacker $\mathcal{A}$ against our scheme and an INDr–sID-CPA challenger.

Game 1. Recall that in Game 0 the challenger generates the public parameters PP by choosing three random matrices A_0, A_1, B in $\mathbb{Z}_q^{n \times m}$ such that a trapdoor T_{A_0} is known for $\Lambda_q^\perp(A_0)$. At the challenge phase the challenger generates a challenge ciphertext CT^*. We let $R^* \in \{-1,1\}^{m\times m}$ denote the random matrix generated for the creation of CT^* (in step 3 of $\mathsf{Encrypt}$).

In Game 1 we slightly change the way that the challenger generates A_1 in the public parameters. Let id^* be the identity that $\mathcal{A}$ intends to attack. The Game 1 challenger chooses R^* at the setup phase and constructs A_1 as

$$A_1 \leftarrow A_0\, R^* - H(\mathsf{id}^*)\, B \tag{8}$$

The remainder of the game is unchanged.

We show that Game 0 is statistically indistinguishable from Game 1 by Lemma 11. Observe that in Game 1 the matrix R^* is used only in the construction of A_1 and in the construction of the challenge ciphertext where $z \leftarrow (R^*)^\top y$. By Lemma 11 the distribution $(A_0,\ A_0\, R^*,\ z)$ is statistically close to the distribution $(A_0,\ A_1',\ z)$ where A_1' is a uniform $\mathbb{Z}_q^{n\times m}$ matrix. It follows that in the adversary's view, the matrix $A_0\, R^*$ is statistically close to uniform and therefore A_1 as defined in (8) is close to uniform. Hence, A_1 in Games 0 and 1 are indistinguishable.

Game 2. We now change how A_0 and B in PP are chosen. In Game 2 we generate A_0 as a random matrix in $\mathbb{Z}_q^{n\times m}$, but generate B using algorithm $\mathsf{TrapGen}$ so that B is a random matrix in $\mathbb{Z}_q^{n\times m}$ for which the challenger has a trapdoor T_B for $\Lambda_q^\perp(B)$. The construction of A_1 remains as in Game 1, namely $A_1 = A_0 \cdot R^* - H(\mathsf{id}^*) \cdot B$.

The challenger responds to private key queries using the trapdoor T_B. To respond to a private key query for $\mathsf{id} \neq \mathsf{id}^*$ the challenger needs a short $e \in \Lambda_q^u(F_{\mathsf{id}})$ where

$$F_{\mathsf{id}} := (A_0 \mid A_1 + H(\mathsf{id}) \cdot B) = \big(A_0 \mid A_0 R^* + \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big)B\big)\ .$$

By construction, $[H(\mathsf{id}) - H(\mathsf{id}^*)]$ is non-singular and therefore T_B is also a trapdoor for $\Lambda_q^\perp(B')$ where $B' := \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big)B$. Moreover, since B is rank n w.h.p, so is B'. The challenger can now respond to the private key query by running

$$e \leftarrow \mathsf{SampleRight}\big(A_0,\ \ \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big)B,\ \ R^*,\ T_B,\ \ u,\ \ \sigma\big) \in \mathbb{Z}_q^{2m}$$

and sending $\mathsf{SK}_{\mathsf{id}} := e$ to $\mathcal{A}$. Theorem 16 shows that when $\sigma > \|\widetilde{T_B}\| s_R\, \omega(\sqrt{\log m})$ the generated e is distributed close to $D_{\Lambda_q^u(F_{\mathsf{id}}),\sigma}$, as in Game 1. Recall that $\|\widetilde{T_B}\| \le \sigma_{\mathrm{TG}}$ by Theorem 4 and $s_R = \|R^*\| \le O(\sqrt{m})$ w.h.p by Lemma 13. Therefore σ used in the system, as defined in (7), is sufficiently large to satisfy the conditions of Theorem 16.

Game 2 is otherwise the same as Game 1. Since A_0, B and responses to private key queries are statistically close to those in Game 1, the adversary's advantage in Game 2 is at most negligibly different from its advantage in Game 1.

Game 3. Game 3 is identical to Game 2 except that the challenge ciphertext (c_0^*, c_1^*) is *always* chosen as a random independent element in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$. Since the challenge ciphertext is always a fresh random element in the ciphertext space, $\mathcal{A}$'s advantage in this game is zero.

It remains to show that Game 2 and Game 3 are computationally indistinguishable for a PPT adversary, which we do by giving a reduction from the LWE problem.

Reduction from LWE. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing Games 2 and 3. We use $\mathcal{A}$ to construct an LWE algorithm $\mathcal{B}$.

Recall from Definition 7 that an LWE problem instance is provided as a sampling oracle $\mathcal{O}$ which can be either truly random $\mathcal{O}_\$$ or a noisy pseudo-random $\mathcal{O}_s$ for some secret $s \in \mathbb{Z}_q^n$. The simulator $\mathcal{B}$ uses the adversary $\mathcal{A}$ to distinguish between the two, and proceeds as follows:

Instance. $\mathcal{B}$ requests from $\mathcal{O}$ and receives, for each $i = 0, \ldots, m$, a fresh pair $(u_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Targeting. $\mathcal{A}$ announces to $\mathcal{B}$ the identity id^* that it intends to attack.

Setup. $\mathcal{B}$ constructs the system's public parameters PP as follows:

1. Assemble the random matrix $A_0 \in \mathbb{Z}_q^{n \times m}$ from m of the previously given LWE samples by letting the i-th column of A_0 be the n-vector u_i for all $i = 1, \ldots, m$.
2. Assign the zeroth LWE sample (so far unused) to become the public random n-vector $u_0 \in \mathbb{Z}_q^n$.
3. The remainder of the public parameters, namely A_1 and B, are constructed as in Game 2 using id^* and R^*.

Queries. $\mathcal{B}$ answers each private-key extraction query as in Game 2.

Challenge. $\mathcal{B}$ prepares, when prompted by $\mathcal{A}$ with a message bit $b^* \in \{0, 1\}$, a challenge ciphertext for the target identity id^*, as follows:

1. Let $v_0, \ldots, v_m$ be entries from the LWE instance. Set $v^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m$.
2. Blind the message bit by letting $c_0^* = v_0 + b^* \lfloor \frac{q}{2} \rceil \in \mathbb{Z}_q$.
3. Set $c_1^* = \begin{bmatrix} v^* \\ (R^*)^\top v^* \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
4. Choose a random bit $r \xleftarrow{R} \{0, 1\}$. If $r = 0$ send $\mathsf{CT}^* = (c_0^*, c_1^*)$ to the adversary. If $r = 1$ choose a random $(c_0, c_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$ and send (c_0, c_1) to the adversary.

We argue that when the LWE oracle is pseudorandom (i.e. $\mathcal{O} = \mathcal{O}_s$) then CT^* is distributed exactly as in Game 2. First, observe that $F_{\mathsf{id}^*} = (A_0 \mid A_0 R^*)$. Second, by definition of $\mathcal{O}_s$ we know that $v^* = A_0^\top s + y$ for some random noise vector $y \in \mathbb{Z}_q^m$ distributed as $\bar{\Psi}_\alpha^m$. Therefore, c_1^* defined in step (3) above satisfies

$$c_1^* = \begin{bmatrix} A_0^\top s + y \\ (R^*)^\top A_0^\top s + (R^*)^\top y \end{bmatrix} = \begin{bmatrix} A_0^\top s + y \\ (A_0 R^*)^\top s + (R^*)^\top y \end{bmatrix} = (F_{\mathsf{id}^*})^\top s + \begin{bmatrix} y \\ (R^*)^\top y \end{bmatrix}$$

and the quantity on the right is precisely the c_1 part of a valid challenge ciphertext in Game 2. Also note that $v_0 = u_0^\top s + x$, just as the c_0 part of the challenge ciphertext in Game 2.

When $\mathcal{O} = \mathcal{O}_\$$ we have that v_0 is uniform in $\mathbb{Z}_q$ and v^* is uniform in $\mathbb{Z}_q^m$. Therefore c_1^* as defined in step (3) above is uniform and independent in $\mathbb{Z}_q^{2m}$ by the standard left over hash lemma (e.g. Theorem 8.38 of [36]) where the hash function is defined by the matrix $(A_0^\top | v^*)$. Consequently, the challenge ciphertext is always uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$, as in Game 3.

Guess. After being allowed to make additional queries, $\mathcal{A}$ guesses if it is interacting with a Game 2 or Game 3 challenger. Our simulator outputs $\mathcal{A}$'s guess as the answer to the LWE challenge it is trying to solve.

We already argued that when $\mathcal{O} = \mathcal{O}_s$ the adversary's view is as in Game 2. When $\mathcal{O} = \mathcal{O}_\$$ the adversary's view is as in Game 3. Hence, $\mathcal{B}$'s advantage in solving LWE is the same as $\mathcal{A}$'s advantage in distinguishing Games 2 and 3, as required. This completes the description of algorithm $\mathcal{B}$ and completes the proof. □

6.5 Multi-Bit Encryption

We briefly note that, as in [22], it is possible to reuse the same ephemeral encryption randomness s to encrypt multiple message bits. An N-bit message can thus be encrypted as N components c_0 plus a single component c_1, where the same ephemeral $s \in \mathbb{Z}_q^n$ is used throughout. The total ciphertext size with this technique is 1 element of $\mathbb{Z}_q$ for each bit of the message, plus a constant $2\,m$ elements of $\mathbb{Z}_q$ regardless of the message length. The ciphertext size is thus $(N + 2\,m)$ elements of $\mathbb{Z}_q$.

To do this for an N-bit message we need include N vectors $u_1, \ldots, u_N \in \mathbb{Z}_q^N$ in the public parameters PP (as opposed to only one vector u in the basic scheme). Message bit number i is encrypted as in the basic scheme, but using the vector u_i. The proof of security remains mostly unchanged, except that in the reduction to LWE the simulator queries the LWE oracle $m+N$ times instead of $m+1$ times. This enables the simulator to prepare a challenge ciphertext that encrypts N message bits using a single random vector $s \in \mathbb{Z}_q^n$. The vectors generated by the unused N LWE queries make up the vectors $u_1, \ldots, u_N$ in the public parameters.

7 Extension 1: Adaptively Security IBE

Recall that Waters [38] showed how to convert the selectively-secure IBE in [8] to an adaptively secure IBE. We show that a similar technique, also used in Boyen [14], can convert our basic IBE construction to an adaptively secure IBE. The size of the private keys and ciphertexts in the resulting system is essentially the same as in the basic scheme, though the public parameters are larger. The system is simpler and with shorter ciphertexts than the recent construction of Cash et al. [17].

7.1 Intuition

We treat an identity id as a sequence of ℓ bits $\mathsf{id} = (b_1, \ldots, b_\ell)$ in $\{1, -1\}^\ell$. Then during encryption we use the matrix

$$F_{\mathsf{id}} := \big(\ A_0 \mid B + \sum_{i=1}^{\ell} b_i\, A_i\ \big) \quad \in \mathbb{Z}_q^{n \times 2m}$$

where $A_0, A_1, \ldots, A_\ell, B$ are random matrices in the public parameters. The master key is a trapdoor T_{A_0} for A_0, as in the basic scheme.

In the security reduction, we construct each matrix A_i (excluding A_0) as

$$A_i := A_0 R_i + h_i B \quad \text{for } i = 1, \ldots, \ell$$

where all the matrices R_i are random in $\{1, -1\}^{m \times m}$ and h_i is a secret coefficient in $\mathbb{Z}_q$. Then

$$F_{\mathsf{id}} = \Big(\ A_0\ \Big|\ A_0\ \big(\sum_{i=1}^{\ell} b_i R_i\big) + \big(1 + \sum_{i=1}^{\ell} b_i h_i\big)\ B\ \Big)$$

The simulator will know a trapdoor (i.e. a short basis) T_B for B, and thus also for F_{id}, unless the coefficient of B in F_{id} cancels to zero. Such cancellation occurs for identities id for which $1+\sum_{i=1}^{\ell} b_i h_i = 0$ and these identities are unknown to the attacker. For those special identities the simulator will be unable to answer key-extraction queries, but will be able to construct a useful challenge to solve the given LWE problem instance.

The security proof will require that we choose $q = 2Q$ where Q is the number of chosen identity queries issued by the adversary. The framework of Boyen [14] enables us to reduce the modulus q to a value similar to the one in the selective IBE system of the previous section.

7.2 Full-IBE Construction

Setup(λ)**:** On input a security parameter λ, set the parameters q, n, m, σ, α as specified in Section 7.3 below. Next do:

1. Use algorithm $\mathsf{TrapGen}(q, n)$ to select a uniformly random $n \times m$-matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ with a basis T_{A_0} for $\Lambda_q^{\perp}(A_0)$ such that $\|\widetilde{T_{A_0}}\| \le O(\sqrt{n \log q})$.
2. Select $\ell + 1$ uniformly random $n \times m$ matrices $A_1, \dots, A_\ell, B \in \mathbb{Z}_q^{n\times m}$.
3. Select a uniformly random n-vector $u \in \mathbb{Z}_q^n$.
4. Output the public parameters and master key,

$$\mathsf{PP} \;=\; \big(\; A_0, A_1, \dots, A_\ell, B \;\big) \qquad , \qquad \mathsf{MK} \;=\; \big(\; T_{A_0} \;\big)$$

Extract$(\mathsf{PP}, \mathsf{MK}, \mathsf{id})$**:** On input public parameters PP, a master key MK, and an identity $\mathsf{id} = (b_1, \dots, b_\ell) \in \{1, -1\}^\ell$:

1. Let $A_{\mathsf{id}} = B + \sum_{i=1}^{\ell} b_i\, A_i \in \mathbb{Z}_q^{n\times m}$.
2. Sample $e \in \mathbb{Z}_q^{2m}$ as $e \leftarrow \mathsf{SampleLeft}(A_0,\ A_{\mathsf{id}},\ T_{A_0}, u, \sigma)$.
   Note that A_0 is rank n w.h.p as explained in Section 6.3.
3. Output $\mathsf{SK}_{\mathsf{id}} := e\ \in \mathbb{Z}^{2m}$.

Let $F_{\mathsf{id}} := \big(A_0 \mid A_{\mathsf{id}}\big)$, then $F_{\mathsf{id}} \cdot e = u$ in $\mathbb{Z}_q$ and e is distributed as $D_{\Lambda_q^u(F_{\mathsf{id}}),\sigma}$ by Theorem 14.

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

1. Let $A_{\mathsf{id}} = B + \sum_{i=1}^{\ell} b_i\, A_i \in \mathbb{Z}_q^{n\times m}$ and $F_{\mathsf{id}} := (A_0|A_{\mathsf{id}}) \in \mathbb{Z}_q^{n\times 2m}$.
2. Choose a uniformly random $s \xleftarrow{R} \mathbb{Z}_q^n$.
3. Choose ℓ uniformly random matrices $R_i \xleftarrow{R} \{-1, 1\}^{m\times m}$ for $i = 1, \dots, \ell$
   and define $R_{\mathsf{id}} = \sum_{i=1}^{\ell} b_i\, R_i \in \{-\ell, \dots, \ell\}^{m\times m}$.
4. Choose noise vectors $x \xleftarrow{\bar{\Psi}_\alpha} \mathbb{Z}_q$ and $y \xleftarrow{\bar{\Psi}_\alpha^m} \in \mathbb{Z}_q^m$, set $z \leftarrow R_{\mathsf{id}}^{\top} y \in \mathbb{Z}_q^m$,
5. Set $c_0 \leftarrow u^{\top} s + x + b\,\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$ and $c_1 \leftarrow F_{\mathsf{id}}^{\top}\, s + \begin{bmatrix} y \\ z \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
6. Output the ciphertext $\mathsf{CT} := (c_0, c_1)\ \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$.

As an optimization, note that Step 3 can be performed more efficiently by directly constructing an m-by-m-matrix R_{id} whose elements are i.i.d. from the binomial distribution assumed by the sum of ℓ independent coins in $\{-1, 1\}$.

Decrypt$(\mathsf{PP}, \mathsf{SK}_{\mathsf{id}}, \mathsf{CT})$**:** On input public parameters PP, a private key $\mathsf{SK}_{\mathsf{id}} = e_{\mathsf{id}}$, and a ciphertext $\mathsf{CT} = (c_0, c_1)$, do:

1. Compute $w \leftarrow c_0 - e_{\mathsf{id}}^{\top} c_1 \in \mathbb{Z}_q$.
2. Compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in $\mathbb{Z}$. If they are close, i.e., if $\left| w - \lfloor \frac{q}{2} \rfloor \right| < \lfloor \frac{q}{4} \rfloor$ in $\mathbb{Z}$, output 1, otherwise output 0.

7.3 Parameters and Correctness

As in Section 6.3, when the cryptosystem is operated as specified, we have,

$$w \; = c_0 - e_{\mathsf{id}}^{\top} c_1 = b \lfloor \frac{q}{2} \rfloor + \underbrace{x - e_{\mathsf{id}}^{\top} \begin{bmatrix} y \\ z \end{bmatrix}}_{\text{error term}}$$

Lemma 22. *For an ℓ bit identity* $\mathsf{id} = (b_1, \ldots, b_\ell) \in \{1, -1\}^\ell$*, the norm of the error term is bounded w.h.p by*

$$q\sigma\ell m\alpha \, \omega(\sqrt{\log m}) + O(\sigma m^{3/2})$$

Proof. The proof is identical to the proof of Lemma 19 except that the matrix R is replaced by $R_{\mathsf{id}} := R_{\ell+1} + \sum_{i=1}^{\ell} b_i R_i$. Since $\|R_{\mathsf{id}}\| \le \sum_{i=1}^{\ell} \|R_i\|$ we have by Lemma 13 that $\|R_{\mathsf{id}}| \le O(\ell\sqrt{m})$ w.h.p. This leads to the extra factor of ℓ in the error bound. □

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

- the error term is less than $q/5$ w.h.p (i.e. $\alpha < [\sigma\ell m\omega(\sqrt{\log m})]^{-1}$ and $q = \Omega(\sigma m^{3/2})$),
- that TrapGen can operate (i.e. $m > 6n \log q$),
- that σ is sufficiently large for SampleLeft and SampleRight
  (i.e. $\sigma > \sigma_{\mathrm{TG}} \ell \sqrt{m} \, \omega(\sqrt{\log m}\,) = \ell m \, \omega(\sqrt{\log m}\,)$),
- that Regev's reduction applies (i.e. $q > 2\sqrt{n}/\alpha$), and
- that our security 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:

$$\begin{aligned} m = 6\, n^{1+\delta} \qquad &, \qquad q = 2Q \\ \sigma = m\ell \cdot \omega(\sqrt{\log n}\,) \qquad &, \qquad \alpha = [\ell^2 m^2 \cdot \omega(\sqrt{\log n}\,)]^{-1} \end{aligned} \tag{9}$$

and round up m to the nearest larger integer and q to the nearest larger prime. Here we assume that δ is such that $n^\delta > \lceil \log q \rceil = O(\log n)$.

Finally, we note that the framework of Boyen [14] enables us to reduce the modulus q to a value similar to the one in the selective IBE system of the previous section.

7.4 Proving Full Security

We show that the full IBE construction is indistinguishable from random under the adaptive identity attack ($\mathsf{INDr\text{–}ID\text{-}CPA}$) defined in Section 2.1. Recall that indistinguishable from random means that the challenge ciphertext is indistinguishable from a random element in the ciphertext space. This property implies both semantic security and recipient anonymity. The reduction requires that the underlying modulus q be larger than $2Q$ where Q is the number of adaptive identity queries issued by the adversary.

Theorem 23. *The full IBE system with parameters $(q, n, m, \sigma, \alpha)$ as in (9) is $\mathsf{INDr\text{–}ID\text{-}CPA}$ secure provided that the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE assumption holds.*

In particular, suppose there exists a probabilistic algorithm $\mathcal{A}$ that wins the $\mathsf{INDr\text{–}ID\text{-}CPA}$ game with probability ϵ, making no more than $Q \leq q/2$ adaptive chosen-identity queries. Then is a probabilistic algorithm $\mathcal{B}$ that solves the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE problem in about the same time as $\mathcal{A}$ and with probability $\epsilon' \geq \epsilon/(2\,q)$.

7.4.1 Abort-resistant hash functions

The proof of Theorem 22 will use an information theoretic hashing concept we call *abort-resistant hash functions* defined as follows.

Definition 24. Let $\mathcal{H} := \{h : X \to Y\}$ be a family of hash functions from X to Y. For a set of $Q+1$ inputs $\bar{x} = (x_0, x_1, \ldots, x_Q) \in X^{Q+1}$, define the non-abort probability of $\bar{x}$ as the quantity

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

where the probability is over the random choice of H in $\mathcal{H}$.
We say that $\mathcal{H}$ is $(Q, \epsilon_{\min}, \epsilon_{\max})$ **abort-resistant** if for all $\bar{x} = (x_0, x_1, \ldots, x_Q) \in X^{Q+1}$ with $x_0 \notin \{x_1, \ldots, x_Q\}$ we have $\alpha(\bar{x}) \in [\epsilon_{\min}, \epsilon_{\max}]$. □

We will use the following abort-resistant hash family used in [38, 26, 7].
For a prime q let $(\mathbb{Z}_q^\ell)^* := \mathbb{Z}_q^\ell \setminus \{0^\ell\}$ and define the family $\mathcal{H}_{\text{Wat}} : \{\ H_h : (\mathbb{Z}_q^\ell)^* \to \mathbb{Z}_q\ \}_{h \in \mathbb{Z}_q^\ell}$ as

$$H_h(\mathsf{id}) := 1 + \sum_{i=1}^{\ell} h_i b_i \ \in \mathbb{Z}_q \quad \text{where } \mathsf{id} = (b_1, \ldots, b_\ell) \in (\mathbb{Z}_q^\ell)^* \text{ and } h = (h_1, \ldots, h_\ell) \in \mathbb{Z}_q^\ell \tag{10}$$

In our application we will only use these hash functions with inputs in $\{1, -1\}^\ell$. Since abort resistance holds for the larger domain $(\mathbb{Z}_q^\ell)^*$ we state the more generate result.

Lemma 25. *Let q be a prime and $0 < Q < q$. Then the hash family $\mathcal{H}_{Wat}$ defined in (10) is $\big(Q,\ \frac{1}{q}\big(1 - \frac{Q}{q}\big),\ \frac{1}{q}\big)$ abort-resistant.*

Proof. The proof uses an argument similar to the one in [38, 26, 7]. Consider a set $\bar{\mathsf{id}}$ of $Q+1$ inputs $\mathsf{id}_0, \ldots, \mathsf{id}_Q$ in $(\mathbb{Z}_q^\ell)^*$ where $\mathsf{id}_0 \notin \{\mathsf{id}_1, \ldots, \mathsf{id}_Q\}$. For $i = 0, \ldots, Q+1$ let S_i be the set of functions H in $\mathcal{H}_{\text{Wat}}$ such that $H(\mathsf{id}_i) = 0$ and observe that $|S_i| = q^{\ell-1}$. Moreover, $|S_0 \cap S_j| \leq q^{\ell-2}$ for every $j > 0$. The set of functions in $\mathcal{H}_{\text{Wat}}$ such that $H(\mathsf{id}_0) = 0$ but $H(\mathsf{id}_i) \neq 0$ for $i = 1, \ldots, Q$ is exactly $S := S_0 \setminus (S_1 \cup \ldots S_Q)$. Now,

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

Therefore the no-abort probability of $\bar{\mathsf{id}}$, which is $|S|/q^{\ell}$, is at least $\frac{1}{q}(1 - \frac{Q}{q})$. The no-abort probability is at most $1/q$ since $|S| \le |S_0| = q^{\ell-1}$. Since $\bar{\mathsf{id}}$ was arbitrary, the lemma follows. □

7.4.2 Proof of Theorem 22

Using these tools we can now prove full IBE security.

Proof of Theorem 22. The proof proceeds in a sequence of games where the first game is identical to the INDr–ID-CPA game from Section 2.1. In the last game in the sequence the adversary has advantage zero. We show that a PPT adversary cannot distinguish between the games which will prove that the adversary has negligible advantage in winning the original INDr–ID-CPA game. The LWE problem is used in proving that Games 3 and 4 are indistinguishable. In game i we let W_i denote the event that $r = r'$ at the end of the game. The adversary's advantage in Game i is $|\Pr[W_i] - \frac{1}{2}|$.

Game 0. This is the original INDr–ID-CPA game from Section 2.1 between an attacker $\mathcal{A}$ against our scheme and an INDr–ID-CPA challenger.

Game 1. Recall that in Game 0 the challenger generates the public parameters PP by choosing $\ell+2$ random matrices $A_0, A_1, \ldots, A_\ell, B$ in $\mathbb{Z}_q^{n\times m}$ such that a trapdoor T_{A_0} is known for $\Lambda_q^{\perp}(A_0)$. At the challenge phase the challenger generates a challenge ciphertext CT^*. We let $R_i^* \in \{-1,1\}^{m\times m}$ for $i = 1,\ldots,\ell$ denote the ℓ ephemeral random matrices generated for the creation of CT^* (in step 3 of Encrypt).

In Game 1 we slightly change the way that the challenger generates the matrices $A_i,\ i \in [1,\ell]$ in the public parameters. The Game 1 challenger chooses $R_i^*, i \in [\ell]$ at the setup phase and also chooses ℓ random scalars $h_i \in \mathbb{Z}_q$ for $i = 1,\ldots,\ell$. Next it generates matrices A_0 and B as in Game 0 and constructs the matrices A_i for $i = 1,\ldots,\ell$ as

$$A_i \leftarrow A_0 \cdot R_i^* - h_i \cdot B \quad \in \mathbb{Z}_q^{n\times m} \tag{11}$$

The remainder of the game is unchanged. Note that the $R_i^* \in \{-1,1\}^{m\times m}$ are chosen in advance, during the setup phase, and that the knowledge of the challenge identity id^* is not needed in order to do so.

We show that Game 0 is statistically indistinguishable from Game 1 by Lemma 11. Observe that in Game 1 the matrices $R_i^*, i \in [\ell]$ are used only in the construction of the matrices A_i and in the construction of the challenge ciphertext where $z \leftarrow (R_{\mathsf{id}}^*)^{\top} y \in \mathbb{Z}_q^m$ (and where $R_{\mathsf{id}}^* = \sum_{i=1}^{\ell} b_i^* R_i^*$). By Lemma 11, the distribution $\big(A_0\ ,\ A_0 \cdot (R_1^* \mid \ldots \mid R_\ell^*),\ z\big)$ is statistically close to the distribution $(A_0,\ (A_1' \mid \ldots \mid A_\ell'),\ z)$ where $A_i',\ i \in [\ell]$ are uniform matrices in $\mathbb{Z}_q^{n\times m}$. It follows that in the adversary's view, the matrices $A_0 R_i^*$ are statistically close to uniform and therefore the A_i as defined in (11) are close to uniform. Hence, all the A_i for $i = 1,\ldots,\ell$ are random independent matrices in the attacker's view, as in Game 0. This shows that

$$\Pr[W_0] = \Pr[W_1] \tag{12}$$

Game 2. Game 2 is identical to Game 1 except that we add an abort event that is independent of the adversary's view. We use the abort-resistant family of hash functions $\mathcal{H}_{\mathrm{Wat}}$ introduced in Lemma 24. Recall that $\mathcal{H}_{\mathrm{Wat}}$ is a $(Q, \epsilon_{\min}, \epsilon_{\max})$ abort-resistant family, where $\epsilon_{\min} = \frac{1}{q}(1 - Q/q)$ by Lemma 24. Since $q = 2Q$ we have $\epsilon_{\min} = 1/(2q)$.

The Game 2 challenger behaves as follows:

- The setup phase is identical to Game 1 except that the challenger also chooses a random hash function $H \in \mathcal{H}_{\text{Wat}}$ and keeps it to itself.
- The challenger responds to identity queries and issues the challenge ciphertext exactly as in Game 1 (using a random bit $r \in \{0,1\}$ to select the type of challenge). Let $\mathsf{id}_1, \ldots, \mathsf{id}^Q$ be the identities where the attacker queries and let id^* be the challenge identity. By definition, id^* is not in $\{\mathsf{id}_1, \ldots, \mathsf{id}_Q\}$.
- In the final guess phase, the attacker outputs its guess $r' \in \{0,1\}$ for r. The challenger now does the following:
  1. **Abort check:** the challenger checks if $H(\mathsf{id}^*) = 0$ and $H(\mathsf{id}_i) \neq 0$ for $i = 1, \ldots, Q$. If not, it overwrites r' with a fresh random bit in $\{0,1\}$ and we say that the challenger aborted the game. Note that the adversary never sees H and has no idea if an abort event took place. While it is convenient to describe this abort at the end of the game, nothing would change if the challenger aborted the game as soon as the abort condition becomes true.
  2. **Artificial abort:** the challenger samples a bit $\Gamma \in \{0,1\}$ such that $\Pr[\Gamma = 1] = \gamma(\mathsf{id}^*, \mathsf{id}_1, \ldots, \mathsf{id}_q)$ where the function $\gamma(\cdot)$ is defined in Lemma 25 below. If $\Gamma = 1$ the challenger r' with a fresh random bit in $\{0,1\}$ and we say that the challenger aborted the game due to an artificial abort. The reason for this step is explained in Lemma 25.

This completes the description of Game 2. Note that the abort condition is determined using a hash function H that is independent of the attacker's view.

Lemma 26. *For $i = 1,2$ let W_1 be the event that $r = r'$ at the end of Game i. Then*

$$\left|\Pr[W_2] - \frac{1}{2}\right| \geq 2\epsilon_{min} \left|\Pr[W_1] - \frac{1}{2}\right| \tag{13}$$

So as not to interrupt the proof of Theorem 22, we come back to this lemma at the end of the proof where we also define the function $\gamma(\cdot)$.

Game 3. We now change how A_0 and B in Game 2 are chosen. In Game 3 we generate A_0 as a random matrix in $\mathbb{Z}_q^{n \times m}$, but generate B using algorithm $\mathsf{TrapGen}$ so that B is a random matrix in $\mathbb{Z}_q^{n \times m}$ for which the challenger has a trapdoor T_B for $\Lambda_q^{\perp}(B)$. The construction of A_i for $i = 1, \ldots, \ell$ remains as in Game 2, namely, $A_i = A_0 \cdot R_i^* - h_i \cdot B$.

The challenger responds to private key queries using the trapdoor T_B. To respond to a private key query for $\mathsf{id} = (b_1, \ldots, b_\ell) \in \{1, -1\}^\ell$ the challenger needs a short vector $e \in \Lambda_q^u(F_{\mathsf{id}})$ where

$$F_{\mathsf{id}} := \Big(\; A_0 \; \Big| \; B + \sum_{i=1}^{\ell} b_i \, A_i \; \Big) = \Big(A_0 \mid A_0 \, R_{\mathsf{id}} + h_{\mathsf{id}} B \Big)$$

and where

$$R_{\mathsf{id}} \leftarrow \sum_{i=1}^{\ell} b_i \, R_i^* \in \mathbb{Z}_q^{m \times m} \quad \text{and} \quad h_{\mathsf{id}} \leftarrow 1 + \sum_{i=1}^{\ell} b_i \, h_i \in \mathbb{Z}_q \tag{14}$$

Note that $h_{\mathsf{id}} = H(\mathsf{id})$ where H the hash function in $\mathcal{H}_{\text{Wat}}$ defined by $(h_1, \ldots, h_\ell)$.

The challenger now does the following:

1. Construct h_{id} and R_{id} as in (14). If $h_{\mathsf{id}} = 0$ abort the game and pretend that the adversary outputs a random bit r' in $\{0, 1\}$, as in Game 2.

2. Set $e \leftarrow \mathsf{SampleRight}(A_0,\ h_{\mathsf{id}}B,\ R_{\mathsf{id}},\ T_B, u, \sigma) \in \mathbb{Z}_q^{2m}$.

3. Send $\mathsf{SK}_{\mathsf{id}} := e$ to $\mathcal{A}$.

Since h_{id} in Step 2 is non-zero the set T_B is also a trapdoor for $h_{\mathsf{id}}B$. Moreover, since B is rank n w.h.p so is $h_{\mathsf{id}}B$. Theorem 16 shows that when $\sigma > \|\widetilde{T_B}\| s_R\, \omega(\sqrt{\log m})$ with $s_R := \|R_{\mathsf{id}}\|$, the generated e is distributed close to $D_{\Lambda_q^u(F_{\mathsf{id}}),\sigma}$, as in Game 2. Recall that $\|\widetilde{T_B}\| \le \sigma_{\mathrm{TG}}$ by Theorem 4 and

$$s_R = \|R_{\mathsf{id}}\| \le \sum_{i=1}^{\ell} \|R_i^*\| = O(\ell\sqrt{m})$$

w.h.p by Lemma 13. Therefore σ used in the system, as defined in (9), is sufficiently large to satisfy the conditions of Theorem 16.

Game 3 is otherwise the same as Game 2. In particular, in the challenge phase the challenger checks if the challenge identity $\mathsf{id}^* = (b_1^*, \dots, b_\ell^*) \in \{1, -1\}^\ell$ satisfies $h_{\mathsf{id}^*} := 1 + \sum_{i=1}^{\ell} b_i^*\, h_i = 0$. If not, the challenger aborts the game (and pretends that the adversary output a random bit r' in $\{0, 1\}$), as in Game 2. Similarly, in Game 3 the challenger implements an artificial abort in the guess phase.

Since Games 2 and 3 are identical in the attacker's view (the public parameters, responses to private key queries, the challenge ciphertext, and abort conditions) the adversary's advantage in Game 3 is identical to its advantage in Game 2, namely

$$\Pr[W_2] = \Pr[W_3] \tag{15}$$

Game 4. Game 4 is identical to Game 3 except that the challenge ciphertext (c_0^*, c_1^*) is *always* chosen as a random independent element in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$. Since the challenge ciphertext is always a fresh random element in the ciphertext space, $\mathcal{A}$'s advantage in this game is zero.

It remains to show that Games 3 and 4 are computationally indistinguishable for a PPT adversary, which we do by giving a reduction from the LWE problem. If an abort event happens then the games are clearly indistinguishable. Therefore, it suffice to focus on sequences of queries that do not cause an abort.

Reduction from LWE. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing Games 3 and 4. We use $\mathcal{A}$ to construct an LWE algorithm denoted $\mathcal{B}$.

Recall from Definition 7 that an LWE problem instance is provided as a sampling oracle $\mathcal{O}$ which can be either truly random $\mathcal{O}_\$$ or a noisy pseudo-random $\mathcal{O}_s$ for some secret $s \in \mathbb{Z}_q^n$. The simulator $\mathcal{B}$ uses the adversary $\mathcal{A}$ to distinguish between the two, and proceeds as follows:

Instance. $\mathcal{B}$ requests from $\mathcal{O}$ and receives, for each $i = 0, \dots, m$, a fresh pair $(u_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Setup. $\mathcal{B}$ constructs the system's public parameters PP as follows:

1. Assemble the random matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ from m of the previously given LWE samples by letting the i-th column of A_0 be the n-vector u_i for all $i = 1, \dots, m$.
2. Assign the zeroth LWE sample (so far unused) to become the public random n-vector $u_0 \in \mathbb{Z}_q^n$.

3. The remainder of the public parameters, namely A_i, $i > 0$ and B, are constructed as in Game 3 using random h_i and R_i^*.

Queries. $\mathcal{B}$ answers each private-key extraction query as in Game 3.

Challenge. $\mathcal{B}$ prepares, when prompted by $\mathcal{A}$ with a message bit $b^* \in \{0,1\}$ and a target identity $\mathsf{id}^* = (b_1^*, \dots, b_\ell^*)$, a challenge ciphertext for the target identity id^*, as follows:

1. Let $v_0, \dots, v_m$ be entries from the LWE instance. Set $v^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m$.
2. Blind the message bit by letting $c_0^* = v_0 + b^* \lfloor \frac{q}{2} \rceil \in \mathbb{Z}_q$.
3. Set $R_{\mathsf{id}^*}^* := \sum_{i=1}^{\ell} b_i^* \, R_i^*$.
4. Set $c_1^* = \begin{bmatrix} v^* \\ (R_{\mathsf{id}^*}^*)^\top v^* \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
5. Choose a random bit $r \xleftarrow{R} \{0,1\}$. If $r = 0$ send $\mathsf{CT}^* = (c_0^*, c_1^*)$ to the adversary. If $r = 1$ choose a random $(c_0, c_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$ and send (c_0, c_1) to the adversary.

We argue that when the LWE oracle is pseudorandom (i.e. $\mathcal{O} = \mathcal{O}_s$) then CT^* is distributed exactly as in Game 3. First, since $h_{\mathsf{id}^*} = 0$ we have that $F_{\mathsf{id}^*} = (A_0 \mid A_0 R_{\mathsf{id}^*}^*)$. Second, by definition of $\mathcal{O}_s$ we know that $v^* = A_0^\top s + y$ for some random noise vector $y \in \mathbb{Z}_q^m$ distributed as $\bar{\Psi}_\alpha^m$. Therefore, c_1^* defined in step (3) above satisfies

$$c_1^* = \begin{bmatrix} A_0^\top s + y \\ (R_{\mathsf{id}^*}^*)^\top A_0^\top s + (R_{\mathsf{id}^*}^*)^\top y \end{bmatrix} = \begin{bmatrix} A_0^\top s + y \\ (A_0 R_{\mathsf{id}^*}^*)^\top s + (R_{\mathsf{id}^*}^*)^\top y \end{bmatrix} = (F_{\mathsf{id}^*})^\top s + \begin{bmatrix} y \\ (R_{\mathsf{id}^*}^*)^\top y \end{bmatrix}$$

and the quantity on the right is precisely the c_1 part of a valid challenge ciphertext in Game 3. Also note that $v_0 = u_0^\top s + x$, just as the c_0 part of the challenge ciphertext in Game 3.

When $\mathcal{O} = \mathcal{O}_\$$ we have that v_0 is uniform in $\mathbb{Z}_q$ and v^* is uniform in $\mathbb{Z}_q^m$. Therefore c_1^* as defined in step (3) above is uniform and independent in $\mathbb{Z}_q^{2m}$ by the standard left over hash lemma (e.g. Theorem 8.38 of [36]) where the hash function is defined by the matrix $(A_0^\top | v^*)$. Consequently, the challenge ciphertext is always uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$, as in Game 4.

Guess. After being allowed to make additional queries, $\mathcal{A}$ guesses if it is interacting with a Game 3 or Game 4 challenger. Our simulator implements the artificial abort from Games 3 and 4 and outputs the final guess as the answer to the LWE challenge it is trying to solve.

We already argued that when $\mathcal{O} = \mathcal{O}_s$ the adversary's view is as in Game 3. When $\mathcal{O} = \mathcal{O}_\$$ the adversary's view is as in Game 3. Hence, $\mathcal{B}$'s advantage in solving LWE is the same as $\mathcal{A}$'s advantage in distinguishing Games 3 and 4, as required. This completes the description of algorithm $\mathcal{B}$ and since $\Pr[W_4] = 1/2$ we obtain

$$|\Pr[W_3] - \frac{1}{2}| = |\Pr[W_3] - \Pr[W_4]| \leq \mathsf{LWE\text{-}adv}[\mathcal{B}] \tag{16}$$

Summary. Combining (12), (13), (15) and (16) we obtain that

$$|\Pr[W_0] - \frac{1}{2}| \le (1/\epsilon_{\min})\ \mathsf{LWE\text{-}adv}[\mathcal{B}] = 4q \cdot \mathsf{LWE\text{-}adv}[\mathcal{B}]$$

as required in the statement of Theorem 22. □

Proof of Lemma 25. To complete the proof of Theorem 22 we prove Lemma 25 which expresses the advantage of the adversary in Game 2 as a function of its advantage in Game 1.

Proof of Lemma 25. A detailed analysis of the effect of the abort condition in Game 2 is given in [7]. We review the main points here. Let $\mathcal{I} := (\{1,-1\}^{\ell})^{Q+1}$ be the set of all $(Q+1)$-tuples of identities. For $I \in \mathcal{I}$:

- Let $\kappa(I)$ be the event that in Game 1 the adversary uses the first entry in I as the challenge ciphertext and issues identity queries for the remaining Q identities. Then $\sum_{i\in\mathcal{I}} \kappa(I) = 1$.
- Let $\beta_2(I) \subseteq \kappa(I)$ be the event that $r = r'$ in Game 2 when $\kappa(I)$ happens. Similarly, let $\beta_1(I) \subseteq \kappa(I)$ be the event that $r = r'$ in Game 1 when $\kappa(I)$ happens. Then $\sum_{I\in\mathcal{I}} \beta_i(I) = \Pr[W_i]$ for $i = 1, 2$.
- Let $\mathcal{E}$ be the event that that challenger aborts the game in the guessing phase of Game 2 and let $\epsilon(I) := \Pr[\neg\mathcal{E} \mid \kappa(I)]$. Then $\epsilon(I) \in [\epsilon_{\min}, \epsilon_{\max}]$.

Then for all $I \in \mathcal{I}$ we have

$$\begin{aligned}
\Pr\left[\beta_2(I)\ \wedge\ \mathcal{E}\right] &= \Pr[\ (r = r')\ \wedge\ \kappa(I)\ \wedge\ \mathcal{E}] = {}^1\!/\!_2\ \Pr[\kappa(I)\ \wedge\ \mathcal{E}] \\
\Pr\left[\beta_2(I)\ \wedge\ \neg\mathcal{E}\right] &= \Pr\left[\beta_1(I)\ \wedge\ \neg\mathcal{E}\right] = \Pr\left[\beta_1(I)\right]\ \epsilon(I) \\
\Pr\left[\kappa(I)\ \wedge\ \neg\mathcal{E}(I)\right] &= \Pr[\kappa(I)]\ \epsilon(I)
\end{aligned}$$

Since $\Pr[\beta_2(I)] = \Pr[\beta_2(I) \wedge \mathcal{E}(I)] + \Pr[\beta_2(I) \wedge \neg\mathcal{E}(I)]$ and the same holds for $\kappa(I)$ we obtain:

$$\begin{aligned}
\left|\Pr[W_2] - \frac{1}{2}\right| &= \left|\sum_{I\in\mathcal{I}} \Big(\Pr\left[\beta_2(I)\right] - {}^1\!/\!_2\ \Pr\left[\kappa(I)\right]\Big)\right| \\
&= \left|\sum_{I\in\mathcal{I}} \Big(\Pr\left[\beta_1(I)\right] - {}^1\!/\!_2\ \Pr\left[\kappa(I)\right]\Big)\ \epsilon(I)\ \right| .
\end{aligned}$$

To lower bound this expression we separate the positive and negative terms and use the fact that $\epsilon(I) \in [\epsilon_{\min}, \epsilon_{\max}]$. We also use the fact that

$$\left|\Pr[W_1] - \frac{1}{2}\right| = \left|\sum_{I\in\mathcal{I}} \Big(Pr\left[\beta_1(I)\right] - {}^1\!/\!_2\ \Pr\left[\kappa(I)\right]\Big)\right| \le {}^1\!/\!_2$$

and obtain

$$\left|\Pr[W_2] - \frac{1}{2}\right| \ge \epsilon_{\min} \left|\Pr[W_1] - \frac{1}{2}\right| - {}^1\!/\!_2(\epsilon_{\max} - \epsilon_{\min})\ .$$

If $\epsilon(I)$ were the same for all $I \in \mathcal{I}$ then $\epsilon_{\max} = \epsilon_{\min}$ and we would be done. Unfortunately, without the artificial abort, $\epsilon(I)$ are not all the same. There are two ways to deal with this. Waters [38] introduces an artificial abort so that $\epsilon(I)$ is close to its minimum value $\epsilon_{\min}$ for all $I \in \mathcal{I}$. Bellare and Ristenpart [7] set the parameters in the reduction so that $(\epsilon_{\max}-\epsilon_{\min})$ is negligible. In our case, $(\epsilon_{\max}-\epsilon_{\min})$ is bounded by Q/q^2 and therefore the latter approach would require making q larger which would negatively impact the performance of the system. We therefore opt for the Waters approach and introduce an artificial abort. The analysis of the artificial abort and the definition of the function $\gamma(I)$ are as in [38]. With the artificial abort, $(\epsilon_{\max}-\epsilon_{\min})$ is less than $|\Pr[W_1] - 1/2|$ and the proof of (13) is complete. $\square$

7.5 Further Improvements and Extensions

Smaller q. It is possible to decouple the choice of modulus q from the number of key extraction queries Q, by using one of the techniques used in the very recent lattice-based signature scheme of [14]. The benefit is that then q can be chosen as small as possible, which improves the overall time and space efficiency of the scheme.

Smaller ℓ. The value of ℓ can be reduced, thus shrinking the ciphertext and private key size. Recall that the system above treats identities as elements in $\{1,-1\}^\ell$. It is possible to treat identities as elements in $\{-B,\dots,B\}^{\ell'}$ and then get away with a smaller ℓ' since we would only need $B^{\ell'} > 2^\ell$. Lemma 24 holds when identities are in $\mathbb{Z}_q^\ell$ (i.e. $B = q/2$). However, the norm of the matrix R_{id} used in encryption and in the simulation will grow with B. Therefore, while ℓ' shrinks, the size of q will need to grow to compensate for the increased norm of R_{id}. Clearly setting $B = 1$ is sub-optimal, but we do not calculate the optimal B here.

Multi-bit encryption. The system can encrypt multiple bits at once using the same method used in Section 6.5.

Induced signature scheme. The full IBE system gives rise to an existentially unforgeable signature scheme via the standard conversion of IBE to a signature [11] . Security follows from the ISIS problem rather than LWE. **Describe signature scheme.**

8 Extension 2: Hierarchical IBE

We show how the basis delegation techniques from [17, 31] can convert the basic IBE construction to an HIBE. Identities are vectors $\mathsf{id} = (\mathsf{id}_1,\dots,\mathsf{id}_\ell)$ where the id_i are in $\mathbb{Z}_q^n \setminus \{0\}$. Excluding 0 coordinates is necessary for the proof of security.

For a hierarchy of maximum depth d the public parameters will contain random matrices $A_0, A_1, \dots, A_\ell, B$ in $\mathbb{Z}_q^{n\times m}$. Then to encrypt to an identity $\mathsf{id} = (\mathsf{id}_1,\dots,\mathsf{id}_\ell)$ at depth $\ell \le d$ we use the matrix

$$F_{\mathsf{id}} := \big(\ A_0 \mid A_1 + H(\mathsf{id}_1)B \mid \ \dots\ \mid A_\ell + H(\mathsf{id}_\ell)B\ \big) \quad \in \mathbb{Z}_q^{n\times(\ell+1)m} \tag{17}$$

where H is the FRD map $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ defined in Section 5. The master key is a trapdoor T_{A_0} for A_0, as in the basic scheme. The secret key for the identity id consists of a short basis for the lattice $\Lambda_q^\perp(F_{\mathsf{id}})$. Key delegation, namely where user id issues a secret key to user $(\mathsf{id}|\mathsf{id}_{\ell+1})$, is done by using the short basis for $\Lambda_q^\perp(F_{\mathsf{id}})$ to generate a random short basis for $\Lambda_q^\perp(F_{\mathsf{id}|\mathsf{id}_{\ell+1}})$.

In the selective security reduction, the attacker declares at the beginning of the game an identity $\mathsf{id}^* = (\mathsf{id}_1^*, \ldots, \mathsf{id}_\ell^*)$ for $\ell \le d$ where it wishes to be challenged. If $\ell < d$ we pad id^* with $d - \ell$ zeros so that id^* is in $(\mathbb{Z}_q^n)^d$. The simulator chooses the matrices A_0 and B uniformly at random in $\mathbb{Z}_q^{n\times m}$ and constructs the matrices A_i for $i = 1, \ldots, d$ as

$$A_i := A_0 R_i - H(\mathsf{id}_i^*) B \quad \text{for } i = 1, \ldots, d$$

where all the matrices R_i are random in $\{1, -1\}^{m\times m}$. As in Section 6.4 the matrices $A_0\, R_i$ are uniform and independent in $\mathbb{Z}_q^{n\times m}$ and therefore $A_i,\ i = 1, \ldots, d$ are uniform in $\mathbb{Z}_q^{n\times m}$ as in the real system. We provide the simulator with a trapdoor T_B for $\Lambda_q^\perp(B)$, but no trapdoor for $\Lambda_q^\perp(A_0)$.

Now, to respond to a private key query for an identity $\mathsf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$, the simulator must produce a short basis for $\Lambda_q^\perp(F_{\mathsf{id}})$ where

$$F_{\mathsf{id}} = \Big(A_0 \;\Big|\; A_0 R_{\mathsf{id}} + B_{\mathsf{id}} \Big) \quad \in \mathbb{Z}_q^{n\times(\ell+1)m} \tag{18}$$

and where

$$R_{\mathsf{id}} := (R_1 \mid \ldots \mid R_\ell) \quad \in \mathbb{Z}_q^{m\times \ell m} \tag{19}$$

$$B_{\mathsf{id}} := \Big(\big(H(\mathsf{id}_1) - H(\mathsf{id}_1^*)\big)\ B \;\Big|\; \ldots \;\Big|\; \big(H(\mathsf{id}_\ell) - H(\mathsf{id}_\ell^*)\big)\ B\ \Big) \quad \in \mathbb{Z}_q^{n\times \ell m} \tag{20}$$

When id is not a prefix of id^* we know that $H(\mathsf{id}_i) - H(\mathsf{id}_i^*)$ is full rank for some $i \in \{1, \ldots, \ell\}$. We show that the simulator can then extend T_B to a short basis for the entire lattice $\Lambda_q^\perp(B_{\mathsf{id}})$. The simulator can now generate short vectors in $\Lambda_q^\perp(F_{\mathsf{id}})$ using algorithm SampleRight, which is sufficient for constructing the required short basis for $\Lambda_q^\perp(F_{\mathsf{id}})$.

When id is a prefix of id^* the matrix F_{id} no longer depends on B and the simulator's trapdoor disappears. Consequently, the simulator can generate private keys for all identities other than prefixes of id^*. As we will see, for id^* the simulator can produce a challenge ciphertext that helps it solve the given LWE challenge.

8.1 Sampling a random basis

Let Λ be an m-dimenaional lattice and let $\mathcal{O}(\Lambda, \sigma)$ be an algorithm that generates independent samples from a distribution statistically close to $\mathcal{D}_{\Lambda,\sigma}$. The following algorithm called $\mathsf{SampleBasis}^{\mathcal{O}}(\Lambda, \sigma)$ uses $\mathcal{O}$ to genenerate a basis T of Λ:

1. For $i = 1, \ldots, m$, let $v \leftarrow \mathcal{O}(\Lambda, \sigma)$ and
   if v is independent of $\{v_1, \ldots, v_{i-1}\}$, set $v_i \leftarrow v$, if not, repeat.
2. Convert the set of independent vectors $v_1, \ldots, v_m$ to a basis T using Lemma 3 (and using some canonical basis of Λ) and output T.

The following theorem summarizes properties of this algorithm. Recall that Gentry et al. [22, Sec. 3] define $\tilde{bl}(\Lambda)$ as the Gram-Schmidt norm of the shortest bases of Λ, namely $\tilde{bl}(\Lambda) := \min_T \|\tilde{T}\|$ where the minimum is over all ordered bases T of Λ.

Lemma 27. *For $\sigma > \tilde{bl}(\Lambda)\ \omega(\sqrt{\log m})$ algorithm $\mathsf{SampleBasis}^{\mathcal{O}}(\Lambda, \sigma)$ satisfies the following properties:*

1. *Step 1 requires at most $O(m \log m)$ w.h.p and $2m$ samples in expectation.*

2. *With overwhelming probability* $\|\widetilde{T}\| \le \|T\| \le \sigma\sqrt{m}$.
3. *Up to a statistical distance, the distribution of* T *does not depend on the implementation of* $\mathcal{O}$. *That is, the random variable* $\mathsf{SampleBasis}^{\mathcal{O}}(\sigma)$ *is statistically close to* $\mathsf{SampleBasis}^{\mathcal{O}'}(\sigma)$ *for any algorithm* $\mathcal{O}'$ *that samples from a distribution statistically close to* $\mathcal{D}_{\Lambda,\sigma}$.

Proof. Part (1) follows from Lemma 6 part (2) where the expectation statement is by a result from [2]. Part (2) follows from Lemma 6 part (1). Part (3) is immediate. □

Algorithm SampleBasisLeft. The lattice Λ of interest is $\Lambda_q^{\perp}(F_{\mathsf{id}})$ where F_{id} is defined in (17) for $\mathsf{id} = (\mathsf{id}_1, \dots, \mathsf{id}_\ell)$. Write $F_{\mathsf{id}} = (A \mid M)$ for some matrices A and M, then given a short basis T_A for $\Lambda_q^{\perp}(A)$ we can implement algorithm $\mathcal{O}(F_{\mathsf{id}}, \sigma)$ as

$$\mathcal{O}_{\mathrm{SL}}(F_{\mathsf{id}}, \sigma) := \mathsf{SampleLeft}(A, M, T_A, 0, \sigma) \ .$$

When $\sigma > \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log(\ell m)})$ Theorem 14 shows that the resulting vector is distributed statistically close to $\mathcal{D}_{\Lambda_q^{\perp}(F_{\mathsf{id}}),\sigma}$, as required for $\mathsf{SampleBasis}$.

Using $\mathcal{O}_{\mathrm{SL}}$ in algorithm $\mathsf{SampleBasis}$ leads to an algorithm to sample a random basis of $\Lambda_q^{\perp}(F_{\mathsf{id}})$ given a short basis of A. We refer to this algorithm as $\mathsf{SampleBasisLeft}(A,\ M,\ T_A,\ \sigma)$ and summarize its properties in the following corollarly.

Corollary 28. *Algorithm* $\mathsf{SampleBasisLeft}(A,\ M,\ T_A,\ \sigma)$ *outputs a basis of* $\Lambda_q^{\perp}(F_{\mathsf{id}})$ *satisfying the three properties in Lemma* **??** *provided that* A *is rank* n *and* $\sigma > \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log(\ell m)})$.

Algorithm SampleBasisRight. In the simulation, the matrix F_{id} is defined as in (18). In this case, given a short basis T_B for $\Lambda_q^{\perp}(B)$ we can implement algorithm $\mathcal{O}$ as follows:
Algorithm $\mathcal{O}_{\mathrm{SR}}(F_{\mathsf{id}}, \sigma)$:

1. First, use Theorem **??** to extend the basis T_B for $\Lambda_q^{\perp}(B)$ to a basis T for $\Lambda_q^{\perp}(B_{\mathsf{id}})$ such that $\|\widetilde{T}\| = \|\widetilde{T_B}\|$.

2. Then run $\mathsf{SampleRight}(A_0,\ B_{\mathsf{id}},\ R_{\mathsf{id}},\ T_B,\ 0,\ \sigma)$ and output the result.
   When B is rank n and id is not a prefix of id^* the matrix B_{id} is rank n as required for $\mathsf{SampleRight}$.

Let $s_R := \|R_{\mathsf{id}}\|$ be the norm of the matrix R_{id}. When $\sigma > \|\widetilde{T_B}\| \cdot s_R\ \omega(\sqrt{\log m})$ Theorem 16 shows that the resulting vector is distributed statistically close to $\mathcal{D}_{\Lambda_q^{\perp}(F_{\mathsf{id}}),\sigma}$, as required for $\mathsf{SampleBasis}$.

Using $\mathcal{O}_{\mathrm{SR}}$ in algorithm $\mathsf{SampleBasis}$ leads to an algorithm to sample a random basis of $\Lambda_q^{\perp}(F_{\mathsf{id}})$ for F_{id} defined in (18) given a short basis of B. We refer to this algorithm as $\mathsf{SampleBasisRight}(A_0,\ B_{\mathsf{id}},\ R_{\mathsf{id}},\ T_B,$ and summarize its properties in the following corollary.

Corollary 29. *Algorithm* $\mathsf{SampleBasisRight}(A_0,\ B_{\mathsf{id}},\ R_{\mathsf{id}},\ T_B,\ \sigma)$ *outputs a basis of* $\Lambda_q^{\perp}(F_{\mathsf{id}})$ *satisfying the three properties in Lemma* **??** *provided that* B *is rank* n, *that* id *is not a prefix of* id^*, *and that* $\sigma > \|\widetilde{T_B}\| \cdot s_R\ \omega(\sqrt{\log m})$ *where* $s_R := \|R_{\mathsf{id}}\|$.

An important fact for the proof of security is that for sufficiently large σ algorithms $\mathsf{SampleBasisLeft}$ and $\mathsf{SampleBasisRight}$ sample a basis of F_{id} from statistically close distributions.

8.2 The HIBE Construction

We can now describe the HIBE system. Recall that identities are vectors $\mathsf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell)$ where the id_i are in $\mathbb{Z}_q^n \setminus \{0\}$.

Setup(d, λ)**:** On input a security parameter λ, and a maximum hierarchy depth d, set the parameters $q, n, m, \bar{\sigma}, \bar{\alpha}$ as specified in Section 8.3 below. The vectors $\bar{\sigma}$ and $\bar{\alpha}$ live in $\mathbb{R}^d$ and we use σ_ℓ and α_ℓ to refer to their ℓ-th coordinate. Next do:

1. Use algorithm $\mathsf{TrapGen}(q, n)$ to select a uniformly random $n \times m$-matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ with a basis T_{A_0} for $\Lambda_q^\perp(A_0)$ such that $\|\widetilde{T_{A_0}}\| \le O(\sqrt{n \log q})$
2. Select $d+1$ uniformly random $n \times m$ matrices $A_1, \ldots, A_d$ and B in $\mathbb{Z}_q^{n\times m}$.
3. Select a uniformly random n-vector $u \xleftarrow{R} \mathbb{Z}_q^n$.
4. Output the public parameters and master key,

$$\mathsf{PP} = \Big(\; A_0, A_1, \ldots, A_d, B, u \; \Big) \qquad ; \qquad \mathsf{MK} = \Big(\; T_{A_0} \; \Big) \quad \in \mathbb{Z}^{m\times m}$$

Derive$(\mathsf{PP},\ (\mathsf{id}|\mathsf{id}_\ell),\ \mathsf{SK}_{\mathsf{id}})$**:** On input public parameters PP and a secret key corresponding to an identity id at depth $\ell-1$ the algorithm outputs a secret key for the identity $(\mathsf{id}|\mathsf{id}_\ell)$ at depth ℓ. It works as follows:

Recall that $\mathsf{SK}_{\mathsf{id}}$ is a short basis for $\Lambda_q^\perp(F_{\mathsf{id}})$ and let $F_{\mathsf{id}|\mathsf{id}_\ell} = \big(F_{\mathsf{id}} \;\big|\; A_\ell + H(\mathsf{id}_\ell)B\big)$. Construct short basis for $\Lambda_q^\perp(F_{\mathsf{id}|\mathsf{id}_\ell})$ by invoking

$$S \leftarrow \mathsf{SampleBasisLeft}\Big(F_{\mathsf{id}},\ \big(A_\ell + H(\mathsf{id}_\ell)B\big),\ \mathsf{SK}_{\mathsf{id}},\ \sigma_\ell\Big)$$

and output $\mathsf{SK}_{\mathsf{id}|\mathsf{id}_\ell} \leftarrow S$.

Algorithm $\mathsf{Extract}(\mathsf{PP},\ (\epsilon|\mathsf{id}),\ \mathsf{MK})$ works the same way by setting $F_\epsilon := A_0$.

Encrypt$(\mathsf{PP}, \mathsf{id}, b)$**:** On input public parameters PP, an identity id at depth ℓ and a message $b \in \{0, 1\}$ do:

1. Build the identity-based encryption matrix

$$F_{\mathsf{id}} := \big(\; A_0 \mid A_1 + H(\mathsf{id}_1)B \mid \ \ldots \ \mid A_\ell + H(\mathsf{id}_\ell)B \;\big) \quad \in \mathbb{Z}_q^{n\times(\ell+1)m}$$

2. Choose a uniformly random vector $s \xleftarrow{R} \mathbb{Z}_q^n$.
3. Choose ℓ uniformly random $m \times m$ matrices $R_1, \ldots, R_\ell$ in $\{-1, 1\}^{m\times m}$.
4. Choose noise vectors $x \xleftarrow{\bar{\Psi}_{\alpha_\ell}} \mathbb{Z}_q$ and $y \xleftarrow{\bar{\Psi}_{\alpha_\ell}^m} \mathbb{Z}_q^m$, set $z_i \leftarrow R_i^\top y \in \mathbb{Z}_q^m$ for $i = 1, \ldots, \ell$ (the distribution $\bar{\Psi}_\alpha$ is as in Definition 8).
5. Let z be the vector $(y, z_1, \ldots, z_\ell) \in \mathbb{Z}_q^{(\ell+1)m}$ and output the ciphertext,

$$\mathsf{CT} = \Big(\; c_0 = u_0^T s + x + b \lfloor \frac{q}{2} \rfloor \;, \;\; c_1 = F_{\mathsf{id}}^T s + z \; \Big) \quad \in \mathbb{Z}_q \times \mathbb{Z}_q^{(\ell+1)m}$$

Decrypt$(\mathsf{PP}, \mathsf{SK}_{\mathsf{id}}, \mathsf{CT})$**:** On input public parameters PP, a private key $\mathsf{SK}_{\mathsf{id}}$ where id is an identity at depth ℓ, and a ciphertext $\mathsf{CT} = (c_0, c_1)$:

1. Set $e_{\mathsf{id}} \leftarrow \mathsf{SamplePre}(F_{\mathsf{id}},\ \mathsf{SK}_{\mathsf{id}},\ u_0,\ \sigma_\ell)$. Then $F_{\mathsf{id}}\, e_{\mathsf{id}} = u_0$ and $\|e_{\mathsf{id}}\| \le \sigma_\ell \sqrt{m(\ell+1)}$.
2. Compute $w = c_0 - e_{\mathsf{id}}^T c_1 \in \mathbb{Z}_q$.
3. Compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in $\mathbb{Z}$.
   If they are close, i.e., if $\left| w - \lfloor \frac{q}{2} \rfloor \right| < \lfloor \frac{q}{4} \rfloor$ in $\mathbb{Z}$, output 1. Otherwise output 0.

8.3 Parameters and Correctness

When the cryptosystem is operated as specified, we have,

$$w \ = c_0 - e_{\mathsf{id}}^{\top} c_1 = b \lfloor \frac{q}{2} \rfloor + \underbrace{x - e_{\mathsf{id}}^{\top} z}_{\text{error term}}$$

Lemma 30. *The norm of the error term is bounded by* $[q\ell^2 \sigma_\ell m \alpha_\ell\ \omega(\sqrt{\log m}) + O(\ell^2 \sigma_\ell m^{3/2})]$ *w.h.p.*

Proof. The proof is essentially the same as the proof of Lemma 19. $\square$

Now, for the system to work correctly we need to ensure that for all $1 \le \ell \le d$:

- the error term is less than $q/5$ w.h.p (i.e. $\alpha_\ell < [\sigma_\ell m \omega(\sqrt{\log m})]^{-1}$ and $q = \Omega(\sigma_\ell m^{3/2})$),
- that $\mathsf{TrapGen}$ can operate (i.e. $m > 6n \log q$),
- that σ_ℓ is sufficiently large for $\mathsf{SampleBasisLeft}$ and $\mathsf{SampleBasisRight}$ (i.e. $\sigma_\ell > \sigma_{\mathrm{TG}} \sqrt{m}\ \omega(\sqrt{\log m}\,) = m\ \omega(\sqrt{\log m}\,)$), and
- that Regev's reduction applies (i.e. $q > 2\sqrt{n}/\alpha_\ell$)

To satisfy these requirements we set the parameters $(q, m, \bar{\sigma}, \bar{\alpha})$ as follows, taking n to be the security parameter:

$$\begin{aligned} m &= 6\, n^{1+\delta} \quad , \quad q = m^{2.5} \cdot \omega(\sqrt{\log n}\,) \\ \sigma_\ell &= m \cdot \omega(\sqrt{\log n}\,) \quad , \quad \alpha_\ell = [m^2 \cdot \omega(\sqrt{\log n}\,)]^{-1} \quad \text{for } \ell = 1, \dots, d. \end{aligned} \tag{21}$$

and round up m to the nearest larger integer and q to the nearest larger prime. Here we assume that δ is such that $n^\delta > \lceil \log q \rceil = O(\log n)$.

Since the matrices A_0, B are random in $\mathbb{Z}_q^{n \times m}$ and $m > n \log q$, with overwhelming probability both matrices will have rank n. Hence, calling $\mathsf{SampleBasisLeft}$ in algorithm $\mathsf{Extract}$ succeeds w.h.p.

8.4 Security Reduction

We show that the HIBE construction is indistinguishable from random under a selective identity attack as in Definition 1.

Theorem 31. *The HIBE system with parameters* $(q, n, m, \bar{\sigma}, \bar{\alpha})$ *as in (21) is* $\mathsf{INDr\text{–}sID\text{-}CPA}$ *secure provided that the* $(\mathbb{Z}_q, n, \bar{\Psi}_{\alpha_d})$*-LWE assumption holds.*

9 Conclusion and Open Problems

We constructed an efficient identity-based encryption scheme and proven its security in the standard model from the LWE assumption (which is itself implied by worst-case lattice assumptions). We showed that the basic selective-ID secure scheme extends to provide full adaptive-ID security and to support a delegation mechanism to make it hierarchical.

It would be interesting to improve these constructions by adapting them to ideal lattices [37]. Another open problem is to construct an adaptively secure lattice-based IBE in the standard model where all the data is short (including the public parameters).

Acknowledgments

We are grateful to Chris Peikert for suggesting that we use the basis extension method from [31] to simplify the analysis of algorithm SampleLeft. This suggestion also let us to remove the matrix R from the master secret. We also thank Ron Rivest for pointing out that indistinguishability from random can help IBE systems resist subpoenas.

References

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of Eurocrypt'10*, 2010.

[2] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. Manuscript, 2010.

[3] Shweta Agrawal and Xavier Boyen. Identity-based encryption from lattices in the standard model. Manuscript, 2009. `http://www.cs.stanford.edu/~xb/ab09/`.

[4] Miklos Ajtai. Generating hard instances of the short basis problem. In *Proc. of ICALP'99*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.

[5] A.Litvak, A. Pajor, M. Rudelson, and N. Tomczak-Jaegermann. Smallest singular value of random matrices and geometry of random polytopes. *Advances in Mathematics*, 195(2):491–523, 2005.

[6] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *Proc. of STACS'09*, pages 75–86, 2009.

[7] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simpler proof and improved concrete security for Waters' IBE scheme. In *Proc. of EUROCRYPT'09*, 2009.

[8] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Proc. of EUROCRYPT'04*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[9] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Proc. of CRYPTO'04*, volume 3152 of *LNCS*, pages 443–459, 2004.

[10] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. of Computing (SICOMP)*, 36(5):915–942, 2006. Journal version of [15] and [13].

[11] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Proc. of CRYPTO'01*, volume 2139 of *LNCS*, pages 213–29, 2001.

[12] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *Proc. of FOCS 2007*, pages 647–657, 2007.

[13] Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *Proceedings of CT-RSA 2005*, volume 3376 of *LNCS*. Springer-Verlag, 2005.

[14] Xavier Boyen. Lattice mixing and vanishing trapdoors : A framework for fully secure short signatures and more. In *Proc. of PKC 2010*, LNCS, 2010.

[15] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–22. Springer-Verlag, 2004.

[16] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *J. Cryptol.*, 20(3):265–294, 2007.

[17] David Cash, Dennis Hofheinz, and Eike Kiltz. How to delegate a lattice basis. Cryptology ePrint Archive, Report 2009/351, 2009. `http://eprint.iacr.org/`.

[18] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of Eurocrypt'10*, 2010.

[19] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA Conference*, pages 26–8, 2001.

[20] Ronald Cramer and Ivan Damgard. On the amortized complexity of zero-knowledge protocols. In *Proc. of CRYPTO '09*, 2009.

[21] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[22] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of STOC'08*, pages 197–206, 2008.

[23] Craig Gentry. Practical identity-based encryption without random oracles. In *Eurocrypt '06*, pages 445–464, 2006.

[24] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *Proc. of ASIACRYPT'02*, pages 548–566. Springer-Verlag, 2002.

[25] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[26] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In *Proc. of CRYPTO'08*, pages 21–38, 2008.

[27] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *Prof. of EUROCRYPT'02*, pages 466–481. Springer-Verlag, 2002.

[28] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *Proc. of ICALP'06*, 2006.

[29] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671. Kluwer Academic Publishers, 2002.

[30] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *Proc. of FOCS '04*, pages 372–381, 2004.

[31] Chris Peikert. Bonsai trees (or, arboriculture in lattice-based cryptography). Cryptology ePrint Archive, Report 2009/359, 2009. `http://eprint.iacr.org/`.

[32] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proc. of STOC '09*, pages 333–342. ACM, 2009.

[33] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proc. of TCC*, 2006.

[34] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC '05*, pages 84–93, 2005.

[35] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO'84*, pages 47–53, 1985.

[36] Victor Shoup. *A Computational Introduction to Number Theory and Algebra, second edition.* Cambridge University Press, 2008.

[37] D. Stehle, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public-key encryption based on ideal lattices. In *Proc. of Asiacrypt'09*, pages 617–635, 2009.

[38] Brent Waters. Efficient identity-based encryption without random oracles. In *Proc. of Eurocrypt 2005*, LNCS, 2005.

[39] Brent Waters. Dual key encryption: Realizing fully secure IBE and HIBE under simple assumption. In *Proc. of CRYPTO'09*, LNCS, 2009.