Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE

Abstract

We present a technique for delegating a short lattice basis that has the advantage of keeping the lattice dimension unchanged upon delegation. Building on this result, we construct two efficient hierarchical identity-based encryption or HIBE schemes (with and without random oracles) that generate shorter ciphertexts than earlier lattice-based constructions. We prove security from classic lattice hardness assumptions.

1 Introduction

Hierarchical identity based encryption (HIBE) is a public key encryption scheme where entities are arranged in a directed tree [HL02, GS02]. Each entity in the tree is provided with a secret key from its parent and can delegate this secret key to its children so that a child entity can decrypt messages intended for it, or for its children, but cannot decrypt messages intended for any other nodes in the tree. This delegation process is one-way: a child node cannot use its secret key to recover the key of its parent or its siblings. We define HIBE more precisely in the next section.

While the first HIBE constructions used bilinear maps [GS02, BB04, BW06, BBG05, GH09, Wat09], recent constructions are based on hard problems on lattices [CHK09, Pei09b]. The secret key in these lattice-based constructions is a "short" basis B of a certain integer lattice L. To delegate the key to a child the parent creates a new lattice L' derived from L and uses B to generate a random short basis for this lattice L'. Unfortunately, in all known constructions the dimension of the child lattice L' is larger than the dimension of the parent lattice L. As a result, private keys and ciphertexts become longer and longer as one descends into the hierarchy.

Our results. We first propose a new delegation mechanism that operates "in place", i.e., without increasing the dimension of the lattices involved. We then use this delegation mechanism to construct two HIBE systems where the lattices used have the same dimension for all nodes in the hierarchy. Consequently, private keys and ciphertexts in these systems are the same length for all nodes in the hierarchy and are much shorter than in previous lattice-based HIBE systems. Our first construction, in Section 5, provides full HIBE security in the random oracle model. Our second construction, in Appendix J, provides selective security in the standard model, namely without random oracles. We prove security of both constructions using the classic learning with errors (LWE) problem [Reg05].

To briefly explain our delegation technique, let L be a lattice in $\mathbb{Z}^m$ and let $B = \{b_1, \ldots, b_m\}$ be a short basis of L. Let R be a public non-singular matrix in $\mathbb{Z}^{m \times m}$. Observe that the set $B' := \{R\,b_1, \ldots, R\,b_m\}$ is a linearly independent set in the lattice $L' := RL$. If all entries of the matrix R are "small" scalars then the norm of the vectors in B' is not much larger than the norm of vectors in B. Moreover, using standard tools we can convert the set B' into a basis of L' and then "randomize" the basis without increasing the norm of the vectors by much. The end result is

a random short basis of L'. This idea suggests that by associating a public "low norm" matrix R to each child, the parent node can delegate its short basis B to a child by multiplying the vectors in B by the matrix R and randomizing the resulting basis. Note that since the dimension of L' is the same as the dimension of L this delegation does not increase dimensions.

One might wonder if the child can simply multiply its basis vectors B' by R^{-1} in an attempt to get back a short basis of the parent's lattice L. However, for most vectors $v \in \mathbb{Z}^m$ the norm of $R^{-1}v$ is much larger than the norm of v and this attempt fails to produce a short basis of L.

Proving security of this approach is quite technical. The key ingredient (Section 4.2) is a method that given a lattice L (for which no short basis is given) outputs a "low norm" matrix R, a delegated lattice L' and a short basis B' of L' such that $L' = LR$. In other words, if we are allowed to choose a low norm R then we can build a delegated lattice L' for which a short basis is known even though no short basis is given for L. This enables us to publish matrices R so that during the simulation certain private keys are known to the simulator while others are not. The key technical challenge is to show that these simulated matrices R are distributed as in the real system (Section 3).

As part of our security analysis we provide in Appendix F new bounds on the probability that vectors sampled from a discrete Gaussian distribution are linearly independent over $\mathbb{Z}$ and over $\mathbb{Z}_q$.

2 Preliminaries

Notation. Throughout the paper we say that a function $\epsilon : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ 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 $v_1, \ldots, v_n \in \mathbb{Z}^m$ are $\mathbb{Z}_q$-linearly independent if they are linearly independent when reduced modulo q.

2.1 Hierarchical IBE

We first review the definitions of IBE and HIBE. Recall that an Identity-Based Encryption system (IBE) consists of four algorithms [Sha85]: 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 [HL02, GS02], 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 $\ell - 1 > 0$. It outputs the private key $\mathsf{SK}_{\mathsf{Id}}$ for identity Id. For convenience, we sometimes refer to the master key as the private key at depth 0, given which 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 [BF01] defines the indistinguishability of ciphertexts under an adaptive chosen-ciphertext and chosen-identity attack ($\mathsf{IND\text{-}ID\text{-}CCA2}$). An adaptive chosen-identity attack means that the adversary is allowed to narrow in adaptively to the identity it wishes to target (i.e., the public key on which it will be challenged). A weaker notion of IBE security given by Canetti, Halevi, and Katz [CHK07] 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). We refer to such a system as a selective identity, chosen-ciphertext secure IBE.

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 and chosen-plaintext attack (IND-ID-CPA) and under a selective-identity chosen-plaintext attack (IND-sID-CPA) respectively.

The games used to define secure HIBE are listed in Appendix A.

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

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

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. Properties of the statistical distance that we will need are presented in Appendix B.

2.3 Integer Lattices

Definition 1. Let $B = \left[\, b_1 \,\middle|\, \ldots \,\middle|\, b_m \,\right] \in \mathbb{R}^{m\times m}$ be an $m\times m$ matrix whose columns are linearly independent vectors $b_1,\ldots,b_m\in\mathbb{R}^m$. The m-dimensional full-rank lattice Λ generated by B is the set,

$$\Lambda = \mathcal{L}(B) = \left\{\; y\in\mathbb{R}^m \quad \text{s.t.} \quad \exists s\in\mathbb{Z}^m \;,\;\; y = B\,s = \sum_{i=1}^{m} s_i\, b_i \;\right\}$$

Here, we are interested in integer lattices, i.e, when L is contained in $\mathbb{Z}^m$. We let $\det(\Lambda)$ denote the determinant of Λ and let Λ^* denote the dual lattice of Λ (see [Lov86]).

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

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

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 $\Lambda_q^\perp(A)$.
It is also easy to show that $\Lambda_q(A)^* = (1/q)\Lambda_q^\perp(A)$.

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 standard 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 [MG02] 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 ([MG02, 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*

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

Ajtai [Ajt99] 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 version of Ajtai's basis sampling algorithm from [AP09]. The following follows from Theorem 3.2 of [AP09] taking $\delta := 1/3$. The theorem produces a matrix A statistically close to uniform in $\mathbb{Z}_q^{n\times m}$ along with a short basis. Since m is so much larger than n, the matrix A is rank n with overwhelming probability. Hence, we can state the theorem as saying that A is statistically close to a uniform rank n matrix in $\mathbb{Z}_q^{n\times m}$.

Theorem 4. *Let $q \geq 3$ 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 rank n matrix in $\mathbb{Z}_q^{n\times m}$ and S is a basis for $\Lambda_q^\perp(A)$ satisfying*

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

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 $\mathsf{TrapGen}(q, n)$.

2.5 Discrete Gaussians

Definition 5. 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) discrete integral of $\rho_{\sigma,c}$ over L,

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

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

For notational convenience, $\rho_{\sigma,0}$ and $\mathcal{D}_{L,\sigma,0}$ are abbreviated as ρ_σ and $\mathcal{D}_{L,\sigma}$. When $\sigma = 1$ 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 $t \in \mathbb{Z}^m$. A few properties of discrete Gaussiants that we will need are presented in Appendix D.

2.6 Sampling from a discrete Gaussian

Gentry et al. [GPV08] construct the following algorithms for sampling from discrete Gaussians. Let $q \geq 2$ and let A be a matrix in $\mathbb{Z}_q^{n \times m}$. Let T_A be a basis for $\Lambda_q^\perp(A)$ and $\sigma \geq \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log m})$. Let $c \in \mathbb{R}^m$, and $u \in \mathbb{Z}_q^n$. Then:

- Algorithm $\mathsf{SampleGaussian}(A, T_A, \sigma, c)$ returns $x \in \Lambda_q^\perp(A)$ drawn from a distribution statistically close to $\mathcal{D}_{\Lambda,\sigma,c}$ which is a discrete Gaussian centered at c.
- Algorithm $\mathsf{SamplePre}(A, T_A, u, \sigma)$ returns $x \in \Lambda_q^u(A)$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^u(A),\sigma}$.

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

Randomizing a basis: Cash et al. [CHK09] and Peikert [Pei09b] show how to randomize a lattice basis.

$\mathsf{RandBasis}(S, \sigma)$:
On input a basis S of an m-dimensional lattice $\Lambda_q^\perp(A)$ and a gaussian parameter $\sigma \geq \|\widetilde{S}\| \cdot \omega(\sqrt{\log n})$, outputs a new basis S' of $\Lambda_q^\perp(A)$ such that

- with overwhelming probability $\|\widetilde{S'}\| \leq \sigma\sqrt{m}$, and
- up to a statistical distance, the distribution of S' does not depend on S. That is, the random variable $\mathsf{RandBasis}(S, \sigma)$ is statistically close to $\mathsf{RandBasis}(T, \sigma)$ for any other basis T of $\Lambda_q^\perp(A)$ satisfying $\|\widetilde{T}\| \leq \sigma/\omega(\sqrt{\log n})$.

We briefly recall how $\mathsf{RandBasis}$ works:

1. For $i = 1, \ldots, m$, let $v \leftarrow \mathsf{SampleGaussian}(A, S, \sigma, 0)$ 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 S' using lemma 3 (and using some canonical basis of $\Lambda_q^\perp(A)$).
3. Output S'.

The analysis of $\mathsf{RandBasis}$ in [CHK09, Pei09b] uses [Reg05, Corollary 3.16] which shows that a linearly independent set is produced in Step (1) w.h.p. after m^2 samples from $\mathsf{SampleGaussian}(A, S, \sigma, 0)$. In Appendix F.2 we show that only $2m$ samples are needed in expectation.

2.7 Hardness assumption

Security of all our constructions reduces to the LWE (learning with errors) problem, a classic hard problem on lattices defined by Regev [Reg05]. Due to space constraints we state the LWE problem in Appendix E.

3 Gaussian Sampling on a Random Sublattice

Our security proofs require that Gaussian sampling from a *random* sublattice of $\mathbb{Z}^m$ produces approximately the same distribution as Gaussian sampling from $\mathbb{Z}^m$ directly. More precisely, define the following two distributions on $\mathbb{Z}^{m \times m}$.

For a positive integer m and some $\sigma > 2$ define $\mathbf{DIST}_0(\sigma, m)$ as:

1. for $i = 1, \ldots, m$ sample independent $r_i \xleftarrow{R} \mathcal{D}_{\mathbb{Z}^m,\sigma}$; set $R_0 := [\, r_1 \mid \ldots \mid r_m \,] \in \mathbb{Z}^{m \times m}$;
2. if R_0 is full rank over $\mathbb{Z}_q$ output R_0; otherwise go back to step (1).

Note that R_0 in step (1) is sampled from $\left(\mathcal{D}_{\mathbb{Z}^m,\sigma}\right)^m$. Theorem 30 in the appendix shows that for $\sigma > \omega(\sqrt{\log m})$ the expected number of iterations to generate R_0 is less than two.

Let A be a rank n matrix in $\mathbb{Z}_q^{n \times m}$ and let $a_1, \ldots, a_m \in \mathbb{Z}_q^n$ be the m columns of A. Define $\mathbf{DIST}_1(A, \sigma, m, q)$ as:

1. sample a uniform random rank n matrix B in $\mathbb{Z}_q^{n \times m}$;
2. for $i = 1, \ldots, m$ sample independent $r_i \xleftarrow{R} \mathcal{D}_{\Lambda_q^{a_i}(B),\sigma}$ (so that $Br_i = a_i$ modulo q) ; set $R_1 := [\, r_1 \mid \ldots \mid r_m \,] \in \mathbb{Z}^{m \times m}$;
3. if R_1 is full rank over $\mathbb{Z}_q$ output R_1; otherwise go back to step (2).

R_1 in step (2) is sampled from $\mathcal{D}_{\Lambda_q^{a_1}(B),\sigma} \times \cdots \times \mathcal{D}_{\Lambda_q^{a_m}(B),\sigma}$ and hence $B\,R_1 = A \bmod q$.

In this section we analyze these distributions without worrying about the complexity of sampling from them. We give efficient sampling algorithms in the next section. The main result of this section, captured in the following theorem, shows that $\mathrm{DIST}_0(\sigma, m)$ and $\mathrm{DIST}_1(A, \sigma, m, q)$ are statistically close for most matrices A. The proof is given in Appendix G.

Theorem 6. *Let q be a prime and let $m > 2n \log q$. Then for all σ in the range $4 \le \sigma < q/\ln m$ and all rank n matrices $A \in \mathbb{Z}_q^{n \times m}$, except for at most a q^{-n} fraction of such A, the distributions $DIST_0(\sigma, m)$ and $DIST_1(A, \sigma, m, q)$ are statistically close.*

The proof of Theorem 6 proceeds in several steps. Let R_0 be sampled from $\mathrm{DIST}_0(\sigma, m)$ and recall that $(\mathbb{Z}_q^{n \times m})^*$ denotes the set of rank n matrices in $\mathbb{Z}_q^{n \times m}$. First, we show in Lemma 35 that the distance between $\mathrm{DIST}_0(\sigma, m)$ and $\mathrm{DIST}_1(A, \sigma, m, q)$ is equal to the distance of $AR_0^{-1} \bmod q$ from the uniform distribution over $(\mathbb{Z}_q^{n \times m})^*$. Next, let B be a uniform variable over $(\mathbb{Z}_q^{n \times m})^*$. We show that $(B,\ BR_0 \bmod q)$ is close to uniform over $[(\mathbb{Z}_q^{n \times m})^*]^2$. To do so recall that the set of functions $H_B : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ defined by $H_B(v) = B\,v$ is a universal family of hash functions. We then develop a strong version of the left over hash lemma (Lemma 17 and Corollary 19) to show that $(B,\ BR_0 \bmod q)$ is close to uniform. The reason we need a strong version of the left over hash lemma is that requiring R_0 to be full rank over $\mathbb{Z}_q$ introduces a dependency between the columns of R_0. This dependency prevents the standard left over hash (e.g. Theorem 8.38 in [Sho08]) from being used to extract the entropy from R_0. Our version can extract entropy from dependent samples. Finally, since $(B,\ BR_0 \bmod q)$ is close to uniform, we argue using properties of statistical distance (Lemma 13) that for most A the random variable $AR_0^{-1} \bmod q$ is close to uniform. This is what we needed to prove that $\mathrm{DIST}_0(\sigma, m)$ and $\mathrm{DIST}_1(A, \sigma, m, q)$ are statistically close. The complete details are given in Appendix G.

4 Basis Delegation Without Dimension Increase

Let A be a matrix in $\mathbb{Z}_q^{n \times m}$ and let T_A be a "short" basis of $\Lambda_q^\perp(A)$, both given. We wish to "delegate" the basis T_A in the following sense: we want to deterministically generate a matrix B

from A and a random basis T_B for $\Lambda_q^\perp(B)$ such that from A, B and T_B it is difficult to recover any short basis (such as T_A) for $\Lambda_q^\perp(A)$.

Basis delegation was studied by Cash et al [CHK09] and Peikert [Pei09b]. In all those delegation mechanisms, the dimension B was larger than the dimension of A. In the resulting HIBE systems ciphertext and private key sizes increase as the hierarchy deepens.

Here we consider a simple delegation mechanism that does not increase the dimension. To do so we use a public matrix R in $\mathbb{Z}^{m\times m}$ where the columns of R have "low" norm. We require that R be invertible mod q. Now, define $B := AR^{-1}$ and observe that B has the same dimension as A. We show how to build a "short" basis of $\Lambda_q^\perp(B)$ from which it is difficult to recover a short basis of A. In the next section we use this to build new HIBE systems.

We begin by defining distributions on matrices whose columns are low norm vectors. We then define the basis delegation mechanism.

Distributions on low norm matrices. Our construction makes use of invertible matrices R in $\mathbb{Z}^{m\times m}$ where all the columns of R are "small" or "low norm". We say that a matrix R in $\mathbb{Z}^{m\times m}$ is $\mathbb{Z}_q$**-invertible** if R mod q is invertible as a matrix in $\mathbb{Z}_q^{m\times m}$.

Definition 7. Define $\sigma_{\mathrm{R}} := \sigma_{\mathrm{TG}}\,\omega(\sqrt{\log m}) = \sqrt{n\log q}\cdot\omega(\sqrt{\log m})$. We let $\mathcal{D}_{m\times m}$ denote the distribution $\mathrm{DIST}_0(\sigma_{\mathrm{R}}, m)$ on $\mathbb{Z}_q$-invertible matrices in $\mathbb{Z}^{m\times m}$ from Section 3.

Algorithm SampleR(1^m)**.** The following simple algorithm samples matrices in $\mathbb{Z}^{m\times m}$ from a distribution that is statistically close to $\mathcal{D}_{m\times m}$.

1. Let T be the canonical basis of the lattice $\mathbb{Z}^m$.
2. For $i = 1, \ldots, m$ do $r_i \xleftarrow{R} \mathsf{SampleGaussian}(\mathbb{Z}^m, T, \sigma_{\mathrm{R}}, 0)$.
3. If R is $\mathbb{Z}_q$-invertible, output R; otherwise repeat step 2.

Theorem 30 shows that step 2 will need to be repeated fewer than two times in expectation.

4.1 Basis delegation: algorithm BasisDel(A, R, T_A, σ)

We now describe a simple basis delegation algorithm that does not increase the dimension of the underlying matrices.

Inputs: a rank n matrix A in $\mathbb{Z}_q^{n\times m}$,
a $\mathbb{Z}_q$-invertible matrix R in $\mathbb{Z}^{m\times m}$ sampled from $\mathcal{D}_{m\times m}$ (or a product of such),
a basis T_A of $\Lambda_q^\perp(A)$,
and a parameter $\sigma \in \mathbb{R}_{>0}$. (1)

Output: Let $B := AR^{-1}$ in $\mathbb{Z}_q^{n\times m}$. The algorithm outputs a basis T_B of $\Lambda_q^\perp(B)$.

Algorithm BasisDel(A, R, T_A, σ) works as follows:

1. Let $T_A = \{a_1, \ldots, a_m\} \subseteq \mathbb{Z}^m$. Calculate $T'_B := \{Ra_1, \ldots, Ra_m\} \subseteq \mathbb{Z}^m$.
   Observe that T'_B is a set of independent vectors in $\Lambda_q^\perp(B)$.

2. Use Lemma 3 to convert T'_B into a basis T''_B of $\Lambda_q^\perp(B)$.
   The algorithm in the lemma takes as input T'_B and an arbitrary basis of $\Lambda_q^\perp(B)$ and outputs a basis T''_B whose Gram-Schmidt norm is no more than that of T'_B.

3. Call $\mathsf{RandBasis}(T''_B, \sigma)$ and output the resulting basis T_B of $\Lambda_q^\perp(B)$.

The following theorem shows that $\mathsf{BasisDel}$ produces a random basis of $\Lambda_q^\perp(B)$ whose Gram-Schmidt norm is bounded as a function of $\|\widetilde{T_A}\|$. The proof is given in Appendix H.

Theorem 8. *Using the notation in (1), suppose R is sampled from $\mathcal{D}_{m\times m}$ and σ satisfies*

$$\sigma > \|\widetilde{T_A}\| \cdot \sigma_{\mathrm{R}} \sqrt{m}\, \omega(\log^{3/2} m) \ .$$

Let T_B be the basis of $\Lambda_q^\perp(AR^{-1})$ output by $\mathsf{BasisDel}$.
Then T_B is distributed statistically close to the distribution $\mathsf{RandBasis}(T, \sigma)$ where T is an arbitrary basis of $\Lambda_q^\perp(AR^{-1})$ satisfying $\|\widetilde{T}\| < \sigma/\omega(\sqrt{\log m})$. If R is a product of ℓ matrices sampled from $\mathcal{D}_{m\times m}$ then the bound on σ degrades to $\sigma > \|\widetilde{T_A}\| \cdot \left(\sigma_{\mathrm{R}} \sqrt{m}\, \omega(\log^{1/2} m)\right)^\ell \cdot \omega(\log m)$.

Note that for the smallest possible σ in Theorem 8 we obtain that with overwhelming probability

$$\|\widetilde{T_B}\| \ / \ \|\widetilde{T_A}\| \le \sigma_{\mathrm{R}} m\, \omega(\log^{3/2} m) = m^{3/2}\, \omega(\log^2 m) \ .$$

This quantity is the minimum degradation in basis quality as we delegate from level to level in the HIBE hierarchy.

4.2 The main simulation tool: algorithm $\mathsf{SampleRwithBasis}(A)$

All our proofs of security make heavy use of an algorithm $\mathsf{SampleRwithBasis}$ that given a matrix A in $\mathbb{Z}_q^{n\times m}$ as input generates a "low-norm" matrix R (i.e., a matrix sampled from $\mathcal{D}_{m\times m}$) along with a short basis for $\Lambda_q^\perp(AR^{-1})$.

Algorithm $\mathsf{SampleRwithBasis}(A)$. Let $a_1, \ldots, a_m \in \mathbb{Z}_q^n$ be the m columns of A.

1. Run $\mathsf{TrapGen}(q, n)$ to generate a uniform matrix $B \in \mathbb{Z}_q^{n\times m}$ and a basis T_B of $\Lambda_q^\perp(B)$ such that $\|\widetilde{T_B}\| \le \sigma_{\mathrm{TG}} = \sigma_{\mathrm{R}}/\omega(\sqrt{\log m})$.

2. for $i = 1, \ldots, m$ do:
   (2a) sample $r_i \in \mathbb{Z}^m$ as the output of $\mathsf{SamplePre}(B, T_B, a_i, \sigma_{\mathrm{R}})$,
   then $Br_i = a_i \bmod q$ and the distribution of r_i is statistically close to $\mathcal{D}_{\Lambda_q^{a_i}(B), \sigma_{\mathrm{R}}}$.
   (2b) repeat step (2a) until r_i is $\mathbb{Z}_q$ linearly independent of $r_1, \ldots, r_{i-1}$.

3. Let $R \in \mathbb{Z}^{m\times m}$ be the matrix whose columns are $r_1, \ldots, r_m$.
   Then R has rank m over $\mathbb{Z}_q$. Output R and T_B.

By construction $BR = A \bmod q$ and therefore $B = AR^{-1} \bmod q$. Hence, the basis T_B is a short basis of $\Lambda_q^\perp(AR^{-1})$. It remains to show that R is sampled from a distribution close to $\mathcal{D}_{m\times m}$.

Theorem 9. *Let $m > 2n \log q$ and $q > 2$. For all but at most a q^{-n} fraction of rank n matrices A in $\mathbb{Z}_q^{n\times m}$ algorithm $\mathsf{SampleRwithBasis}(A)$ outputs a matrix R in $\mathbb{Z}^{m\times m}$ sampled from a distribution statistically close to $\mathcal{D}_{m\times m}$. The generated basis T_B of $\Lambda_q^\perp(AR^{-1})$ satisfies $\|\widetilde{T_B}\| \le \sigma_{\mathrm{R}}/\omega(\sqrt{\log m})$ with overwhelming probability.*

Proof. It suffices to argue that R is sampled from a distribution statistically close to $\mathcal{D}_{m\times m}$. This follows directly from Theorem 6. To see why observe that the matrix R is sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{a_1}(B),\sigma_{\mathrm{R}}} \times \cdots \times \mathcal{D}_{\Lambda_q^{a_m}(B),\sigma_{\mathrm{R}}}$ conditioned on R being full rank over $\mathbb{Z}_q$. But this is precisely the distribution $\mathrm{DIST}_1(A, \sigma_{\mathrm{R}}, m, q)$ from Section 3. By Theorem 6, for most rank n matrices A, this distribution is statistically close to $\mathrm{DIST}_0(\sigma_{\mathrm{R}}, m)$ which is itself the distribution $\mathcal{D}_{m\times m}$. □

5 Adaptively Secure HIBE in the Random-Oracle Model

In this section, we build an HIBE of depth d, secure in the random oracle model. We present a standard-model HIBE in Appendix J.

To encrypt a message m for identity Id, the encryptor builds a matrix F_{Id} and encrypts m using the dual Regev public key system (described in [GPV08, sec. 7]) using F_{Id} as the public key. The matrix F_{Id} is built by multiplying a fixed matrix A, specified in the public parameters, by ℓ "low norm" square matrices generated by a random oracle H described in (2) below.

At level ℓ, let $\mathsf{Id} = (\mathsf{Id}_1, \mathsf{Id}_2, \ldots, \mathsf{Id}_\ell) \in (\{0,1\}^*)^\ell$, where $\ell \in [d]$. We assume the availability of a hash function

$$H \;:\; \mathbb{Z}_{\geq 0} \times \{0,1\}^* \;\rightarrow\; \mathbb{Z}_q^{m\times m} \quad:\quad (i, \mathsf{Id}_i) \;\mapsto\; H(i, \mathsf{Id}_i) \sim \mathcal{D}_{m\times m} \tag{2}$$

where $H(i, \mathsf{Id}_i) \sim \mathcal{D}_{m\times m}$ is over the choice of H, which is modeled as a random oracle (the distribution $\mathcal{D}_{m\times m}$ is as in Definition 7). H can be constructed explicitly from any "standard" random function $h : \{0,1\}^* \rightarrow \{0,1\}$ by using h as a coin generator for the sampling process in Algorithm $\mathsf{SampleR}(1^m)$.

5.1 Parameters

The parameters n, m and q are fixed across the levels of the hierarchy. In addition, we have level-dependent parameters, which are σ_ℓ (gaussian parameter) and α_ℓ (noise parameter). As we delegate from level to level down the hierarchy, the length of the delegated lattice basis increases. As a result, we will need to increase σ_ℓ so that $\sigma_\ell > \sigma_{\ell-1}\, m^{3/2}\omega(\log^2 m)$ and decrease α_ℓ.

To meet these requirements, we set:

$$q := \tilde{\omega}((nd)^{2d}) \qquad \text{and} \qquad m := \tilde{\omega}(nd)$$

The value of q is comparable to those in the HIBE schemes of [CHK09, Pei09b] which also need field size q to be exponential in the maximal depth d. However, the width of our lattice m only depends linearly on d where as in previous constructions the width had a quadratic dependence on d. Moreover, since $\sigma_1 = \sigma_{\mathrm{TG}}$ we obtain

$$\sigma_\ell \geq \tilde{\omega}\big((nd)^{2d}\big) \qquad \text{and} \qquad \alpha_\ell \leq \frac{1}{\tilde{\omega}\big((nd)^{2d}\big)\sqrt{m}}$$

5.2 Construction

The scheme works as follows:

Setup$(1^n, 1^d)$ On input a security parameter n and maximum depth d:

1. Set the parameters of the system $q, m, \sigma_\ell, \alpha_\ell$ as described above.
2. Invoke $\mathsf{TrapGen}(q, n)$ to generate a uniformly random matrix $A \in \mathbb{Z}_q^{n \times m}$ along with a basis $T_A = [a_1 | \ldots | a_m] \in \mathbb{Z}^{m \times m}$ generating the lattice $\Lambda_q^\perp(A)$.
3. Generate a uniformly random vector $u_0 \in \mathbb{Z}_q^n$.
4. Output the public parameters PP and master key MK given by,

$$\mathsf{PP} = \left(A,\ u_0 \right) \qquad\qquad \mathsf{MK} = \left(T_A \right)$$

Extract$(\mathsf{MK}, \mathsf{Id})$ On input a master key MK and an identity $\mathsf{Id} = (\mathsf{Id}_1, \ldots, \mathsf{Id}_\ell)$ of length $|\mathsf{Id}| = \ell$:

1. Compute the matrix product $R_{\mathsf{Id}} \leftarrow H(\ell, \mathsf{Id}_\ell) \ldots H(2, \mathsf{Id}_2)\, H(1, \mathsf{Id}_1)$ in $\mathbb{Z}^{m \times m}$. Define $F_{\mathsf{Id}} = A\, R_{\mathsf{Id}}^{-1} \bmod q$, $F_{\mathsf{Id}} \in \mathbb{Z}_q^{m \times m}$.
2. Construct a randomized basis for $\Lambda_q^\perp(F_{\mathsf{Id}})$ by running $S \leftarrow \mathsf{BasisDel}(A, R_{\mathsf{Id}}, T_A, \sigma_\ell)$.
3. Output the requested identity-based private key $\mathsf{SK}_{\mathsf{Id}} = S$.

Derive$(\mathsf{PP}, \mathsf{SK}_{\mathsf{Id}_{|\ell-1}}, \mathsf{Id})$**:** On input public parameters PP, a secret key $\mathsf{SK}_{\mathsf{Id}_{|\ell-1}}$ corresponding to a "parent" identity $\mathsf{Id}_{|\ell-1} = (\mathsf{Id}_1, \ldots, \mathsf{Id}_{\ell-1})$, and a "child" identity $\mathsf{Id} = (\mathsf{Id}_1, \ldots, \mathsf{Id}_{\ell-1}, \mathsf{Id}_\ell)$:

1. Let $R_{\mathsf{Id}_{|\ell-1}} = H(\ell-1, \mathsf{Id}_{|\ell-1}) \ldots H(2, \mathsf{Id}_2)\, H(1, \mathsf{Id}_1)$. Set $F_{\mathsf{Id}_{|\ell-1}} \leftarrow A\, R_{\mathsf{Id}_{|\ell-1}}^{-1} \in \mathbb{Z}_q^{m \times m}$. Recall that $\mathsf{SK}_{\mathsf{Id}_{|\ell-1}}$ is a short basis for $\Lambda_q^\perp(F_{\mathsf{Id}_{|\ell-1}})$. Let $F_{\mathsf{Id}} = F_{\mathsf{Id}_{|\ell-1}}\, H(\ell, \mathsf{Id}_\ell)^{-1} \in \mathbb{Z}_q^{m \times m}$.
2. Evaluate $S' \leftarrow \mathsf{BasisDel}(F_{\mathsf{Id}_{|\ell-1}}, H(\ell, \mathsf{Id}_\ell), \mathsf{SK}_{\mathsf{Id}_{|\ell-1}}, \sigma_\ell)$ to obtain a short random basis for $\Lambda_q^\perp(F_{\mathsf{Id}})$.
3. Output the delegated private key $\mathsf{SK}_{\mathsf{Id}} = S'$.

Encrypt$(\mathsf{PP}, \mathsf{Id}, b)$**:** On input public parameters PP, a recipient identity Id of depth $|\mathsf{Id}| = \ell$, and a message bit $b \in \{0, 1\}$:

1. Compute $R_{\mathsf{Id}} \leftarrow H(\ell, \mathsf{Id}_\ell) \ldots H(2, \mathsf{Id}_2)\, H(1, \mathsf{Id}_1)$.
2. Compute the encryption matrix $F_{\mathsf{Id}} \leftarrow A\, R_{\mathsf{Id}}^{-1} \bmod q$. Then F_{Id} is in $\mathbb{Z}_q^{n \times m}$.
3. Now encrypt the message using Regev's dual public key encryption (as defined in [GPV08, sec. 7]) using F_{Id} as the public key. To do so,
   (a) Pick a uniformly random vector $s \xleftarrow{R} \mathbb{Z}_q^n$.
   (b) Choose noise vectors $x \xleftarrow{\overline{\Psi}_{\alpha_\ell}} \mathbb{Z}_q$ and $y \xleftarrow{\overline{\Psi}_{\alpha_\ell}^m} \in \mathbb{Z}_q^m$. ($\overline{\Psi}_\alpha$ is defined in Appendix E)
   (c) Output the ciphertext,

$$\mathsf{CT} \ = \ \left(\ c_0 = u_0^T\, s + x + b\, \lfloor \frac{q}{2} \rfloor \ , \ \ c_1 = F_{\mathsf{Id}}^T\, s + y \ \right) \ \in \ \mathbb{Z}_q \times \mathbb{Z}_q^m$$

Decrypt$(\mathsf{PP}, \mathsf{SK}_{\mathsf{Id}}, \mathsf{CT})$**:** On input public parameters PP, a private key $\mathsf{SK}_{\mathsf{Id}}$ for an identity Id of length $|\mathsf{Id}| = \ell$, and a ciphertext CT:

1. Set $d_{\mathsf{Id}} \leftarrow \mathsf{SamplePre}(F_{\mathsf{Id}}, \mathsf{SK}_{\mathsf{Id}}, u_0, \sigma_\ell)$. Note that $F_{\mathsf{Id}}\, d_{\mathsf{Id}} = u_0$.
2. Compute $w = c_0 - d_{\mathsf{Id}}^T\, c_1 \in \mathbb{Z}_q$.

3. Compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in $[q] \subset \mathbb{Z}$:
   if they are close, i.e., if $\left| w - \lfloor \frac{q}{2} \rfloor \right| < \lfloor \frac{q}{4} \rfloor$ in $\mathbb{Z}$, then output 1; otherwise output 0.

In Appendix I we show that the scheme is consistent, namely that valid ciphertexts are correctly decrypted w.h.p.

5.3 Security

Theorem 10. *If, in the random-oracle model, there exists a PPT adversary $\mathcal{A}$ with* IND-ID-CPA *advantage $\epsilon = \Omega(\lambda^k)$ for some $k > 0$ against the adaptive HIBE scheme above, then there exists a PPT algorithm $\mathcal{B}$ that decides the LWE problem with advantage $\epsilon' = \Omega(\lambda^k / Q_H^d)$, where Q_H is the number of queries made by $\mathcal{A}$ to the random oracle H and d is the hierarchy depth.*

Proof. Let $\mathcal{A}$ be an IND-ID-CPA attacker. We will show that a non-negligible advantage ϵ in the IND-ID-CPA game can be used to decide the LWE problem with advantage ϵ / Q_H^d. This will prove that under the LWE assumption no polynomial-time attacker can have non-negligible advantage in the IND-ID-CPA game. Recall that LWE is about recognizing an oracle $\mathcal{O}$ (see Appendix E).

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

As the number of oracle calls is known *a priori*, the samples can be supplied non-interactively at the beginning, e.g., here in the form of an instance with $(m+1)(n+1)$ elements of $\mathbb{Z}_q$.

Setup. $\mathcal{B}$ prepares a simulated attack environment for $\mathcal{A}$ as follows.

1. For each $i = 1, \ldots, d$:
   (a) Select a uniform random integer $Q_i^* \in [Q_H]$, where Q_H is the maximum number of random-oracle queries to H that $\mathcal{A}$ can make.
   (b) Sample a random matrix $R_i^* \sim \mathcal{D}_{m \times m}$ by using $R_i^* \leftarrow \mathsf{SampleR}(1^m)$; save R_i^* for future use.
2. 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 $i = 1, \ldots, m$.
   Set $A \leftarrow A_0 R_d^* \cdots R_1^*$.
3. Publish the public parameters $\mathsf{PP} = \big(A, u_0\big)$.

Random-oracle hash queries. $\mathcal{A}$ may query the random oracle H on any pair (i, Id_i) of its choice, adaptively, and at any time. $\mathcal{B}$ answers the Q-th such query as follows. (We assume w.l.o.g. that the queries are unique; otherwise the simulator simply returns the same output on the same input without incrementing the query counter Q.)

If $Q = Q_i^*$, define $H(i, \mathsf{Id}_i) \leftarrow R_i^*$, return $H(i, \mathsf{Id}_i)$, and stop.

Otherwise, if $Q \neq Q_i^*$:

1. Define the constant $A_i = A \cdot \big(R_{i-1}^* \cdots R_2^* R_1^*\big)^{-1} \in \mathbb{Z}_q^{m \times m}$ (letting $A_i = A$ for $i = 1$).
2. Run $\mathsf{SampleRwithBasis}(A_i)$ to obtain a random $R \sim \mathcal{D}_{m \times m}$ and a short basis T_B for $B = A_i R^{-1} \bmod q$. Then R is a random low-norm matrix in $\mathbb{Z}^{m \times m}$ that maps the constant matrix A_i to the matrix $B = A_i R^{-1} \bmod q$ for which T_B is a short basis.

3. Save the tuple $(i, \mathsf{Id}_i, R, B, T_B)$ for future use, and return $H(i, \mathsf{Id}_i) \leftarrow R$.

Queries 1. $\mathcal{A}$ makes interactive key-extraction queries on arbitrary identities Id, chosen adaptively. $\mathcal{B}$ answers a query on $\mathsf{Id} = \mathsf{Id}_1\mathsf{Id}_2 \dots \mathsf{Id}_k$ of length $|\mathsf{Id}| = k \in [d]$ as follows.

1. Construct $R_{\mathsf{Id}} \leftarrow H(k, \mathsf{Id}_k) \cdots H(2, \mathsf{Id}_2) \cdot H(1, \mathsf{Id}_1)$, querying the oracle H as needed.
2. Let $j \in [k]$ be the shallowest level at which the components of Id and Id^* differ (letting $j = 1$ if Id^* has not been defined yet). Retrieve the saved tuple $(j, \mathsf{Id}_j, R, B, T_B)$ from the hash oracle query history (w.l.o.g., we can assume that an extraction query on Id is preceded by a hash query on each component of Id). Then by construction $A_j = B\,R \bmod q$ or equivalently $B = A \cdot (R_1^*)^{-1} \cdots (R_{j-1}^*)^{-1} \cdot H(j, \mathsf{Id}_j)^{-1} \bmod q$.

   Notice that T_B is a short basis for $\Lambda_q^\perp(B)$, and that B is exactly the encryption matrix $F_{\mathsf{Id}_{|j}}$ (as defined in the Encrypt algorithm) for the ancestor identity $\mathsf{Id}_{|j} = \mathsf{Id}_1\mathsf{Id}_2 \dots \mathsf{Id}_j$ obtained by "truncating" Id to the first j levels. Hence T_B is a trapdoor for $\Lambda_q^\perp(F_{\mathsf{Id}_{|j}})$.
3. Starting from $F_{\mathsf{Id}_{|j}} = B$ and using its short basis T_B, construct a short basis T for $F_{\mathsf{Id}} = A \cdot R_{\mathsf{Id}}^{-1} \bmod q = B \cdot H(j+1, \mathsf{Id}_{j+1})^{-1} \cdot \cdots \cdot H(k, \mathsf{Id}_k)^{-1}$ by invoking the delegation algorithm $\mathsf{BasisDel}(B, R_{k,\mathsf{Id}_k} \cdots R_{j+1,\mathsf{Id}_{j+1}}, T_B, \sigma_k)$.
4. Return T as private decryption key $\mathsf{SK}_{\mathsf{Id}}$ corresponding to the queried identity Id.

Challenge. $\mathcal{A}$ announces to $\mathcal{B}$ the identity Id^* on which it wishes to be challenged. Say $|\mathsf{Id}^*| = \ell$. We require that Id^* not be equal to, or a descendant of, any identity Id for which a private key has been or will be requested in any preceding and subsequent key extraction query. $\mathcal{A}$ also submits a message bit $b^* \in \{0, 1\}$ to be encrypted.

If the challenge identity Id^* contains one or more component(s) Id_i^* such that $H(i, \mathsf{Id}_i^*) \neq R_i^*$, then the simulator must abort. (Indeed, when this is the case, $\mathcal{B}$ is able extract a private key for Id^* and thus answer by itself the challenge that it intended to ask.)

If o.t.o.h. $\mathsf{Id}^* = \mathsf{Id}_1^* \dots \mathsf{Id}_\ell^*$ is such that $H(i, \mathsf{Id}_i^*) = R_i^*$ for all $i \in [\ell]$, then $\mathcal{B}$ proceeds as follows:

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

Queries 2. $\mathcal{A}$ makes more extraction queries, answered by $\mathcal{B}$ in the same manner as before.

Guess. After being allowed to make additional queries, $\mathcal{A}$ guesses whether CT^* was an encryption of b^* for Id^*. Upon receiving $\mathcal{A}$'s guess, $\mathcal{B}$ end the simulation and outputs its answer to LWE:

- If $\mathcal{A}$ guesses "good", $\mathcal{B}$ answers "pseudo-random".
- If $\mathcal{A}$ guesses "bad", $\mathcal{B}$ answers "random".

Note that by Theorem 8, the distribution of the public parameters and private key responses is indistinguishable from that in the main system. The simulator can proceed without aborting in the challenge phase with probability $\Pr[\neg\text{abort}] \geq Q_H^{-\ell} \geq Q_H^{-d}$ in the worst case. By a standard argument, if $\mathcal{A}$ has advantage $\epsilon \geq 0$ in the above game, then, in the worst case (when $\ell = d$), $\mathcal{B}$ has advantage $Q_H^{-d} \cdot \epsilon/2$ in deciding the LWE problem instance. □

References

[Ajt99] Miklos Ajtai. Generating hard instances of the short basis problem. In Jir Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *ICALP*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.

[AP09] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *STACS*, pages 75–86, 2009.

[Ban95] Wojciech Banaszczyk. Inequalites for convex bodies and polar reciprocal lattices in r^n. *Discrete and Computational Geometry*, 13:217–231, 1995.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Advances in Cryptology (EUROCRYPT 2004)*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*. Springer-Verlag, 2005.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–29. Springer-Verlag, 2001.

[BW06] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Advances in Cryptology—CRYPTO 2006*, volume 4117 of *LNCS*, pages 290–307. Springer-Verlag, 2006.

[CHK07] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *J. Cryptol.*, 20(3):265–294, 2007.

[CHK09] David Cash, Dennis Hofheinz, and Eike Kiltz. How to delegate a lattice basis. Cryptology ePrint Archive, Report 2009/351, 2009. `http://eprint.iacr.org/`.

[DORS08] 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.

[GH09] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In *Theory of Cryptography—TCC 2009*, volume 5444 of *LNCS*, pages 437–56. Springer-Verlag, 2009.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 197–206. ACM, 2008.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT '02: Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, pages 548–566, London, UK, 2002. Springer-Verlag.

[HILL99] 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.

[HL02] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT '02: Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pages 466–481, London, UK, 2002. Springer-Verlag.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal Of the American Statistical Association*, 58(301):13–30, 1963.

[Lov86] L. Lovasz. *An Algorthmic Theory of Numbers, Graphs, and Convexity.* SIAM, 1986.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 372–381, Washington, DC, USA, 2004. IEEE Computer Society.

[Pei09a] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC '09*, 2009.

[Pei09b] Chris Peikert. Bonsai trees (or, arboriculture in lattice-based cryptography). Cryptology ePrint Archive, Report 2009/359, 2009. `http://eprint.iacr.org/`.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 84–93, New York, NY, USA, 2005. ACM.

[Sha85] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[Sho08] Victor Shoup. *A Computational Introduction to Number Theory and Algebra, second edition.* Cambridge University Press, 2008.

[Wat09] Brent Waters. Dual key encryption: Realizing fully secure IBE and HIBE under simple assumption. In *Advances in Cryptology—CRYPTO 2009*, 2009.

A HIBE Security Game

For a security parameter λ, we let $\mathcal{M}_\lambda$ denote the message space and let $\mathcal{C}_\lambda$ denote the ciphertext space. We define anonymous IBE and HIBE semantic security under a selective-identity attack (for a hierarchy of maximum depth d) using the following game between a challenger and an adversary. The game captures a property called *ciphertext privacy* 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.

Init: The adversary is given as input the maximum depth of the hierarchy d from the challenger, and outputs an identity $\mathsf{Id}^* = (\mathsf{I}_1^*, \ldots, \mathsf{I}_k^*), k \leq d$ where it wishes to be challenged.

Setup: The challenger runs the Setup algorithm giving it the maximum depth d as input (where $d = 1$ for IBE) and the security parameter λ. It 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 corresponding to 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 $\mathsf{Id}_i \neq \mathsf{Id}^*$ and Id_i is not a prefix of Id^*. The challenger responds as in Phase 1.

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

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{CP\text{-}sID\text{-}CPA}$ adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking an HIBE scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Derive}, \mathsf{Encrypt}, \mathsf{Decrypt})$, or an IBE scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$, 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 11. We say that an IBE or a depth d HIBE system $\mathcal{E}$ is selective-identity, ciphertext private if for all $\mathsf{IND\text{-}sID\text{-}CPA}$ 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 $\mathsf{IND\text{-}sID\text{-}CPA}$ 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 11, and is denoted $\mathsf{IND\text{-}ID\text{-}CPA}$.

B Statistical distance

Properties of the statistical distance are presented in, e.g. Shoup [Sho08, Chapter 8.8]. We state a few properties we will need here. Recall that if X and W are random variables taking values in some sets Ω_{x} and Ω_{w} respectively, then for all $w \in \Omega_{\mathrm{w}}$ the random variable $R(w) := (X|W = w)$ over Ω_{x} is defined as $\Pr[R(w) = x] := \Pr[X = x \mid W = w]$ for all $x \in \Omega_{\mathrm{x}}$. If X and Y take values in Ω then[1]

$$E_W\big[\Delta(X|W\ ;\ Y|W)\big] := \sum_{w\in\Omega_{\mathrm{w}}} \Pr[W = w]\ \Delta\big(\ (X|W = w)\ ;\ (Y|W = w)\ \big) \tag{3}$$

Lemma 12. *For random variables X, Y, Z taking values in a finite set Ω and a random variable W taking values in a finite set Ω_{w} we have:*

1. $\Delta(X;Y) \le \Delta(X;Z) + \Delta(Z;Y)$.
2. *For all functions $f : \Omega \to \Omega'$ we have $\Delta(f(X);f(Y)) \le \Delta(X;Y)$.*
3. *If W is independent of X and Y then $\Delta(X;Y) = \Delta\big(X, W\ ;\ Y, W\big)$.*
4. $\Delta(X, W\ ;\ Y, W) = E_W[\Delta(X|W\ ;\ Y|W)]$.
5. *Let $A \subseteq \Omega$ where $\Pr[X \in A] = 1 - \epsilon$. Let $(X|A)$ be a random variable taking values in A defined by*
$$\Pr[(X|A) = s] := \Pr[X = s]/\Pr[X \in A] \quad \textit{for all } s \in A.$$
Then $\Delta\big(X\ ;\ (X|A)\big) = \epsilon$.

Proof. Part (1) is [Sho08, Theorem 8.30]. Part (2) is [Sho08, Theorem 8.32]. Part (3) is [Sho08, Theorem 8.33]. Part (4) is a generalization of part (3) which is proved as follows:

$$\begin{aligned}\Delta(X, W\ ;\ Y, W) &= \frac{1}{2}\sum_{w\in\Omega_{\mathrm{w}}} \Pr[W = w] \sum_{x\in\Omega} \Big|\Pr[X = x|W = w] - \Pr[Y = x|W = w]\Big| \\ &= \sum_{w\in\Omega_{\mathrm{w}}} \Pr[W = w]\ \Delta(X|W = w\ ;\ Y|W = w) = E_W[\Delta(X|W\ ;\ Y|W)]\end{aligned}$$

For part (5) observe that for $s \in A$:

$$\Pr[(X|A) = s] = \Pr[X = s]/(1 - \epsilon) = \Pr[X = s] + \Pr[X = s]\frac{\epsilon}{1-\epsilon}$$

from which it follows that

$$\sum_{s\in A} \big|\Pr[(X|A) = s] - \Pr[X = s]\big| = \sum_{s\in A} \Pr[X = s] \cdot \frac{\epsilon}{1-\epsilon} = \Pr[X \in A] \cdot \frac{\epsilon}{1-\epsilon} = \epsilon.$$

Moreover $\sum_{s\notin A} \big|\Pr[(X|A) = s] - \Pr[X = s]\big| = \sum_{s\notin A} \big|\Pr[X = s]\big| = \epsilon$. Therefore

$$\Delta\big(X\ ;\ (X|A)\big) = (1/2)[\epsilon + \epsilon] = \epsilon$$

□

[1] More precisely, the sum in (3) is over $w \in \Omega_{\mathrm{w}}$ for which $\Pr[W = w] \neq 0$ so that conditioning on $W = w$ is well defined.

Lemma 13. *Let $T = (X, Y)$ be a δ-uniform random variable over $\Omega_x \times \Omega_y$. Suppose X is uniform over Ω_x. For $x \in \Omega_x$ let $Y_x := (Y|X = x)$. Then for $\epsilon > 0$ the set S_ϵ of $x \in \Omega_x$ for which Y_x is (δ/ϵ)-uniform over Ω_y has size at least $(1-\epsilon)|\Omega_x|$.*

Proof. Let U_{x} be an independent uniform random variable over Ω_{x} and U_{y} an independent uniform random variable over Ω_{y}. Then by assumption $\Delta(X\,;\,U_{\mathrm{x}}) = 0$ and therefore

$$\begin{aligned}
E_X[\Delta(Y|X\,;\,U_{\mathrm{y}})] &= E_X(Y|X\,;\,U_{\mathrm{y}}|X) && \text{(by independence of } X \text{ and } U_{\mathrm{y}}) \\
&= \Delta(Y, X\,;\,U_{\mathrm{y}}, X) && \text{(by Lemma 12 part 4)} \\
&\le \Delta(Y, X\,;\,U_{\mathrm{y}}, U_{\mathrm{x}}) + \Delta(U_{\mathrm{y}}, U_{\mathrm{x}}\,;\,U_{\mathrm{y}}, X) && \text{(by Lemma 12 part 1)} \\
&\le \delta + \Delta(U_{\mathrm{x}}\,;\,X) = \delta && \text{(by assumption)}
\end{aligned}$$

Applying Markov's inequality to the random variable $\Delta(\,(Y|X)\,;\,U_{\mathrm{y}}\,)$ whose expectation is at most δ shows that $\Pr_X[\Delta(Y|X\,;\,U_{\mathrm{y}}\,) \le \delta/\epsilon] \ge 1-\epsilon$. Therefore $\Pr[X \in S_\epsilon] \ge 1-\epsilon$, which proves the lemma. □

Definition 14. 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.

C The left over hash lemma and generalizations

Let X be a random variable taking values in some set Ω. Recall that the guessing probability of X is defined as $\gamma(X) := \max_{x\in\Omega} \Pr[X = x]$. Also, recall that a family of hash functions $\mathcal{H} = \{h : \Omega \to T\}_{h\in\mathcal{H}}$ is universal if for all $x_1 \neq x_2 \in \Omega$ we have that $\Pr_{h\in\mathcal{H}}[h(x_1) = h(x_2)] = 1/|T|$. Let U_T denote a uniform independent random variable in T. The "classic" left-over-hash-lemma states that when h is uniform in $\mathcal{H}$ and independent of X, the distribution $(h, h(X))$ is statistically close to (h, U_T), assuming the random variable X has sufficient min-entropy [HILL99] (see also [Sho08, Theorem 8.37]).

Lemma 15 (left-over hash lemma)**.** *Let $\mathcal{H} = \{h : \Omega \to T\}_{h\in\mathcal{H}}$ be a universal hash family and let h be a uniform random variable in $\mathcal{H}$. Let X be a random variable independent of h and taking values in Ω. Then $(h, h(X))$ is δ-uniform over $\mathcal{H} \times T$ for*

$$\delta \le \frac{1}{2}\sqrt{\gamma(X)\,|T|}\ .$$

We also state a version of the left-over-hash lemma that deals with the case when more information about the variable X is output. For a random variable X and a random variable W defined over Ω_{w} recall that[2]

$$E_W[\gamma(X|W)] := \sum_{w\in\Omega_{\mathrm{w}}} \Pr[W = w]\gamma(X|W = w) \tag{4}$$

This quantity is called the conditional guessing probability of X with respect to W. The following lemma is a slight generalization of Lemma 2.4 from [DORS08]. For completeness we provide a proof.

[2]More precisely, the sum in (4) is over $w \in \Omega_{\mathrm{w}}$ where $\Pr[W = w] \neq 0$.

Lemma 16. *Let $\mathcal{H} = \{h : \Omega \to T\}_{h\in\mathcal{H}}$ be a universal hash family. Let X and W be random variables where X takes values in Ω. Let h be a uniform random variable in $\mathcal{H}$ independent of X and W. Suppose that*

$$E_W\big[\gamma(X|W)\big] \le \delta$$

and let U be an independent uniform random variable in T. Then

$$\Delta(h, h(X), W \ ; \ h, U, W) \le \frac{1}{2}\sqrt{\delta \cdot |T|}$$

Proof. Suppose W takes values in Ω_{w}. For $w \in \Omega_{\mathrm{w}}$ define $R(w) := (X|W = w)$. Since h is independent of $R(w)$ we know by Lemma 15 that the random variable $\big(h, h(R(w))\big)$ is δ'-uniform for $\delta' \le \frac{1}{2}\sqrt{\gamma(R(w))\,|T|}$. We use this fact in transition (5) below. The lemma now follows from the following calculation:

$$\begin{aligned}
&\Delta\big(h,h(X),W \ ; \ h,U,W\big) \\
&\quad = E_W\big[\Delta\big(\ (h,h(X))|W \ \ ; \ \ (h,U)|W\ \big)\big] && \text{(by Lemma 12 part (4))} \\
&\quad = E_W\big[\Delta\big(\ (h,h(X))|W \ \ ; \ \ (h,U)\ \big)\big] && \text{(by indep. of } h,U \text{ from } W) \\
&\quad = E_W\big[\Delta\big(\ h,h(R(W)) \ ; \ (h,U)\ \big)\big] && \text{(by indep. of } h \text{ from } W) \\
&\quad \le E_W\left[\frac{1}{2}\sqrt{\gamma(X|W)\cdot|T|}\ \right] && \text{(by Lemma 15)} \qquad (5) \\
&\quad \le \frac{1}{2}\sqrt{E_W\big[\gamma(X|W)\cdot|T|\big]} && \text{(by concavity of square root)} \\
&\quad \le \frac{1}{2}\sqrt{\delta\cdot|T|} && \text{(by assumption)}
\end{aligned}$$

□

We will need a generalization of Lemma 15 for the case when the same hash function h is applied to multiple correlated random variables $X_1, \ldots, X_m$. The lemma holds as long as the guessing probability of each variable conditioned on the rest is sufficiently small.

Lemma 17. *Let $\mathcal{H} = \{h : \Omega \to T\}_{h\in\mathcal{H}}$ be a universal hash family. Let $X_1, \ldots, X_m$ be random variables taking values in Ω. Let h be a uniform random variable in $\mathcal{H}$ independent of $(X_1, \ldots, X_m)$. Suppose that*

$$E_{X_{i+1},\ldots,X_m}\Big[\gamma\big(X_i \mid (X_{i+1}, \ldots, X_m)\big)\Big] \le \delta \quad \textit{for all } i = 1, \ldots, m.$$

Then $\big(h, h(X_1), \ldots, h(X_m)\big)$ is δ'-uniform over $\mathcal{H} \times T^m$ for $\delta' \le \frac{1}{2}m\sqrt{\delta\cdot|T|}$.

Proof. The proof is similar to the proof of Theorem 8.38 in [Sho08]. Let $U_1, \ldots, U_m$ be mutually independent uniform random variables in T. Define random variables $Z_0, \ldots, Z_m$ as follows:

$$\begin{aligned}
Z_0 &:= (h, h(X_1), \ldots, h(X_m)), \\
Z_i &:= (h, U_1, \ldots, U_i, h(X_{i+1}), \ldots, h(X_m)), \\
Z_m &:= (h, U_1, \ldots, U_m)
\end{aligned}$$

Then

$$\begin{aligned}
\Delta(Z_0; Z_m) &\le \sum_{i=1}^{m} \Delta(Z_{i-1}; Z_i) && \text{(by Lemma 12 part (1))}\\
&\le \sum_{i=1}^{m} \Delta(\ h, U_1, \ldots, U_{i-1}, \quad h(X_i), \quad X_{i+1}, \ldots, X_m\ ; \\
&\qquad\qquad h, U_1, \ldots, U_{i-1}, \quad U_i, \quad X_{i+1}, \ldots, X_m\ ) && \text{(by Lemma 12 part (2))}\\
&= \sum_{i=1}^{m} \Delta(\ h, \quad h(X_i), \quad X_{i+1}, \ldots, X_m\ ; \\
&\qquad\qquad h, \quad U_i, \quad X_{i+1}, \ldots, X_m\ ) && \text{(by Lemma 12 part (3))}\\
&\le \sum_{i=1}^{m} \frac{1}{2}\sqrt{\delta \cdot |T|} = \frac{1}{2} m \sqrt{\delta \cdot |T|} && \text{(by Lemma 16)}
\end{aligned}$$

□

We will apply Lemma 17 to a classic universal hash family described in the following lemma.

Lemma 18. *Let q be a prime and n, m positive integers with $m > n$. For a matrix $A \in \mathbb{Z}_q^{n\times m}$ let $h_A : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ be the function $h_A(v) = A\,v$. Then $\mathcal{H} := \{h_A \ : \ A \in \mathbb{Z}_q^{n\times m}\}$ is universal.*

Proof. Follows from Example 8.39 of [Sho08]. □

The next corollary is the one we actually use. We let $(\mathbb{Z}_q^{n\times m})^*$ denote the set of rank n matrices in $\mathbb{Z}_q^{n\times m}$.

Corollary 19. *Let q be a prime and n, m integers with $m > n$. Let $R = [r_1|\ldots|r_m]$ be a random variable taking values in $\mathbb{Z}_q^{m\times m}$ where $r_1, \ldots, r_m \in \mathbb{Z}_q^m$ are the columns of R. Suppose that*

$$E_{r_{i+1},\ldots,r_m}\Big[\gamma\big(r_i \mid (r_{i+1}, \ldots, r_m)\big)\Big] \le \delta \quad \text{for all } i = 1, \ldots, m. \tag{6}$$

Let B' be a uniform matrix in $(\mathbb{Z}_q^{n\times m})^*$ independent of R. Then the random variable $(B',\ B'\,R)$ is δ'-uniform over $[(\mathbb{Z}_q^{n\times m})^*]^2$ for $\delta' \le \frac{1}{2} m \sqrt{\delta\, q^n} + 3q^{n-m}$.

Proof. Let B be a uniform matrix in $\mathbb{Z}_q^{n\times m}$ independent of R. Then by Lemma 17 and Lemma 18 the random variable $(B,\ B\,R)$ is δ''-uniform over $(\mathbb{Z}_q^{n\times m})^2$ for $\delta'' \le \frac{1}{2} m \sqrt{\delta\, q^n}$.

Let $\mathcal{E}$ be the event that B is rank n. Then $\Pr[\mathcal{E}] \ge 1 - q^{n-m}$ and observe that B' is distributed as $(B|\mathcal{E})$. Therefore, by Lemma 12 part (5) we obtain that $\Delta(B, BR\ ;\ B', B'R) < q^{n-m}$. Similarly if U_1, U_2 are uniform in $\mathbb{Z}_q^{n\times m}$ and U_1', U_2' are uniform in $(\mathbb{Z}_q^{n\times m})^*$ then $\Delta(U_i; U_i') \le q^{n-m}$ for $i = 1, 2$. The lemma now follows from Lemma 12 part (1):

$$\Delta(B', B'R\ ;\ U_1', U_2') \le \Delta(B', B'R\ ;\ B, BR) + \Delta(B, BR\ ;\ U_1, U_2) + \Delta(U_1, U_2\ ;\ U_1', U_2') \le q^{n-m} + \delta'' + 2q^{n-m}$$

□

D Properties of Discrete Gaussians

The smoothing parameter. For an n-dimensional lattice Λ and positive real $\epsilon > 0$ the smoothing parameter $\eta_\epsilon(\Lambda)$ of Λ is the smallest s such that $\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq \epsilon$ [MR04].

The weight of a lattice. The following lemma bounds the Gaussian weight of a lattice Λ.

Lemma 20 ([Reg05, Claim 3.8])**.** *For a lattice Λ in $\mathbb{R}^m$, vector $c \in \mathbb{R}^m$ and $\sigma > \eta_\epsilon(\Lambda)$ we have $\rho_{\sigma,c}(\Lambda) = \sigma^m det(\Lambda^*)(1+\delta)$ where $|\delta| < \epsilon$. When $c = 0$ then $\delta \in [0, \epsilon]$.*

We will also need the following bounds on $\rho(\mathbb{Z}^m)$.

Lemma 21. *For all $m \in \mathbb{Z}_{>0}$ and reals $\epsilon > 0$ and $\sigma > \sqrt{\log(2m(1+1/\epsilon))/\pi}$ we have*

$$\sigma^m \leq \rho_\sigma(\mathbb{Z}^m) \leq \sigma^m(1+\epsilon) \tag{7}$$

In particular, for any function $\sigma(m) > \omega(\sqrt{\log m})$ there is a negligible function $\epsilon(m)$ such that (7) holds.

Proof. Since the lattice $\mathbb{Z}^m$ is its own dual and has determinant 1, we obtain from Lemma 20 (with $c = 0$) that $\sigma^m \leq \rho_\sigma(\mathbb{Z}^m) \leq (1+\epsilon)\sigma^m$ for $\sigma > \eta_\epsilon(\mathbb{Z}^m)$. Since all the successive minima of $\mathbb{Z}^m$ are equal to 1, Lemma 3.3 of [MR04] shows that the smoothing parameter of $\mathbb{Z}^m$ satisfies $\eta_\epsilon(\mathbb{Z}^m) \leq \sqrt{\log(2m(1+1/\epsilon))/\pi}$, from which the lemma follows. □

The weight outside a ball around the origin. Banaszczyk bounds the weight of a discrete Gaussian outside a certain ball in $\mathbb{R}^m$. In the following, $\mathcal{B}_\infty^m$ denotes the closed m-dimensional L_∞ unit ball centered at the origin.

Lemma 22 ([Ban95, Lemma 2.10])**.** *For all m-dimensional lattices $\Lambda \subset \mathbb{R}^m$ and any real $r > 0$ we have*

$$\rho(\Lambda \setminus r\mathcal{B}_\infty^m) < 2m \exp(-\pi r^2)\ \rho(\Lambda)$$

We restate the lemma so that it is easier for us to use.

Corollary 23. *For all m-dimensional lattices Λ and reals σ, r where $2m\, e^{-\pi(r/\sigma)^2} < 1$ we have*

$$\rho_\sigma(\Lambda) \leq \rho_\sigma(\Lambda \cap r\mathcal{B}_\infty^m) \,/\, \left(1 - 2m\, e^{-\pi(r/\sigma)^2}\right)$$

Proof.

$$\begin{aligned}
\rho_\sigma(\Lambda) &= \rho_\sigma(\Lambda \cap r\mathcal{B}_\infty^m) + \rho_\sigma(\Lambda \setminus r\mathcal{B}_\infty^m) \\
&= \rho_\sigma(\Lambda \cap r\mathcal{B}_\infty^m) + \rho(\sigma^{-1}(\Lambda \setminus r\mathcal{B}_\infty^m)) \\
&\leq \rho_\sigma(\Lambda \cap r\mathcal{B}_\infty^m) + 2m \exp(-\pi(r/\sigma)^2)\ \rho(\sigma^{-1}\Lambda) \\
&= \rho_\sigma(\Lambda \cap r\mathcal{B}_\infty^m) + 2m \exp(-\pi(r/\sigma)^2)\ \rho_\sigma(\Lambda)
\end{aligned}$$

Solving for $\rho_\sigma(\Lambda)$ proves the corollary. □

Using these facts we bound the guessing probability of $\mathcal{D}_{\mathbb{Z}^m,\sigma}$ modulo a prime q (the guessing probability γ is defined in Section C).

Lemma 24. *Let q be a prime, $m > 2$ an integer and $0 < \sigma < q/\ln m$. Let r be an independent random variable sampled from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$. Then*

$$\gamma(r \bmod q) \leq 2/\sigma^m \ .$$

Proof. Since the highest point in the distribution $r \bmod q$ is when $r \bmod q = 0$, it suffices to bound the probability that r is in $q\mathbb{Z}^m$.

$$\gamma(r \bmod q) = \Pr[r \in q\mathbb{Z}^m] = \rho_\sigma(q\mathbb{Z}^m)/\rho_\sigma(\mathbb{Z}^m) \leq \rho_\sigma(q\mathbb{Z}^m)/\sigma^m \tag{8}$$

where the last inequality follows from Lemma 21. Then by applying Corollary 23 to the lattice $q\mathbb{Z}^m$ with $r := q - 0.0001$ we obtain that

$$\rho_\sigma(q\mathbb{Z}^m) \leq \rho_\sigma(0)/(1 - 2me^{-\pi(q/\sigma)^2}) \leq \rho_\sigma(0)/0.5 = 2 \tag{9}$$

where the second inequality follows from $\sigma < q/\ln m$. The lemma follows by combining (8) and (9). □

Large deviation bounds: Micciancio and Regev showed that the norm of vectors sampled from discrete Gaussians is small with high probability.

Lemma 25 ([MR04, Lemma 4.4])**.** *For any m-dimensional lattice Λ, any $c \in \mathbb{R}^m$, and any two reals $\epsilon \in (0,1)$ and $\sigma \geq \eta_\epsilon(\Lambda)$,*

$$\Pr\left\{ \ x \sim \mathcal{D}_{\Lambda,\sigma,c} \ : \ \|x - c\| > \sqrt{m}\,\sigma \ \right\} \ \leq \ \frac{1+\epsilon}{1-\epsilon}\,2^{-m}$$

E The LWE hardness assumption

LWE (learning with errors) is a classic hard problem on lattices. It has been extensively studied, and is defined, e.g., in [Reg05]. We give an equivalent restatement of this decisional problem:

Definition 26. 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 noisy pseudo-random samples of the form $(u_i, v_i) = \left(u_i,\ u_i^T\, s + x_i\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where, $s \in \mathbb{Z}_q^n$ is a uniformly distributed persistent secret key that is invariant across invocations, $x_i \in \mathbb{Z}_q$ is a freshly generated ephemeral additive noise component with distribution χ, and $u_i \in \mathbb{Z}_q^n$ is a fresh uniformly distributed vector revealed as part of the output.

$\mathcal{O}_\$$: outputs truly random samples $\left(u_i, v_i\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, drawn independently uniformly at random in the entire domain $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

The $(\mathbb{Z}_q, n, \chi)$-LWE problem statement, or LWE for short, allows an unspecified number of queries to be made to the challenge oracle $\mathcal{O}$, with no stated prior bound. We say that an algorithm $\mathcal{A}$ decides the $(\mathbb{Z}_q, n, \chi)$-LWE problem if $\left|\Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_\$} = 1]\right|$ is non-negligible for a random $s \in \mathbb{Z}_q^n$.

Regev [Reg05] shows that the 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 [Pei09a]).

Definition 27. Consider a real parameter $\alpha = \alpha(n) \in (0,1)$ and a prime $q = q(n) > 2\sqrt{n}/\alpha$. 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 28 ([Reg05])**.** *If there exists an efficient, possibly quantum, algorithm for deciding the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem 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 28 that deciding the LWE problem is hard when n/α is polynomial in n.

The following lemma, which is implicit in [GPV08, Lemma 8.2], will be used to show that decryption in our systems works correctly.

Lemma 29. *Let $e \in \mathbb{Z}^m$ be a vector satisfying $\|e\| \le \sigma\sqrt{m}$ for some σ. Let $x \in \mathbb{Z}_q^m$ be sampled from $(\overline{\Psi}_\alpha)^m$ and treat x as a vector in $[-q/2, q/2]^m$. Then with overwhelming probability*

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

For decryption we will need $|x^\top e| \le q/5$ which implies that $\alpha < \left(\sigma\sqrt{m}\,\omega(\sqrt{\log m})\right)^{-1}$. Clearly we will also need $m\sigma < q/5$.

F Linear Independence of Gaussian Samples

For a lattice Λ, Regev showed that sampling m^2 vectors from $\mathcal{D}_{\Lambda,\sigma}$, for sufficiently large σ, produces m linearly independent vectors in $\mathbb{Z}^m$ with high probability [Reg05, Corollary 3.16]. In this section we strengthen this result in two ways. First, we show that sampling $\omega(m \log m)$ vectors is sufficient. Second, and more importantly, we show that sampling $\omega(m \log m)$ vectors from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$ will result in m linearly independent vectors in $\mathbb{Z}^m$ that are also linearly independent modulo q for sufficiently large primes q. We start with the second result.

In what follows we say that vectors $r_1, \ldots, r_\ell$ in $\mathbb{Z}^m$ are $\mathbb{Z}_q$-linearly independent if they are linearly independent when reduced modulo q. Similarly, we say that a matrix $R \in \mathbb{Z}^{m\times m}$ is $\mathbb{Z}_q$-invertible if it is invertible when reduced modulo q.

F.1 Linear independence over $\mathbb{Z}_q$

We prove the following theorem which plays an important role in our proofs of security.

Theorem 30. *Let $m \ge n$, $\sigma > \omega(\sqrt{\log m})$ and $q > \sigma\omega(\sqrt{\log m})$. Then the probability that m vectors in $\mathbb{Z}^m$ sampled independently from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$ are $\mathbb{Z}_q$-linearly independent is at least $1 - \frac{1+\epsilon}{\sigma-1}$ for some negligible function ϵ.*

We first derive a bound on the Gaussian weight of an m-dimensional lattice $\Lambda_q(A)$.

Lemma 31. *Let $m \geq n$ and let $q > 2$ be a prime. Let $\sigma \in \mathbb{R}_{>0}$ and define $\epsilon = 2me^{-(\pi/4)\,(q/\sigma)^2}$. If $\epsilon < 1$ then for all matrices $A \in \mathbb{Z}_q^{n\times m}$ we have that*

$$\rho_\sigma(\Lambda_q(A)) \leq \rho_\sigma(\mathbb{Z}^n) \,/\, (1-\epsilon)\ .$$

Proof. First, we can assume without loss of generality that A is rank n. To see why observe that if A is not rank n then there is some rank n matrix A' in $\mathbb{Z}_q^{n\times m}$ such that $\Lambda_q(A) \subseteq \Lambda_q(A')$. Therefore $\rho_\sigma(A) \leq \rho_\sigma(A')$ and it suffices to bound $\rho_\sigma(A')$.

Now, suppose A is rank n. Then there must exist n positions $i_1, \ldots, i_n \in \{1, \ldots, m\}$ such that these n columns of A form a rank n matrix in $\mathbb{Z}_q^{n\times n}$ (otherwise, the column rank of A is at most $n-1$). Define the projection map:

$$\pi : \Lambda_q(A) \to \mathbb{Z}^n \qquad \text{defined by} \qquad \pi(\ (x_1, \ldots, x_m)\ ) \to (x_{i_1}, \ldots, x_{i_n})$$

This map has two important properties on the set $T := \Lambda_q(A) \cap (q/2)\mathcal{B}_\infty^m$.

Property 1: $\rho_\sigma(x) \leq \rho_\sigma(\pi(x))$ for all $x \in T$. This follows from the fact that $\|\pi(x)\| \leq \|x\|$ and the definition of ρ_σ.

Property 2: The map π is injective on T, that is $\pi(x) \neq \pi(y)$ for all $x \neq y \in T$. To see why, suppose that $x \neq y \in T$ satisfy $\pi(x) = \pi(y)$. Then there are s_x, s_y in $\mathbb{Z}_q^n$ such that $x = A^\top s_x$ and $y = A^\top s_y$ modulo q. Let $A_n \in \mathbb{Z}_q^{n\times n}$ be the matrix whose columns are the n columns of A at positions $i_1, \ldots, i_n$. Recall that A_n is full rank. Then $A_n^\top s_x = A_n^\top s_y$ and therefore $s_x = s_y$ in $\mathbb{Z}_q^n$. But then $x = A^\top s_x = A^\top s_y = y$ modulo q which implies that $x - y \in q\mathbb{Z}^m \setminus \{0\}$. This contradicts the fact that both x and y are inside $(q/2)\mathcal{B}_\infty^m$ proving that π is injective on T.

Using these two properties we obtain:

$$\rho_\sigma(T) = \sum_{x\in T} \rho_\sigma(x) \leq \sum_{x\in T} \rho_\sigma(\pi(x)) \leq \sum_{x\in\mathbb{Z}^n} \rho_\sigma(x) = \rho_\sigma(\mathbb{Z}^n) \tag{10}$$

The first inequality follows from property (1) and the second from property (2). Plugging (10) into Corollary 23 we obtain:

$$\rho_\sigma(\Lambda_q(A)) \leq \rho_\sigma(\mathbb{Z}^n) / \big(1 - 2m\,\exp(-\pi(q/(2\sigma))^2)\big)$$

as required. □

Corollary 32. *Let $m \geq n$, $\sigma > \omega(\sqrt{\log m})$ and q a prime such that $q > \sigma\omega(\sqrt{\log m})$. Let $v \in \mathbb{Z}^m$ be a vector sampled from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$. Then for all matrices A in $\mathbb{Z}_q^{n\times m}$ there is a negligible function $\epsilon = \epsilon(m)$ such that*

$$\Pr[v \in \Lambda_q(A)] \leq (1+\epsilon)\,\sigma^{-(m-n)}$$

Proof. Since $\sigma > \omega(\sqrt{\log m})$ we know by Lemma 21 that there is a negligible function $\delta = \delta(m)$ such that

$$\frac{\rho_\sigma(\mathbb{Z}^n)}{\rho_\sigma(\mathbb{Z}^m)} \leq \frac{\sigma^n(1+\delta)}{\sigma^m} = \frac{1+\delta}{\sigma^{m-n}}$$

Let $\tau = 2m \exp\big(-(\pi/4)(q/\sigma)^2\big)$. Then by Lemma 31

$$\Pr[v \in \Lambda_q(A)] = \frac{\rho_\sigma(\Lambda_q(A))}{\rho_\sigma(\mathbb{Z}^m)} \leq \frac{\rho_\sigma(\mathbb{Z}^n)}{(1-\tau)\rho(\mathbb{Z}^m)} \leq \frac{1+\delta}{1-\tau} \cdot \sigma^{-(m-n)}$$

Since $q/\sigma > \omega(\sqrt{\log m})$ we know that $\tau = \tau(m)$ is a negligible function, and the corollary follows. □

Proof of Theorem 30. Let $v_1, \ldots, v_m$ be m vectors in $\mathbb{Z}^m$ sampled independently from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$. For $i = 1, \ldots, m$ let $A_i \in \mathbb{Z}_q^{i \times m}$ be the matrix whose rows are the vectors $v_j \bmod q$ for $j = 1, \ldots, i$. Then the probability that $v_1, \ldots, v_m$ are not $\mathbb{Z}_q$ linearly independent is at most $\sum_{i=1}^{m-1} \Pr[v_{i+1} \in \Lambda_q(A_i)]$. By Corollary 32 there is a negligible function $\epsilon = \epsilon(m)$ such that

$$\sum_{i=1}^{m-1} \Pr[v_{i+1} \in \Lambda_q(A_i)] \leq \sum_{i=1}^{m-1} (1+\epsilon)\,\sigma^{-(m-i)} = (1+\epsilon) \sum_{i=1}^{m-1} \sigma^{-(m-i)} \leq (1+\epsilon) \cdot \frac{1}{\sigma - 1}$$

as required. □

F.2 Linear independence over $\mathbb{Z}$

We generalize Regev's analysis of linear independence [Reg05, Corollary 3.16] to show that sampling m vectors from an m-dimensional lattice Λ gives m linearly independent vectors with constant probability. The main theorem for this section is the following.

Theorem 33. *Let Λ be an m-dimensional lattice and $\sigma > k\,\eta_\epsilon(\Lambda)$ for some $k > 1$ and $\epsilon > 0$. Then the probability that m vectors in $\mathbb{Z}^m$ sampled independently from $\mathcal{D}_{\Lambda,\sigma}$ are linearly independent in $\mathbb{R}^m$ is at least $1 - \frac{1+\epsilon}{k-1}$.*

When $k = 3 + 2\epsilon$ we see that m samples from $\mathcal{D}_{\Lambda,\sigma}$ are linearly independent with probability at least $1/2$. Therefore, to generate a random basis of Λ by sampling from $\mathcal{D}_{\Lambda,\sigma}$ (as in algorithm RandBasis) we would need to sample two sets of m vectors in expectation. To generate a basis with high probability we would need to generate $\omega(\log m)$ sets of m samples each, for a total of $\omega(m \log m)$ samples. If $\sigma > \omega(\eta_\epsilon(\Lambda))$ then $O(m \log m)$ samples are sufficient. This improves the running time analysis of algorithm RandBasis which, using Regev's Corollary 3.16, needed m^2 samples.

To prove Theorem 33 we first generalize Regev's Lemma 3.15.

Lemma 34. *Let Λ be an m-dimensional lattice and $\sigma > k\,\eta_\epsilon(\Lambda)$ for some $k > 1$ and $\epsilon > 0$. Let H be a subspace of $\mathbb{R}^m$ of dimension at most $m - n$ for some $1 \leq n \leq m$. Then when x is sampled from $\mathcal{D}_{\Lambda,\sigma}$ the probability that $x \in H$ is at most $(1+\epsilon)/k^n$.*

Proof. Since ρ is unchanged under rotation, we may assume without loss of generality that H is orthogonal to the first n rows of the $m \times m$ identity matrix. Let $x = (x_1, \ldots, x_m)$ and we bound the probability that $x \in H$ namely that $x_1 = \ldots = x_n = 0$. First, let $d_1, \ldots, d_m \in \mathbb{R}_{>0}$ and $d = \max_i(d_i)$. Then by the Poisson summation formula:

$$\sum_{x \in \Lambda} \prod_{i=1}^{m} e^{-\pi(d_i x_i/\sigma)^2} = \frac{\det(\Lambda^*)\sigma^m}{d_1 \cdots d_m} \sum_{x \in \Lambda^*} \prod_{i=1}^{m} e^{-\pi(\sigma x_i/d_i)^2} \leq \frac{\det(\Lambda^*)\sigma^m}{d_1 \cdots d_m}\, \rho_{d/\sigma}(\Lambda^*) \tag{11}$$

and when $\sigma/d > \eta_\epsilon(\Lambda)$ we know that $\rho_{d/\sigma}(\Lambda^*) \leq 1+\epsilon$. Now, let $t = \sqrt{k^2-1}$. Then using (11) we obtain

$$E\left[\prod_{i=1}^{n} e^{-\pi(tx_i/\sigma)^2}\right] = \frac{1}{\rho_\sigma(\Lambda)} \sum_{x\in\Lambda}\left[\prod_{i=1}^{n} e^{-\pi(tx_i/\sigma)^2} \prod_{i=1}^{m} e^{-\pi(x_i/\sigma)^2}\right]$$

$$= \frac{1}{\rho_\sigma(\Lambda)} \sum_{x\in\Lambda}\left[\prod_{i=1}^{n} e^{-\pi(kx_i/\sigma)^2} \prod_{i=n+1}^{m} e^{-\pi(x_i/\sigma)^2}\right]$$

$$\leq \frac{\det(\Lambda^*)\sigma^m}{k^n\rho_\sigma(\Lambda)}\,\rho_{k/\sigma}(\Lambda^*) \leq \frac{\det(\Lambda^*)\sigma^m}{k^n\rho_\sigma(\Lambda)}\,(1+\epsilon) \leq \frac{1+\epsilon}{k^n}$$

where the last inequality follows from $\rho_\sigma(\Lambda) \geq \det(\Lambda^*)\sigma^m$ by the Poisson summation formula. By Markov's inequality we therefore obtain

$$\Pr[x_1 = \ldots = x_n = 0] = \Pr\left[\prod_{i=1}^{n} e^{-\pi(tx_i/\sigma)^2} = 1\right] \leq (1+\epsilon)/k^n$$

as required. □

Proof of Theorem 33. Let $v_1, \ldots, v_m$ be m vectors in $\mathbb{Z}^m$ sampled independently from $\mathcal{D}_{\Lambda,\sigma}$. The probability that $v_1, \ldots, v_m$ are not linearly independent is at most $\sum_{i=1}^{m-1} \Pr[v_{i+1} \in \mathrm{span}_{\mathbb{R}}(v_1, \ldots, v_i)]$. By Lemma 34

$$\sum_{i=1}^{m-1} \Pr[v_{i+1} \in \mathrm{span}_{\mathbb{R}}(v_1, \ldots, v_i)] \leq \sum_{i=1}^{m-1} (1+\epsilon)/k^i = (1+\epsilon)\sum_{i=1}^{m-1} k^{-i} \leq (1+\epsilon)/(k-1)$$

as required. □

G Proof of Theorem 6

Our first step is a simpler expression for the statistical distance between $\mathrm{DIST}_0(\sigma, m)$ and $\mathrm{DIST}_1(A, \sigma, m, q)$. For a fixed rank n matrix $A \in \mathbb{Z}_q^{n\times m}$ define the following four random variables:

- Let R_0 be distributed as $\mathrm{DIST}_0(\sigma, m)$ and $R_{1,A}$ distributed as $\mathrm{DIST}_1(A, \sigma, m, q)$.
- Let B be a uniform random variable in $(\mathbb{Z}_q^{n\times m})^*$ (namely a uniform rank n matrix in $\mathbb{Z}_q^{n\times m}$).
- Let B'_A be defined as follows: sample R as $\mathrm{DIST}_0(\sigma, m)$ and set $B'_A \leftarrow AR^{-1} \bmod q$.

The following lemma relates these variables.

Lemma 35. *For a fixed rank n matrix $A \in \mathbb{Z}_q^{n\times m}$ we have $\Delta(R_0; R_{1,A}) = \Delta(B; B'_A)$.*

For completeness, before proving Lemma 35 we state the following immediate lemma.

Lemma 36. *Let $\mathcal{D}$ be a distribution and let $\mathcal{E}$ be some event. Consider the random variable R' defined by: (1) sample a new independent R according to $\mathcal{D}$, and (2) if $R \in \mathcal{E}$ output $R' \leftarrow R$; otherwise go back to step (1). Then R' is distributed as $(R|\mathcal{E})$.*

Proof. For $r \in \mathcal{E}$ we know that $\Pr[R' = r]$ is the probability that $R = r$ in the first iteration, plus the probability that $R = r$ in the second iteration, and so on. Therefore,

$$\Pr[R' = r] = \Pr[R = r]\big(1 + Pr[\neg\mathcal{E}] + Pr[\neg\mathcal{E}]^2 + \dots\big) = \frac{\Pr[R = r]}{1 - \Pr[\neg\mathcal{E}]} = \frac{Pr[R = r]}{\Pr[\mathcal{E}]} = \Pr[R = r|\mathcal{E}]$$

which proves the lemma. □

Proof of Lemma 35. Fix some rank n matrix $A \in \mathbb{Z}_q^{n\times m}$ and define the following terminology.

- Let $\mathcal{D}$ be the distribution $(\mathcal{D}_{\mathbb{Z}^m,\sigma})^m$ on $\mathbb{Z}^{m\times m}$.
- Let $\Omega' \subseteq \Omega$ be the set of $\mathbb{Z}_q$-invertible matrices in $\mathbb{Z}^{m\times m}$.
  We let $\mathcal{D}'$ be the distribution on Ω' defined by $\quad \forall R \in \Omega' : \quad \mathcal{D}'(R) := \mathcal{D}(R)/\mathcal{D}(\Omega')$.
- For a matrix $\beta \in \mathbb{Z}_q^{n\times m}$ let Ω_β be the set of matrices $R \in \mathbb{Z}^{m\times m}$ such that $\beta R = A \pmod q$.
  Let $\Omega'_\beta := \Omega_\beta \cap \Omega'$.
- For $\beta \in \mathbb{Z}_q^{n\times m}$ define the distribution $\mathcal{D}_\beta$ on Ω_β as $\quad \forall R \in \Omega_\beta : \quad \mathcal{D}_\beta(R) := \mathcal{D}(R)/\mathcal{D}(\Omega_\beta)$.

Observe that the distribution $\mathcal{D}_{\Lambda_q^{a_1}(\beta),\sigma} \times \cdots \times \mathcal{D}_{\Lambda_q^{a_m}(\beta),\sigma}$ used in Step 2 of $\mathrm{DIST}_1(A, \sigma, m, q)$ is identical to the distribution $\mathcal{D}_\beta$ on Ω_β. Indeed, the weight of a matrix $R = [r_1|\dots|r_m]$ in Ω_β under the distribution $\mathcal{D}_{\Lambda_q^{a_1}(\beta),\sigma} \times \cdots \times \mathcal{D}_{\Lambda_q^{a_m}(\beta),\sigma}$ is

$$\prod_{i=1}^{m} \frac{\rho_\sigma(r_i)}{\rho_\sigma(\Lambda_q^{a_i}(\beta))} = \prod_{i=1}^{m} \frac{\rho_\sigma(r_i)/\rho_\sigma(\mathbb{Z}^m)}{\rho_\sigma(\Lambda_q^{a_i}(\beta))/\rho_\sigma(\mathbb{Z}^m)} = \prod_{i=1}^{m} \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma}(r_i)}{\mathcal{D}_{\mathbb{Z}^m,\sigma}(\Lambda_q^{a_i}(\beta))} = \frac{\mathcal{D}(R)}{\mathcal{D}(\Omega_\beta)} = \mathcal{D}_\beta(R)$$

Next, observe that for every matrix $R \in \Omega'$ there is exactly one matrix $\beta \in \mathbb{Z}_q^{n\times m}$ for which R is in Ω'_β. This β is simply $\beta = AR^{-1} \bmod q$. Hence, the sets Ω'_β where $\beta \in \mathbb{Z}_q^{n\times m}$ are a partition of Ω'.

Using these facts we can calculate the weight of $R \in \Omega'$ under the distribution $\mathrm{DIST}_1(A, \sigma, m, q)$. Let $\beta \in \mathbb{Z}_q^{n\times m}$ be the unique matrix such that $R \in \Omega'_\beta$. Let $\mathcal{E}$ be the event that a matrix sampled from $\mathcal{D}_\beta$ is in Ω'. Then

$$\begin{aligned}
\Pr[R_{1,A} = R] &= \Pr[B = \beta]\ \mathcal{D}_\beta(R)/\Pr[\mathcal{E}] && \text{(by Lemma 36)}\\
&= \Pr[B = \beta]\ \mathcal{D}(R)/\mathcal{D}(\Omega'_\beta) && \text{(since } \Pr[\mathcal{E}] = \mathcal{D}(\Omega'_\beta)/\mathcal{D}(\Omega_\beta)\ \text{)}\\
&= \Pr[B = \beta] \cdot \mathcal{D}'(R)/\mathcal{D}'(\Omega'_\beta)\ .
\end{aligned}$$

Similarly, by Lemma 36, for all $R \in \Omega'$ the weight of R under $\mathrm{DIST}_0(\sigma, m)$ is

$$\Pr[R_0 = R] = \frac{\mathcal{D}(R)}{\mathcal{D}(\Omega')} = \mathcal{D}'(R)$$

Now, since the sets Ω'_β are a partition of Ω' we obtain that

$$\begin{aligned}
\Delta(R_0\,;\,R_{1,A}) &= \sum_{\beta\in(\mathbb{Z}_q^{n\times m})^*}\sum_{R\in\Omega'_\beta}\big|\Pr[R_0=R]-\Pr[R_{1,A}=R]\big| \\
&= \sum_{\beta\in(\mathbb{Z}_q^{n\times m})^*}\sum_{R\in\Omega'_\beta}\left|\mathcal{D}'(R)-\Pr[B=\beta]\frac{\mathcal{D}'(R)}{\mathcal{D}'(\Omega'_\beta)}\right| \\
&= \sum_{\beta\in(\mathbb{Z}_q^{n\times m})^*}\left[\left|1-\Pr[B=\beta]\frac{1}{\mathcal{D}'(\Omega'_\beta)}\right|\sum_{R\in\Omega'_\beta}\mathcal{D}'(R)\right] \\
&= \sum_{\beta\in(\mathbb{Z}_q^{n\times m})^*}\left[\left|1-\Pr[B=\beta]\frac{1}{\mathcal{D}'(\Omega'_\beta)}\right|\mathcal{D}'(\Omega'_\beta)\right] \\
&= \sum_{\beta\in(\mathbb{Z}_q^{n\times m})^*}\big|\mathcal{D}'(\Omega'_\beta)-\Pr[B=\beta]\big|
\end{aligned}$$

Since $\mathcal{D}'(\Omega'_\beta)=\Pr[R_0\in\Omega'_\beta]=\Pr[\beta R_0=A]=\Pr[B'_A=\beta]$ we obtain that

$$\Delta(R_0\,;\,R_{1,A})=\sum_{\beta\in(\mathbb{Z}_q^{n\times m})^*}\big|\Pr[B'_A=\beta]-\Pr[B=\beta]\big|=\Delta(B'_A\,;\,B)$$

as required. □

Next, to bound $\Delta(B;B'_A)$ we first show that the distribution (B,BR) is close to uniform. This is a direct consequence of the generalized left over hash lemma in Corollary 19.

Lemma 37. *Let q be a prime and $m>n$. Let R be a random variable sampled according to $DIST_0(\sigma,m)$ with σ in the range $3\le\sigma<q/\ln m$. Let B be a uniform random rank n matrix in $\mathbb{Z}_q^{n\times m}$ independent of R. Then the distribution $(B,\,BR \bmod q)$ is δ'-uniform over $[(\mathbb{Z}_q^{n\times m})^*]^2$ for*

$$\delta'\le m\sqrt{q^n/\sigma^m}\;+3q^{n-m}\;.$$

In particular, if $m>2n\log q$ then $(B,BR \bmod q)$ is statistically close to uniform over $[(\mathbb{Z}_q^{n\times m})^*]^2$.

Proof. We prove the lemma using Corollary 19. To apply the corollary it suffices to show that $\mathrm{DIST}_0(\sigma,m)$ modulo q satisfies condition (6) with $\delta\le 4/\sigma^m$. We prove condition (6) for $i=1$. The analysis for other i is the same.

First, let r be an independent random variable sampled from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$ and taking values in $\mathbb{Z}^m$. By Lemma 24 the guessing probability of $r \bmod q$ is bounded by

$$\gamma(r \bmod q)\le 2/\sigma^m\;.$$

Next, Let $\rho_2,\ldots,\rho_m\in\mathbb{Z}^m$ be $\mathbb{Z}_q$-linearly independent vectors and let $\mathcal{E}$ be the event that the matrix $[r|\rho_2|\ldots|\rho_m]$ is not full rank modulo q. This event happens whenever $r\in\Lambda_q([\rho_2|\ldots|\rho_m])$ and therefore, by Corollary 32 there is a negligible function ϵ such that

$$\Pr[\mathcal{E}]=\Pr\left[r\in\Lambda_q([\rho_2|\ldots|\rho_m])\right]\le(1+\epsilon)/\sigma<0.5\;.$$

It follows that

$$\gamma(r \bmod q \mid \neg\mathcal{E}) \leq \gamma(r \bmod q)/\Pr[\neg\mathcal{E}] \leq (2/\sigma^m)/\Pr[\neg\mathcal{E}] \leq 4/\sigma^m \ . \tag{12}$$

Now, let $r_1, \ldots, r_m \in \mathbb{Z}^m$ be the columns of R. Consider the random variable

$$\hat{r}_1 := r_1 | (r_2 = \rho_2 \wedge \ldots \wedge r_m = \rho_m)$$

which takes values in $\mathbb{Z}^m$. Observe that $\hat{r}_1$ is distributed as $(r|\neg\mathcal{E})$ and therefore

$$\gamma(\hat{r}_1 \bmod q) = \gamma(r \bmod q \mid \neg\mathcal{E}) \leq 4/\sigma^m \ . \tag{13}$$

Since (13) holds for all $\mathbb{Z}_q$-linearly independent $\rho_2, \ldots, \rho_m \in \mathbb{Z}^m$ it follows that

$$E_{r_2,\ldots,r_m}[\gamma(r_1|(r_2, \ldots, r_m))] \leq 4/\sigma^m \ .$$

We note that tuples $\rho_2, \ldots, \rho_m \in \mathbb{Z}^m$ that are not $\mathbb{Z}_q$-linearly independent can be ignored since they can never be columns of R and hence will have no impact the expectation. Corollary 19 now applies with $\delta = 4/\sigma^m$ which completes the proof of the lemma. □

Now we can complete the proof of Theorem 6.

Proof of Theorem 6. Let B be uniform in $(\mathbb{Z}_q^{n\times m})^*$ and R be chosen from $\mathrm{DIST}_0(\sigma, m)$. Set $A \leftarrow BR \bmod q$. Then by Lemma 37 the pair (A, B) is δ'-uniform in $[(\mathbb{Z}_q^{n\times m})^*]^2$. Moreover, since B is uniform and R is invertible the random variable A is also uniform in $(\mathbb{Z}_q^{n\times m})^*$. Then, applying Lemma 13 to (A, B) with $X := A$ and $Y := B$ shows that there is a set $\mathcal{A}_{\text{good}}$ containing at least $(1 - q^{-n})$ of $(\mathbb{Z}_q^{n\times m})^*$ such that $(B|A = \alpha)$ is $(\delta' q^n)$-uniform in $(\mathbb{Z}_q^{n\times m})^*$ for all $\alpha \in \mathcal{A}_{\text{good}}$.

For $\alpha, \beta \in (\mathbb{Z}_q^{n\times m})^*$ let Ω'_β be the set of invertible matrices in $\mathbb{Z}^{m\times m}$ such that $\alpha = \beta R$ modulo q. Then the random variable $(B|A = \alpha)$ is the same as the variable B'_α used in Lemma 35 since:

$$\begin{aligned}\Pr\left[(B|A = \alpha) = \beta\right] &= \frac{\Pr[A = \alpha \wedge B = \beta]}{\Pr[A = \alpha]} = \frac{\Pr[R \in \Omega'_\beta \wedge B = \beta]}{\Pr[A = \alpha]} \\ &= \frac{\Pr[R \in \Omega'_\beta]\Pr[B = \beta]}{\Pr[A = \alpha]} = \Pr[R \in \Omega'_\beta] = \Pr[B'_\alpha = \beta]\end{aligned}$$

Therefore, whenever $\alpha \in \mathcal{A}_{\text{good}}$ we have

$$\Delta(R_0; R_{1,A}) = \Delta(B; B'_\alpha) = \Delta(B \ ; \ (B|A = \alpha) \ ) \leq q^n \delta' \leq m\sqrt{q^{3n}/\sigma^m} \ + 3q^{2n-m}$$

which is a negligible value, as required. This completes the proof of the theorem. □

H Proof of Theorem 8

In this section we prove Theorem 8. We begin with a few lemmas bounding the length of certain vectors. The first lemma bounds the inner product of a Gaussian sample in $\mathbb{Z}^m$ with a fixed vector in $\mathbb{Z}^m$.

Lemma 38. *Let $u \in \mathbb{R}^m$ and $\sigma > 0$. Let $r \in \mathbb{Z}^m$ be a random variable sampled from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$. Then the inner product $r^\top u$ satisfies $|r^\top u| \leq \|u\| \, \sigma \, \omega(\log m)$ with overwhelming probability.*

Proof. When $r = (r_1, \ldots, r_m)$ is sampled from $\mathcal{D}_{\mathbb{Z}^m,\sigma}$, each r_i is sampled from $\mathcal{D}_{\mathbb{Z},\sigma}$ and $r_1, \ldots, r_m$ are independent of each other. By Lemma 22, applied to the lattice $\mathbb{Z}$, we know that for all $k > 0$

$$\Pr\left[|r_i| > k\sigma\right] = \frac{\rho_\sigma(\mathbb{Z} \setminus k\sigma\mathcal{B}^1_\infty)}{\rho_\sigma(\mathbb{Z})} = \frac{\rho(\,(1/\sigma)\mathbb{Z} \setminus k\mathcal{B}^1_\infty)}{\rho_\sigma(\mathbb{Z})} \leq \frac{2\,\exp(-\pi k^2)\;\rho_\sigma(\mathbb{Z})}{\rho_\sigma(\mathbb{Z})} = 2\,\exp(-\pi k^2)\;.$$

Taking $k = \omega(\sqrt{\log m})$ we obtain that each r_i is at most $\sigma\omega(\sqrt{\log m})$ in absolute value with overwhelming probability.

Now, consider the inner product $r^\top \cdot u = \sum_{i=1}^m r_i u_i \in \mathbb{R}$. Each summond $r_i u_i$ has expectation 0 and is in the interval $[-I, I]$ for $I := u_i\sigma\omega(\sqrt{\log m})$ with overwhelming probability. Moreover, all the $r_i u_i$ are independent of one another. Then by the Hoeffding bound [Hoe63] we obtain for all $k > 0$ that

$$\Pr\left[|r^\top u| > k \cdot \|u\|\sigma\omega(\sqrt{\log m})\right] < 2e^{-k^2/2} + \epsilon(m)$$

where $\epsilon(m)$ is a negligible function in m (the reason for $\epsilon(m)$ is that $r_i u_i$ are in the interval $[-I, I]$ with overwhelming probability, as opposed to probability 1). The lemma now follows by taking $k := \omega(\sqrt{\log m})$. □

The next lemma shows that a matrix R sampled from $\mathcal{D}_{m\times m}$ does not increase the norm of a vector in $\mathbb{R}^m$ by much.

Lemma 39. *Let R in $\mathbb{Z}^{m\times m}$ be distributed as $(\mathcal{D}_{\mathbb{Z}^m,\sigma_\mathrm{R}})^m$ Let $u \in \mathbb{R}^m$. Then with overwhelming probability $\|Ru\| \leq \|u\| \cdot \sigma_\mathrm{R}\sqrt{m}\,\omega(\log m)$.*

Proof. By Lemma 25 the L_2 norm of all rows of R is at most $\sigma_\mathrm{R}\sqrt{m}$ with overwhelming probability. Then by the Cauchy-Schwartz inequality each coordinate of Ru is at most $\|u\| \cdot \sigma_\mathrm{R}\sqrt{m}$ in absolute value from which it follows that $\|Ru\| \leq \|u\| \cdot \sigma_\mathrm{R}\, m$.

We can do better by using the randomness of the rows of R. Lemma 38 shows that for a vector r sampled from $\mathcal{D}_{\mathbb{Z}^m,\sigma_\mathrm{R}}$ the inner product $|r^\top u|$ is less than $\|u\|\,\sigma_\mathrm{R}\,\omega(\log m)$ with overwhelming probability. Since the rows of R are sampled from a distribution statistically close to $\mathcal{D}_{\mathbb{Z}^m,\sigma_\mathrm{R}}$, applying the union bound to the rows of R shows that with overwhelming probability all coordinates of Ru are less than $\|u\|\,\sigma_\mathrm{R}\,\omega(\log m)$ in absolute value. Hence $\|Ru\|$ is at most $\|u\|\,\sigma_\mathrm{R}\sqrt{m}\,\omega(\log m)$ with overwhelming probability, as required. □

The next lemma bounds the Gram-Schmidt norm of a basis after it is transformed by a matrix R.

Lemma 40. *Let R be a matrix in $\mathbb{R}^{m\times m}$ 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. □

Finally, we can prove Theorem 8.

Proof of Theorem 8. Let $\sigma' := \|\widetilde{T_A}\| \cdot \sigma_{\mathrm{R}} \sqrt{m}\, \omega(\log m)$. Then, with overwhelming probability the vectors in T''_B generated in step 2 satisfy:

$$\begin{aligned}
\|\widetilde{T''_B}\| &\leq \|\widetilde{T'_B}\| && \text{(by Lemma 3)} \\
&\leq \max_{1 \leq i \leq m} \|R\,\tilde{a}_i\|_2 && \text{(by Lemma 40)} \\
&\leq \max_{1 \leq i \leq m} \|\tilde{a}_i\| \cdot \sigma_{\mathrm{R}} \sqrt{m}\, \omega(\log m) && \text{(by Lemma 39)} \\
&= \|\widetilde{T_A}\| \cdot \sigma_{\mathrm{R}} \sqrt{m}\, \omega(\log m) = \sigma'
\end{aligned}$$

Since $\sigma > \sigma'\, \omega(\sqrt{\log m})$ by assumption, algorithm $\mathsf{RandBasis}(T''_B, \sigma)$ outputs a random basis of $\Lambda_q^\perp(B)$ as required. □

I Consistency of the HIBE scheme

Lemma 41. *The HIBE scheme of Section 5.2 is consistent.*

Proof. When the cryptosystem is operated as specified, we have,

$$\begin{aligned}
w &= c_0 - d_{\mathsf{Id}}^\top\, c_1 \\
&= u_0^T\, s + x + b\, \lfloor \frac{q}{2} \rfloor - d_{\mathsf{Id}}^\top \big(F_{\mathsf{Id}}^T\, s + y\big) \\
&= u_0^T\, s + x + b\, \lfloor \frac{q}{2} \rfloor - (F_{\mathsf{Id}} d_{\mathsf{Id}})^\top\, s - d_{\mathsf{Id}}^\top y \\
&= u_0^T\, s + x + b\, \lfloor \frac{q}{2} \rfloor - u_0^T\, s - d_{\mathsf{Id}}^\top y \\
&= x + b\, \lfloor \frac{q}{2} \rfloor - d_{\mathsf{Id}}^\top y \\
&= b\, \lfloor \frac{q}{2} \rfloor + \underbrace{x - d_{\mathsf{Id}}^\top y}_{\text{error term}} \quad \approx b\, \lfloor \frac{q}{2} \rfloor
\end{aligned}$$

Since $\|d_{\mathsf{Id}}\| \leq \sigma_d \sqrt{m}$ by Lemma 25, we know by Lemma 29 that $|d_{\mathsf{Id}}^\top \cdot y|$ is smaller than $q/5$ w.h.p by our choice of parameters. Moreover, since $|x|$ is much smaller than q we obtain that $\|x - d_{\mathsf{Id}} \cdot y\|$ is smaller than $q/4$ and hence decryption is correct with high probability. □

J Selectively Secure HIBE in the Standard Model

We build an HIBE of depth d that is selectively secure without random oracles. The construction is actually a binary tree encryption (BTE) which means that identities at each level are binary (i.e. 0 or 1). To build an HIBE with k-bit identities at each level we would assign k levels of the BTE hierarchy to each level of the HIBE. The parameters used by this system are the same as the parameters used for the random oracle system in Section 5.1

J.1 Construction

Setup$(1^\lambda, 1^d)$**:** On input a security parameter λ and a maximum depth d represented in unary:

1. Use algorithm $\mathsf{TrapGen}(q, n)$ to select a uniformly random $n \times m$-matrix $A \in \mathbb{Z}_q^{n \times m}$ with a basis $T_A \in \mathbb{Z}^{m \times m}$ for $\Lambda_q^\perp(A)$ such that $\|\widetilde{T_A}\| \leq m \cdot \omega(\sqrt{\log m})$.
2. Select a uniformly random n-vector $u_0 \in \mathbb{Z}_q^n$.
3. Select $2d$ square matrices $R_{1,0}, R_{1,1}, \ldots, R_{d,0}, R_{d,1} \in \mathbb{Z}^{m \times m}$ from the distribution $\mathcal{D}_{m \times m}$ by invoking $\mathsf{SampleR}(1^\lambda)$ $2d$ times.
4. Output the public parameters and master key,

$$\mathsf{PP} = \left(A\ ,\ \ u_0\ ,\ \ R_{1,0}, R_{1,1}\ ,\ R_{2,0}, R_{2,1}\ ,\ \ \ldots\ ,\ R_{d,0}, R_{d,1} \right) \qquad \mathsf{MK} = \left(T_A \right)$$

Extract$(\mathsf{PP}, \mathsf{MK}, \mathsf{Id})$**:** On input public parameters PP, a master key MK, and an identity Id of depth $|\mathsf{Id}| = \ell \leq d$:

1. Compute $R_{\mathsf{Id}} \leftarrow R_{\ell,\mathsf{Id}_\ell} \cdot R_{\ell-1,\mathsf{Id}_{\ell-1}} \cdots R_{2,\mathsf{Id}_2} \cdot R_{1,\mathsf{Id}_1}$. Define $F_{\mathsf{Id}} = A \cdot R_{\mathsf{Id}}^{-1} \bmod q$.
2. Construct a randomized basis for $\Lambda_q^\perp(F_{\mathsf{Id}})$ by running $S \leftarrow \mathsf{BasisDel}(A, R_{\mathsf{Id}}, \mathsf{MK} = T_A, \sigma_\ell)$.
3. Output the private key $\mathsf{SK}_{\mathsf{Id}} = S$.

Derive$(\mathsf{PP}, \mathsf{SK}_{\mathsf{Id}_{|\ell-1}}, \mathsf{Id})$**:** On input public parameters PP, a child identity Id of depth ℓ, and a secret key corresponding to the parent identity $\mathsf{Id}_{|\ell-1}$ of depth $\ell - 1$:

1. Define $R_{\mathsf{Id}_{|\ell-1}} = R_{\ell-1,\mathsf{Id}_{\ell-1}} \cdots R_{2,\mathsf{Id}_2} \cdot R_{1,\mathsf{Id}_1}$. Compute $F_{\mathsf{Id}_{|\ell-1}} \leftarrow A \cdot R_{\mathsf{Id}_{|\ell-1}}^{-1} \bmod q$.
   Recall that $\mathsf{SK}_{\mathsf{Id}_{|\ell-1}}$ is a short basis for $\Lambda_q^\perp(F_{\mathsf{Id}_{|\ell-1}})$ and let $F_{\mathsf{Id}} = F_{\mathsf{Id}_{|\ell-1}} R_{\ell,\mathsf{Id}_\ell}^{-1} \bmod q$.
2. Construct a randomized short basis for $\Lambda_q^\perp(F_{\mathsf{Id}})$ as $S' \leftarrow \mathsf{BasisDel}(F_{\mathsf{Id}_{|\ell-1}}, R_{\ell,\mathsf{Id}_\ell}, \mathsf{SK}_{\mathsf{Id}_{|\ell-1}}, \sigma_\ell)$.
3. Output the private key $\mathsf{SK}_{\mathsf{Id}} = S'$.

Encrypt$(\mathsf{PP}, \mathsf{Id}, b)$**:** On input public parameters PP, an identity Id of length $|\mathsf{Id}| = \ell$, and a message $b \in \{0, 1\}$:

1. Construct $R_{\mathsf{Id}} = R_{\ell,\mathsf{Id}_\ell} \cdots R_{2,\mathsf{Id}_2} \cdot R_{1,\mathsf{Id}_1}$.
2. Construct the encryption matrix $F_{\mathsf{Id}} = A \cdot R_{\mathsf{Id}}^{-1} \bmod q$.
3. Pick a uniformly random vector $s \xleftarrow{R} \mathbb{Z}_q^n$.
4. Choose noise vectors $x \xleftarrow{\overline{\Psi}_{\alpha_\ell}} \mathbb{Z}_q$ and $y \xleftarrow{\overline{\Psi}_{\alpha_\ell}^m} \in \mathbb{Z}_q^m$. ($\overline{\Psi}_\alpha$ is defined in Appendix E)
5. Output the ciphertext,

$$\mathsf{CT} = \left(\ c_0 = u_0^T s + x + b \lfloor \frac{q}{2} \rfloor\ ,\ \ c_1 = F_{\mathsf{Id}}^T s + y \ \right) \ \in \mathbb{Z}_q \times \mathbb{Z}_q^m$$

Decrypt$(\mathsf{PP}, \mathsf{SK}_{\mathsf{Id}}, \mathsf{CT})$**:** On input public parameters PP, a private key $\mathsf{SK}_{\mathsf{Id}}$ (say that $|\mathsf{Id}| = \ell$), and a ciphertext CT:

1. Set $d_{\mathsf{Id}} \leftarrow \mathsf{SamplePre}(F_{\mathsf{Id}}, \mathsf{SK}_{\mathsf{Id}}, u_0, \sigma_\ell)$.
   Note that $F_{\mathsf{Id}}\, d_{\mathsf{Id}} = u_0$.
2. Compute $w = c_0 - d_{\mathsf{Id}}^T\, c_1 \in \mathbb{Z}_q$.
3. Compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in $[q] \subset \mathbb{Z}$:
   if they are close, i.e., if $\left| w - \lfloor \frac{q}{2} \rfloor \right| < \lfloor \frac{q}{4} \rfloor$ in $\mathbb{Z}$, then output 1; otherwise output 0.

In Appendix I we show that the scheme is consistent, namely that decryption correctly decrypts all valid ciphertexts.

J.2 Security

Theorem 42. *If there exists a PPT adversary $\mathcal{A}$ with IND-sID-CPA advantage $\epsilon > 0$ against the selective HIBE scheme of Section J.1, then there exists a PPT algorithm $\mathcal{B}$ that decides the LWE problem with advantage $\epsilon/2$.*

Proof. Let $\mathcal{A}$ be an IND-sID-CPA attacker. We will show that a non-negligible advantage in the IND-sID-CPA game can be used to solve the LWE problem. This will prove that under the LWE assumption no polynomial time attacker can have non-negligible advantage in the IND-sID-CPA game.

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

As the number of oracle calls is known *a priori*, the samples can be supplied non-interactively at the beginning, e.g., here in the form of an instance with $(m+1)\,(n+1)$ elements of $\mathbb{Z}_q$.

Targeting. $\mathcal{A}$ announces to $\mathcal{B}$ the identity Id^* that it intends to attack. Say $|\mathsf{Id}^*| = \ell$.

Setup.
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. Choose ℓ low norm $m \times m$ matrices $R_{1,\mathsf{Id}_1^*}, \ldots, R_{\ell,\mathsf{Id}_\ell^*}$ by invoking $R_i \leftarrow \mathsf{SampleR}(1^\lambda)$.
3. Set $R_{\mathsf{Id}^*} = R_{\ell,\mathsf{Id}_\ell^*} \cdots R_{2,\mathsf{Id}_2^*} \cdot R_{1,\mathsf{Id}_1^*}$.
4. Let $A \leftarrow A_0 \cdot R_{\mathsf{Id}^*}^{-1}$.
5. For $i = 1, \ldots, \ell$ construct $R_{i,\overline{\mathsf{Id}_i^*}}$ as follows.

– Let $M_i \leftarrow A \cdot R_{1,\mathsf{Id}_1^*}^{-1} \cdots R_{(i-1),\mathsf{Id}_{i-1}^*}^{-1}$.

– Invoke $R_{i,\overline{\mathsf{Id}_i^*}} \leftarrow \mathsf{SampleRwithBasis}(M_i)$ to obtain a random $R_{i,\overline{\mathsf{Id}_i^*}} \sim \mathcal{D}_{m \times m}$ and a short basis T_{B_i} for $\Lambda_q^\perp(B_i)$ where $B_i = M_i\, R_{i,\overline{\mathsf{Id}_i^*}}^{-1} \bmod q$. Thus $R_{i,\overline{\mathsf{Id}_i^*}}$ is a low-norm matrix such that $B_i \cdot R_{i,\overline{\mathsf{Id}_i^*}} \cdot R_{(i-1),\mathsf{Id}_{(i-1)}^*} \cdots R_{(1),\mathsf{Id}_{(1)}^*} = A$.

6. For $i = \ell+1, \ldots, d$, choose low norm $m \times m$ matrices $R_{\ell+1,0}, R_{\ell+1,1}, \ldots, R_{d,0}, R_{d,1}$ as $R_i \leftarrow \mathsf{SampleR}(1^\lambda)$.
7. Publish the system parameters $\mathsf{PP} = \big(A, u_0, R_{1,0}, R_{1,1}, \ldots, R_{d,0}, R_{d,1}\big)$.

Queries 1. $\mathcal{A}$ makes interactive key-extraction queries on identities Id that may be chosen adaptively. Queried identities Id must not be equal to or be a prefix of the challenge identity Id^* to be admissible. $\mathcal{B}$ answers each query as follows.

1. Construct $R_{\mathsf{Id}} = R_{k,\mathsf{Id}_k} \cdots R_{2,\mathsf{Id}_2} \cdot R_{1,\mathsf{Id}_1}$.
2. Construct $F_{\mathsf{Id}} = A \cdot R_{\mathsf{Id}}^{-1}$.
3. Let $j \in [k]$ be the first position where $\mathsf{Id} \neq \mathsf{Id}^*$. Then $B_j \cdot R_{\mathsf{Id}_{|j}} = A$ by construction and we have a trapdoor for $\Lambda_q^{\perp}(B_j) = \Lambda_q^{\perp}(F_{\mathsf{Id}_{|j}})$.
4. Construct a trapdoor for $F_{\mathsf{Id}} = A \cdot R_{\mathsf{Id}}^{-1} = A \cdot R_{\mathsf{Id}_{|j}}^{-1} \cdot R_{j+1,\mathsf{Id}_{j+1}}^{-1} \cdots R_{k,\mathsf{Id}_k}^{-1}$ by invoking algorithm $\mathsf{BasisDel}(F_{\mathsf{Id}_{|j}}, R_{k,\mathsf{Id}_k} \cdots R_{j+1,\mathsf{Id}_{j+1}}, T_{B_j}, \sigma_k)$.

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. For all $i = 0, \ldots, m$, retrieve $v_i \in \mathbb{Z}_q$ from the LWE instance. Let $v^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m$.
2. Blind the message bit by letting $c_0^* = v_0 + b^* \lfloor \frac{q}{2} \rceil \in \mathbb{Z}_q$.
3. Set $c_1^* = v^* \in \mathbb{Z}_q^m$.
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 $(c_0, c_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^m$ and send (c_0, c_1) to $\mathcal{A}$.

Queries 2. $\mathcal{A}$ makes more private-key queries which are answered by $\mathcal{B}$ in the same manner as before.

Guess. After being allowed to make additional queries, $\mathcal{A}$ guesses whether CT^* was an encryption of b^* for Id^*. Upon receiving $\mathcal{A}$'s guess, $\mathcal{B}$ end the simulation and outputs its answer to LWE:

- If $\mathcal{A}$ guesses "good", $\mathcal{B}$ answers "pseudo-random".
- If $\mathcal{A}$ guesses "bad", $\mathcal{B}$ answers "random".

By Theorem 8, the distribution of the public parameters and private key responses is indistinguishable from that in the main system. By a standard argument, if $\mathcal{A}$ has advantage $\epsilon \geq 0$ in the above game then $\mathcal{B}$ has advantage $\epsilon/2$ in the LWE decisional problem. □