Specialization and Subordination from Sampling: Lattice-based A(H)IBE in the Plain Model

Shweta Agrawal [⋆] and Xavier Boyen [⋆⋆]

Preliminary version — July 19, 2009

Abstract. We construct an Identity-Based Encryption (IBE) system without random oracles from hard problems on lattices. The system provides full ciphertext anonymity, and also extends to HIBE by properly randomizing lattice trapdoors used as private keys during delegation.

1 Introduction

An Identity Based Encryption (IBE) system [24, 7] is a public key system where the public key can be an arbitrary string such as an email address. A central authority, called a PKG, uses a master key to issue private keys to identities that request them. Hierarchical IBE (HIBE) [19, 18] is a generalization of IBE that mirrors an organizational hierarchy. An identity at level k of the hierarchy tree can issue private keys to its descendant identities, but cannot decrypt messages intended for other identities.

There are currently three classes of IBE systems: (1) based on groups with a bilinear map [7, 4, 5, 25, 16] (to name a few), (2) based on quadratic residuosity modulo a composite [14, 9, 15], and (3) based on hard problems on lattices [17]. To date, the only constructions without random oracles were based on bilinear maps.

In this paper we present an IBE construction based on hard problems in lattices without relying on random oracles. In fact, the construction is anonymous [6, 1, 10] which means that the ciphertext does not reveal the recipient's identity. Anonymous IBE is used for searching on encrypted data. Our construction generalizes to an HIBE by properly randomizing private keys during delegation. Similar results were obtained independently by Peikert [22] and Cash, Hofheinz, and Kiltz [13].

2 Preliminaries: IBE and HIBE

Recall that an Identity-Based Encryption system (IBE) consists of four algorithms [24, 7]: *Setup*, *Extract*, *Encrypt*, *Decrypt*. The *Setup* algorithm generates system parameters, denoted by *params*, and a master key *mk*. The *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 [19, 18], 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 $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell)$ at depth ℓ and the private key $d_{\mathsf{ID}|\ell-1}$ of the parent identity $\mathsf{ID}_{|\ell-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{\ell-1})$

[⋆] *University of Texas at Austin* — `shweta.a@gmail.com` — Supported by DARPA IAMANET.
[⋆⋆] *Stanford University* — `xb@cs.stanford.edu` — Supported by DARPA IAMANET.

at depth $\ell - 1 > 0$. It outputs the private key d_{ID} for identity ID. For convenience, we sometimes refer to the master key as the private key at depth 0, given which the algorithm *Derive* performs the same function as *Extract*. We note that an IBE system is an HIBE where all identities are at depth 1. 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 [7, 8] defines the indistinguishability of ciphertexts under an adaptive chosen-ciphertext and chosen-identity attack (IND-ID-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 [11, 12] 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. Indistinguishability of ciphertexts against chosen-plaintext attacks is also referred to as semantic security.

Security Game. We define IBE and HIBE semantic security under a selective-identity attack (for a hierarchy of maximum depth L) using the following game between a challenger and an adversary:

Init: The adversary outputs an identity $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$ where it wishes to be challenged.

Setup: The challenger runs the *Setup* algorithm giving it the maximum depth L as input (where $L = 1$ for IBE). It gives the adversary the resulting system parameters *params*. 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 $\mathsf{ID}_i = (\mathrm{I}_1, \ldots, \mathrm{I}_u)$ for some $u \leq \ell$. We require that ID_i is not a prefix of ID^*, (i.e., it is not the case that $u \leq k$ and $\mathrm{I}_i = \mathrm{I}_i^*$ for all $i = 1, \ldots, u$). The challenger responds by running algorithm *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 two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ on which it wishes to be challenged. The challenger picks a random bit $b \in \{0, 1\}$ and sets the challenge ciphertext to $C = \mathit{Encrypt}(\mathit{params}, \mathsf{ID}^*, M_b)$. 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 $b' \in \{0, 1\}$. The adversary wins if $b = b'$.

We refer to such an adversary $\mathcal{A}$ as an IND-sID-CPA adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking an HIBE scheme $\mathcal{E} = (\mathit{Setup}, \mathit{Extract}, \mathit{Derive}, \mathit{Encrypt}, \mathit{Decrypt})$, or an IBE scheme $\mathcal{E} = (\mathit{Setup}, \mathit{Extract}, \mathit{Encrypt}, \mathit{Decrypt})$, as

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}} = \left|\Pr[b = b'] - 1/2\right|$$

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

Definition 1. *We say that an IBE or HIBE system $\mathcal{E}$ is $(t, q_{\mathsf{ID}}, \epsilon)$-secure against a selective-identity, adaptive chosen-plaintext attack, if, for all* IND-sID-CPA *adversary $\mathcal{A}$ that runs in time t and makes at most q_{ID} chosen private-key extraction queries, we have that $Adv_{\mathcal{E},\mathcal{A}} < \epsilon$. We abbreviate this by saying that $\mathcal{E}$ is $(t, q_{\mathsf{ID}}, \epsilon)$-*IND-sID-CPA *secure.*

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

In the sequel, our main focus will be to construct (H)IBE systems in the selective security model. We shall however briefly discuss some ways to attain adaptive-ID security.

Ciphertext Anonymity. We shall also briefly discuss the orthogonal privacy notion of IBE ciphertext anonymity (under chosen-plaintext attacks) in a later section. In short, ciphertext anonymity requires that the intended recipient of a ciphertext not transpire from the ciphertext to whom lacks the decryption key.

2.1 Preliminaries: Hard Lattices for Cryptography

Lattices: A n-dimensional lattice in $\mathbb{R}^m$, for $n \leq m$, is a periodic subset of $\mathbb{R}^n$. Formally, we define a lattice and its dual as follows:

Definition 1. Given n linearly independent vectors $b_1, b_2, \ldots b_n \in \mathbb{R}^m$, the lattice Λ generated by them is denoted $\mathcal{L}(b_1, b_2, \ldots, b_n)$ and defined as:

$$\mathcal{L}(b_1, b_2, \ldots, b_n) = \left\{ \sum x_i b_i | x_i \in \mathbb{Z} \right\}$$

The vectors $b_1, \ldots, b_n$ are called the basis of the lattice.
For a lattice Λ, its dual Λ^* is defined as: $\Lambda^* = \{x \in \mathbb{Z}^n | \forall y \in \Lambda, \langle x, y \rangle \in \mathbb{Z}\}$.

Gaussians on Lattices: Recently a lot of progress in lattice based cryptography has used gaussians on lattices. Here, we provide a brief introduction. We refer the interested reader to [17] for more details.

For any $\sigma > 0$ the Gaussian function on $\mathbb{R}^n$ with center c and standard deviation σ is defined as:

$$\forall x \in \mathbb{R}^n, \rho_{\sigma,c}(x) = \exp(-\pi ||x - c||^2/\sigma^2)$$

For any $c \in \mathbb{R}^n$, and an n dimensional lattice Λ, the **discrete gaussian distribution over Λ** is defined as:

$$\forall x \in \Lambda, D_{\Lambda,\sigma,c}(x) = \frac{\rho_{\sigma,c}(x)}{\rho_{\sigma,c}(\Lambda)}$$

The denominator should be viewed as a normalizing factor. For notational convenience, we write $D_{\Lambda,\sigma}$ when $c = 0$.

The **smoothing parameter** of a lattice is defined as:

Definition 2. [20] For any n-dimensional lattice Λ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest real $\sigma > 0$ such that $\sum_{0 \neq x \in \Lambda^*} \rho_{1/\sigma,0}(x) \leq \epsilon$.

Informally, the smoothing parameter is the amount of Gaussian noise σ one must add to a lattice in order to make the distribution uniform. We refer to [20] for the calculation of $\eta_\epsilon(\Lambda)$.

Hard Random Lattices Two families of random lattices that have appeared in previous work and that we will use are defined below. These families are generated from matrices in the following way. Let $A \in \mathbb{Z}_q^{n \times m}$ for some positive n, m, q. Then define:

$$\Lambda^\perp(A) = \{e \in \mathbb{Z}^m : Ae = 0 \mod q\}$$

$$\Lambda(A) = \{y \in \mathbb{Z}^m : y = A^T s \mod q, \text{for some } s \in \mathbb{Z}^n\}$$

It is easy to see that correctly scaled versions of Λ and $\Lambda^\perp$ are duals [17].

When the matrix A is picked uniformly, solving SVP on $\Lambda^\perp(A)$ is as hard as approximating certain problems on **any** lattice of dimension n to within $poly(n)$ factors [2]. The approximation factors were later improved to $\tilde{O}(n)$ in [23].

2.2 The LWE Hardness Assumption

We recall the "Learning With Errors" or LWE hardness assumption, adapted from a description originally given by Regev [23]. The assumption is stated with respect to a Gaussian error distributions χ; we refer to [23] for its parameterization.

Definition 3. For a dimension parameter $n \in \mathbb{N}$, a positive integer $q = q(n) > 2$, and a secret vector $s \in \mathbb{Z}_q^n$, denote by $\mathfrak{A}_{s,\chi}$ the distribution of the variable $(a, a^T s + x)$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, where the vector $a \in \mathbb{Z}_q^n$ is uniform and the scalar $x \in \mathbb{Z}_q$ is sampled from χ [23].

The (average-case) LWE decision problem is to distinguish, with non-negligible probability, between the distribution $\mathfrak{A}_{s,\chi}$ for some random secret $s \in \mathbb{Z}_q^n$ and the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, given oracle access to samples from the given distribution.

As evidence that this is a hard problem, Regev [23] has shown that, for suitable prime moduli q and Gaussian error distributions χ, the decisional $LWE_{q,\chi}$ problem is as hard as solving the classic worst-case lattice problems SIVP and GapSVP in the ℓ_2 norm, using a quantum algorithm. Peikert [21] subsequently extended Regev's result to hold for SIVP and GapSVP in all ℓ_p norms for $2 \leq p \leq \infty$ with essentially the same approximation factors.

Let $\mathbb{R}/\mathbb{Z}$ be the group $[0, 1)$ with real addition modulo 1. For $\alpha \in \mathbb{R}^+$, let Ψ_α be the distribution on $\mathbb{R}/\mathbb{Z}$ of a Gaussian variable with mean 0 and standard deviation $\alpha/2\pi$, reduced modulo 1. Regev's theorem is stated as follows:

Theorem 1 *[23] Let then $\alpha = \alpha(n) \in (0, 1)$. Let $q = q(n)$ be a prime such that $\alpha q > 2n$. If there exists an efficient (possibly quantum) algorithm that solves LWE_{q,Ψ_α}, then there exists an efficient quantum algorithm for approximating SIVP and GapSVP in ℓ_2 norm, in the worst case, to within $O(n/\alpha)$ factors.*

2.3 The Dual of Regev's Public-Key Cryptosystem

Under the LWE assumption, it is easy to construct a PKE cryptosystem based on the indistinguishability of the pseudo-random vector $(a, a^T\, s{+}x)$ from random. Specifically, the pseudo-random element $a^T\, s{+}x \in \mathbb{Z}_q$ can be used to mask one bit of plaintext. This is the Regev PKE cryptosystem.

In the original Regev system [23], the public keys are points close to lattice points, and thus exponentially sparse in the space, whereas by contrast ciphertexts are uniform in the space. This poses a problem when we seek to construct an IBE system, because it is not obvious how to map identities to valid public keys in the Regev systems. Gentry, Peikert, and Vaikuntanathan [17] propose a simple solution for their IBE cryptosystem, which consists in taking the "dual" of Regev's PKE system, essentially swapping public keys and ciphertexts.

The DualRegev PKE system is as follows:

DualKeyGen: Fix a random matrix $A \in \mathbb{Z}^{n\times m}$ for $m \geq 2n\lg q$. Let $f_A : \mathbb{Z}^n \to \mathbb{Z}^m : e \mapsto A\, e \mod q$. Choose an error vector $e \leftarrow D_{\mathbb{Z}^m,r}$ and compute its syndrome $u = f_A(e) = A\, e \mod q$. The secret key is the vector $e \in \mathbb{Z}_q^n$. The public key is the vector $u \in \mathbb{Z}_q^m$.

DualEncrypt: To encrypt a bit $b \in \{0, 1\}$, draw at random a uniform ephemeral $s \in \mathbb{Z}_q^n$, a noise scalar $x \leftarrow \chi$, and a noise vector $y \leftarrow \chi^m$, and output $C = (c_0, c_1)$ where:

$$c_0 = u^T\, s + x + b \cdot \lfloor \frac{q}{2} \rfloor$$
$$c_1 = A^T\, s + y$$

DualDecrypt: To decrypt a ciphertext $C = (c_0, c_1)$ using the secret key e with respect to the public matrix A: compute $b = c_0 - e^T\, c_1 \in \mathbb{Z}_q$; output 1 if b is closer to $\lfloor \frac{q}{2} \rfloor$ than to 0 modulo q; otherwise output 0.

The authors of [17] use this DualRegev PKE as their main building block to construct an IBE system, using a hash function modeled as a random oracle to map identities to public keys.

2.4 The GPV Pre-Image Samplable Function Family

The authors of [17] define and construct pre-image samplable functions. For completeness, and because we build on it, we briefly review their construction here.

Definition 4. [17] A collection of one-way preimage samplable functions consist of three PPT algorithms $\mathsf{TrapGen}, \mathsf{SampleDom}, \mathsf{SamplePre}$ with the following functionality:

- $\mathsf{TrapGen}(1^\lambda)$: On input parameter λ (expressed in unary), it outputs a tuple (A, T), where A is the description of an efficiently computable function $f_A : D_\lambda \to R_\lambda$ (for some efficiently recognizable domain D_λ and range R_λ), and T is a trapdoor for the function f_A. We will assume A is an implicit parameter in the remaining algorithms.
- $\mathsf{SampleDom}(A)$: On input a function description A, it samples an element x from some distribution χ over the domain D_λ, such that the distribution of $f_A(x)$ is uniform over the range R_λ, and outputs x.
- $\mathsf{SamplePre}(T, y)$: On input a trapdoor description T, and a point $y \in R_\lambda$, it samples an element $x \in D_\lambda$ from the same distribution χ as in $\mathsf{SampleDom}$ conditioned on the event that $f_A(x) = y$, and outputs x.

Correctness. For consistency, the GPV hash functions must satisfy the following:

Correct distributions: SampleDom samples an element x from some distribution χ over the domain D_λ, such that the distribution of $f_A(x)$ is uniform over the range R_λ. SamplePre samples an element $x \in D_\lambda$ from distribution χ (same as in SampleDom) conditioned on $f_A(x) = y$.

Security. For security, the GPV preimage-samplable functions must also satisfy:

1. One-wayness without trapdoor: For any PPT algorithm $\mathcal{A}$, the probability that $\mathcal{A}(1^\lambda, A, y) \in {f_A}^{-1}(y) \subset D_\lambda$ is negligible, where the probability is taken over the choice of A, the target value $y \in R_\lambda$ chosen uniformly at random, and $\mathcal{A}$'s random coins.
2. Preimage min-entropy: For every $y \in R_\lambda$, the conditional min-entropy of $x \leftarrow \mathsf{SampleDom}(A)$ given $f_A(x) = y$ is at least $\omega(\lg \lambda)$.
3. Collision resistance without trapdoor: For any PPT algorithm $\mathcal{A}$, the probability that $\mathcal{A}(1^\lambda, A)$ outputs distinct $x, x' \in D_\lambda$ such that $f_A(x) = f_A(x')$ is negligible, where the probability is taken over choice of A and the random coins consumed by $\mathcal{A}$.

Construction. The GPV trapdoor construction is based on the following result by Ajtai [3] (in this theorem, the value of n serves as a substitute for the security parameter λ):

Theorem 2 *[3] For any prime $q = poly(n)$ and any $m \geq 5n \lg q$, there is a probabilistic polynomial-time algorithm that, on input 1^n, outputs a matrix $A \in \mathbb{Z}_q^{n\times m}$ and a full-rank set $S \subset \Lambda^\perp(A)$, where the distribution of A is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ and the length $||S|| \leq L = m^{2.5}$.*

Note that this set S can be converted efficiently to a basis T of $\Lambda^\perp(A)$ such that $||\tilde{T}|| \leq ||\tilde{S}|| \leq L$, and that can thus serve as a trapdoor for a GPV function.

We will also need the following algorithm constructed by the authors of [17] (they call it SampleD in their work, but we shall use SampleGaussian to avoid confusion with SampleDom):

– $\mathsf{SampleGaussian}(B, \sigma, c)$: This algorithm uses an arbitrary lattice basis B to sample efficiently from the discrete Gaussian distribution of center c and deviation σ over the lattice $\Lambda = \mathcal{L}(B)$, for sufficiently large σ, as formally stated in the following theorem:

Theorem 3 *[17] There is a probabilistic polynomial-time algorithm that, given a basis B of an m-dimensional lattice $\Lambda = \mathcal{L}(B)$, a parameter $\sigma \geq ||\tilde{B}|| \cdot \omega(\sqrt{\lg m})$, and a center $c \in \mathbb{R}^m$, outputs a sample from a distribution that is statistically close to $D_{\Lambda,\sigma,c}$.*

The GPV preimage samplable functions are constructed as follows. For a security parameter $\lambda \in \mathbb{N}$, let $n = \Theta(\lambda)$ and q, m, L be as in Theorem 2. The construction takes the Gaussian smoothing parameter $\sigma \geq L \cdot \omega(\sqrt{\lg m})$ as a parameter, and is as follows:

TrapGen$(1^\lambda, \sigma)$**:** Use the algorithm from theorem 2 to choose matrix $A \in \mathbb{Z}_q^{n\times m}$ and short "trapdoor" basis $T \in \Lambda^\perp(A)$. Let $D_\lambda = \{e \in \mathbb{Z}^m : ||e|| \leq \sigma\sqrt{m}\}$ and $R_\lambda = \mathbb{Z}_q^n$ and $f_A : D_\lambda \to R_\lambda$, such that $f_A(e) = A\,e \mod q$. Output (A, T).

SampleDom(A, σ)**:** Let B_z be the standard basis for $\mathbb{Z}^m$ (where m is the output dimension of A). Use the algorithm $\mathsf{SampleGaussian}(B_z, \sigma, 0)$ to sample from distribution $D_{\mathbb{Z}^m,\sigma}$.

SamplePre(T, σ, y)**:** First choose via linear algebra an arbitrary $k \in \mathbb{Z}^m$ such that $A\,k = y \mod q$. Note that this is not difficult since we do not impose a "smallness" constraint on k, and that such k exists for all but at most q^{-n} fraction of A by Lemma 5.1 in [17]. Then, use the algorithm SampleGaussian$(T, \sigma, -k)$ to sample v from distribution $D_{\Lambda^{\perp}(A),\sigma,-k}$. Note that since $\sigma \geq ||\tilde{T}|| \cdot \omega(\sqrt{\lg m})$ by design, the preconditions of the algorithm are satisfied and it can be applied. By Lemma 2.9 in [17], v lands in the domain D_λ with high probability. Output $e = v + k$.

Proving Correctness. We need to prove the following property:

0. Correctness of Distributions: Let $\chi = D_{\mathbb{Z}^m,\sigma}$. To show that the distributions line up correctly, the authors of [17] appeal to the following theorem:

Theorem 4 *[17] Assume the columns of $A \in \mathbb{Z}_q^{n\times m}$ generate $\mathbb{Z}_q^n$, and let $\epsilon \in (0, \frac{1}{2})$ and $\sigma \geq \eta_\epsilon(\Lambda^{\perp}(A))$. Then for $e \sim D_{\mathbb{Z}^m,\sigma}$, the distribution of the syndrome $u = A\,e \mod q$ is within statistical distance 2ϵ of uniform over $\mathbb{Z}_q^n$.*

Furthermore, fix $u \in \mathbb{Z}_q^n$ and let $k \in \mathbb{Z}^m$ be an arbitrary solution to $Ak = u \mod q$. Then the conditional distribution of $e \sim D_{\mathbb{Z}^m,\sigma}$ given $A\,e = u \mod q$ is exactly $k + D_{\Lambda^{\perp}(A),\sigma,-k}$.

We can now easily prove Property 0 above using Theorem 4. We need:

– $f_A(e) = A\,e \mod q$ is statistically close to uniform over the range $R_\lambda = \mathbb{Z}_q^n$. Note that by selection of A and Lemma 5.1 in [17], the columns of A generate $\mathbb{Z}_q^n$ with probability $(1 - q^{-n})$. Also since $\sigma \geq L \cdot \omega\sqrt{\log m}$, $||T|| \leq L$, and hence by Lemma 3.1 in [17], we have $\sigma \geq \eta_\epsilon(\Lambda^{\perp}(A))$ as desired. This, along with part 1 of Theorem 4 gives us the desired property.
– Output of SamplePre: $(v + k)$ is distributed as $D_{\mathbb{Z}^m,\sigma}$ conditioned on $A(v + k) = y \mod q$. The second part of theorem 4 gives this property.

Proving Security. The GPV functions are one-way and collision-resistant as shown below:

1. One-wayness without trapdoor: Inverting a random function f_A on a uniform output $u \in R_n$ is syntactically equivalent to solving the "inhomogeneous small integer solution" problem $ISIS_{q,m,\sigma\sqrt{m}}$. See [17] for more details. [1]
2. Preimage min-entropy: Since preimages are distributed according to a discrete Gaussian per Theorem 4, and since a discrete Gaussian has min-entropy at least $m - 1$ as shown, e.g., in Lemma 2.10 in [17], the preimage min-entropy is at least $m - 1$.
3. Collision-resistance without trapdoor: A collision $e, e' \in D_\lambda$ for f_A implies $A(e - e') = 0 \mod q$ which implies solving $SIS_{q,m,2\sigma\sqrt{m}}$. [2]

3 A Selectively Secure IBE without Random Oracles

Representation of Identities. The following construction assumes that identities id are arbitrary strings in $\{0, 1\}^k$ for some $k = \Theta(\lambda)$, for the given value λ of the security parameter.

[1] For our purposes, our usage of the GPV trapdoor does not rest on ISIS but on the LWE decisional assumption.
[2] Similarly, our need for collision resistance in the GPV trapdoor shall not rest on SIS but on decisional LWE.

3.1 Basic Construction

Setup(1^λ)**:** Fix a suitable prime modulus q and smoothing parameter σ as a function of λ. Choose a random matrix $A \in \mathbb{Z}_q^{n \times m}$, along with a short basis for $\Lambda^\perp(A)$, say T_A, using Ajtai's construction from Theorem 2. We will denote by $f_A : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ the function defined by $f_A(e) = A\,e \mod q$. Also choose a random vector $u_0 \in \mathbb{Z}_q^n$. Additionally, for each $i = 1, ..., k$ and each $b = 0, 1$, choose a random matrix $H_{i,b} \in \mathbb{Z}_q^{n \times l}$, where $l = m$. Let $\bar{H} = \{(i, b, H_{i,b}) : 1 \leq i \leq k, 0 \leq b \leq 1\}$ denote the ordered set of all the $H_{i,b}$.
The IBE public parameters are the matrix A, the vector u_0, and the matrices in $\bar{H}$.
The IBE master secret is the trapdoor T_A.

Extract$(A, u_0, \bar{H}, \mathsf{id}, T_A)$**:** To extract a decryption key corresponding to the identity $\mathsf{id} \in \{0,1\}^k$ using the master secret T_A:

1. For each $i = 1, ..., k$, let $b_i = \mathrm{bit}_i(\mathsf{id})$ be the i-th bit of id. Assemble the $n \times kl$ matrix $H_{\mathsf{id}} = [H_{1,b_1}|...|H_{k,b_k}] \in \mathbb{Z}_q^{n \times kl}$ as the concatenation of k matrices, where the i-th matrix is thus taken from $\bar{H}$ as either $H_{i,0}$ or $H_{i,1}$ according to b_i.
2. For each $i = 1, ..., k$, sample $r_i \in \mathbb{Z}_q^l$ by running $\mathsf{SampleDom}(H_{i,b_i}, \sigma)$. Let $r \in \mathbb{Z}_q^{kl}$ be the concatenated vector of all the samples, i.e., such that $r^T = [r_1^T|...|r_k^T]$.
3. Let $u = u_0 + H_{\mathsf{id}}\, r \in \mathbb{Z}_q^n$. Equivalently, let $u = u_0 + \sum_{i=1}^k H_{i,b_i}\, r_i$.
4. Apply the $\mathsf{SamplePre}(T_A, \sigma, u)$ function using trapdoor T_A to compute the preimage, say e, of u under the function $f_A(\cdot)$. This amounts to finding an $e \in \mathbb{Z}_q^m$ such that $u = A\,e \in \mathbb{Z}_q^n$ and such that e has discrete Gaussian distribution $D_{\mathbb{Z}^m,\sigma}$ conditioned on $u = A\,e \mod q$.
5. Output the identity-based private key $K = (e, r)$.

Encrypt$(A, u_0, \bar{H}, \mathsf{id}, b)$**:** To encrypt a bit b for recipient identity $\mathsf{id} \in \{0,1\}^k$:

1. Let $H_{\mathsf{id}} = [H_{1,b_1}|...|H_{k,b_k}] \in \mathbb{Z}_q^{n \times kl}$, where $b_i = \mathrm{bit}_i(\mathsf{id})$ is the i-th bit of the query id.
2. Choose a uniformly random vector $s \in \mathbb{Z}_q^n$.
3. Draw a scalar $x \in \mathbb{Z}_q$ and two vectors $y = (y_1, ..., y_m) \in \mathbb{Z}_q^m$ and $z = (z_1, ..., z_{kl}) \in \mathbb{Z}_q^{kl}$, all respectively sampled from the "noise" distributions χ, χ^m, and χ^{kl}, parameterized as in the Regev public-key cryptosystem.
4. Let $c_0 = u_0^T\, s + x + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$.
5. Let $c_1 = A^T\, s + y \in \mathbb{Z}_q^m$.
6. Let $c_2 = H_{\mathsf{id}}^T\, s + z \in \mathbb{Z}_q^{kl}$.
7. Output the ciphertext $C = (c_0, c_1, c_2)$.

Decrypt$(A, u_0, \bar{H}, \mathsf{id}, K, C)$**:** To decrypt a ciphertext $C = (c_0, c_1, c_2) \in \mathbb{Z}_q^{1+m+kl}$ given a private key $K = (e, r) \in \mathbb{Z}_q^{m+kl}$:

1. Compute $v = c_0 - e^T\, c_1 + r^T\, c_2 \in \mathbb{Z}_q$.
2. Compare v and $\lfloor \frac{q}{2} \rfloor$ in $\mathbb{Z}$.
3. If they are close, i.e., if the difference $|v - \lfloor \frac{q}{2} \rfloor| \leq \frac{q}{4}$, output 1. Otherwise, output 0.

It is easy to show that the $\mathsf{Decrypt}$ algorithm will give the correct answer with overwhelming probability, if its inputs are generated according to the protocol and the noise parameters are selected as indicated in [23] or [17]. See those references for an elementary proof.

Multi-bit Encryption. The preceding scheme is rather inefficient, requiring $\Theta(k\,l\,n)$ integers mod q to encrypt a single bit of information. One immediate optimization consists of reusing most of the encryption header to encrypt multiple bits with limited additional overhead. Specifically, the same secret ephemeral vector s can be used to encrypt multiple bits b. In this manner, the ciphertext components c_1 and c_2 remain the same, whereas as many components c_0 are sent as there are bits to encrypt. Notice that each c_0 is a single integer modulo q and thus fairly compact.

3.2 Security Reduction

We show the security of the scheme by using the game-hopping proof technique. We shall construct a sequence of games, whose initial game, Γ_0, is the real attack, and whose terminal game, here Γ_4, will be "unwinnable" by the adversary, in the sense that the adversary will be given an obviously useless challenge and thus cannot do better than to make an uneducated random guess. Each transition from Γ_i to Γ_{i+1} will be shown indistinguishable up to a negligible error under some hardness assumption. As long as the number of games is polynomially bounded (here, constant), and each transition distinguishable only with negligible advantage, we will be able to conclude that the adversary's advantage in a real attack is negligible, under the stated assumption(s). Our proof is set in the standard model (in particular, without random oracles, ideal ciphers, or generic groups).

Game Descriptions We first describe the sequence of games without worrying about the indistinguishability of the transitions.

Game Γ_0. This is the honest indistinguishability game under a selective-identity chosen-plaintext attack, or IND-sID-CPA, between an adversary $\mathcal{A}$. and a challenger $\mathcal{B}$.

Recall that, in a selective-identity attack, $\mathcal{A}$ informs $\mathcal{B}$ of the target identity $\mathsf{id}^\dagger$ it intents to attack, before $\mathcal{B}$ runs the Setup algorithm and gives the public parameters to $\mathcal{A}$.

Game Γ_1. This game is identical to Γ_0, except that, in the Setup phase, the challenger $\mathcal{B}$ generates the matrices $H_{i,b} \in \mathbb{Z}_q^{n\times l}$ for $i = 1, ..., k$ and $b = 0, 1$ not directly, but as the public keys of random GPV trapdoor functions with corresponding trapdoors $T_{i,b}$ (using the algorithm in Theorem 2). Observe that by this process the $2k$ matrices $H_{i,b}$ in $\bar{H}$ are still independently and uniformly distributed in $\mathbb{Z}_q^{n\times l}$.

Game Γ_2. This game is identical to Γ_1, except that the challenger $\mathcal{B}$ does not use the master secret T_A nor the $\mathsf{Extract}$ procedure to answer identity-based private-key queries. Rather, it uses a new procedure $\mathsf{TrapdoorExtract}$ and its knowledge of the trapdoors $T_{i,b}$ for $1 \le i \le k$ and $0 \le b \le 1$. These trapdoors are collected in the ordered set $\bar{T} = \{(i, b, T_{i,b}) : 1 \le i \le k, 0 \le b \le 1\}$.

$\mathsf{TrapdoorExtract}$ requires not all of $\bar{T}$ but only one of its trapdoors, say T_{i^*,b^*}, for an *arbitrary* choice of index $i^* \in \{1, ..., k\}$, where $b^* = \mathrm{bit}_{i^*}(\mathsf{id}) \in \{0, 1\}$ is the i^*-th bit of the query identity id. The procedure is as follows:

$\mathsf{TrapdoorExtract}$$(A, u_0, \bar{H}, \mathsf{id}, i^*, T_{i^*,b^*})$: To extract a decryption key corresponding to the identity id, using the hash trapdoor T_{i^*,b^*} of index $i^* \in \{1, ..., m\}$ and bit $b^* = \mathrm{bit}_{i^*}(\mathsf{id})$:

1. For each $i = 1, ..., k$, let $b_i = \mathrm{bit}_i(\mathsf{id})$ be the i-th bit of id. Assemble $H_{\mathsf{id}} = [H_{1,b_1}|...|H_{k,b_k}] \in \mathbb{Z}_q^{n\times kl}$ as the concatenation of k matrices, whose i-th matrix is thus taken from $\bar{H}$ as either $H_{i,0}$ or $H_{i,1}$ according to b_i.

2. For each $i \in \{1, ..., k\} \setminus \{i^*\}$, sample $r_i \in \mathbb{Z}_q^l$ by running $\mathsf{SampleDom}(H_{i,b_i}, \sigma)$, i.e., sampling from $D_{\mathbb{Z}^l,\sigma}$.
3. Let $\tilde{u} = u_0 + \sum_{i \in \{1,...,k\} \setminus \{i^*\}} H_{i,b_i}\, r_i$. In other words, $\tilde{u} = u_0 + H_{\mathsf{id}}\, \tilde{r} \in \mathbb{Z}_q^n$ where $\tilde{r}$ is the concatenation of all the r_i except that 0 is substituted for r_{i^*}.
4. Sample $e \in \mathbb{Z}_q^m$ from the distribution $D_{\mathbb{Z}^m,\sigma}$ by applying algorithm $\mathsf{SampleDom}(A, \sigma)$, i.e., sampling from $D_{\mathbb{Z}^m,\sigma}$.
5. Compute $u = A\, e \in \mathbb{Z}_q^n$. Let also $\hat{u} = u - \tilde{u}$.
6. Using the trapdoor T_{i^*,b^*}, apply algorithm $\mathsf{SamplePre}(T_{i^*,b^*}, \sigma, \hat{u})$ to sample a suitably distributed preimage $r_{i^*} \in \mathbb{Z}_q^l$ such that $\hat{u} = H_{i^*,b^*}\, r_{i^*}$.
7. Let r be the concatenation of all the r_i for $i = 1, ..., k$, including r_{i^*}. Observe that, by construction, $u = u_0 + H_{\mathsf{id}}\, r$.
8. Output the identity-based private key $K = (e, r)$.

Game $\varGamma_3$. This game is identical to $\varGamma_2$, except in the way the challenger $\mathcal{B}$ computes the hash matrices $\bar{H}$ and their trapdoors $\bar{T}$. It will proceed so that it only knows the trapdoor for the matrix of index i that does not correspond to the i-th bit of the target identity $\mathsf{id}^\dagger$. Precisely:

– For $i = 1, ..., k$, let $\mathrm{bit}_i(\mathsf{id}^\dagger)$ be the i-th bit of the target identity $\mathsf{id}^\dagger$, revealed by $\mathcal{A}$ to $\mathcal{B}$ before Setup.
– In the Setup phase, $\mathcal{B}$ now generates $\hat{H}$ as follows.
  • For each $i = 1, ..., k$ and $b \in \{0, 1\}$ such that $b \neq \mathrm{bit}_i(\mathsf{id}^\dagger)$, it executes the GPV generation function as in game $\varGamma_2$ to obtain both a random hash matrix $H_{i,b}$ and its trapdoor $T_{i,b}$.
  • For each $i = 1, ..., k$ and $b \in \{0, 1\}$ such that $b = \mathrm{bit}_i(\mathsf{id}^\dagger)$, it simply picks a random hash matrix $H_{i,b} \in \mathbb{Z}_q^{n \times l}$ and sets $T_{i,b} = \perp$.

  Let then $\bar{H}$ be the resulting collection of $2k$ hash matrices $H_{i,b}$, and similarly let $\bar{T}$ be the (incomplete) collection of their respective trapdoors $T_{i,b}$.
– To answer private-key extraction queries on any identity $\mathsf{id} \neq \mathsf{id}^\dagger$, the challenger proceeds as in game $\varGamma_2$, except that it is forced to choose an index i^* such that $\mathrm{bit}_{i^*}(\mathsf{id}) \neq \mathrm{bit}_{i^*}(\mathsf{id}^\dagger)$. Such index i^* always exists for a legal query. Let $b^* = \mathrm{bit}_{i^*}(\mathsf{id})$. By construction, $T_{i^*,b^*} \neq \perp$ in $\hat{T}$. The challenger can thus execute $\mathsf{TrapdoorExtract}(A, u_0, \bar{H}, \mathsf{id}, i^*, T_{i^*,b^*})$. It gives the result $K = (e, r)$ to $\mathcal{A}$.

Recall that, up to and including this game, the challenge ciphertext is produced by evaluating $\mathsf{Encrypt}(A, H_0, \hat{H}, \mathsf{id}^\dagger, b^\dagger)$ for a random bit $b^\dagger \in \{0, 1\}$ and outputting the result $C^\dagger = (c_0, c_1, c_2)$. This will change in the following games.

Game $\varGamma_4$. This game is identical to $\varGamma_3$, except that the challenge ciphertext given to $\mathcal{A}$ is no longer created honestly, but completely at random by $\mathcal{B}$. Specifically, in $\varGamma_4$, the challenger no longer evaluates $\mathsf{Encrypt}$ to produce the challenge ciphertext; rather, it draws $C^\dagger = (c_0^\dagger, c_1^\dagger, c_2^\dagger)$ uniformly at random from $\mathbb{Z}_q^{1+m+kl}$.

The view of $\mathcal{A}$ in this last game is thus necessarily independent of the plaintext bit $b^\dagger \in \{0, 1\}$, and hence its advantage at guessing $b^\dagger$ is necessarily zero.

Game Transitions We now show the indistinguishability of each transition between the successive games just described.

From $\mathit{\Gamma}_0$ to $\mathit{\Gamma}_1$. The view of the adversary is identical in both games. The fact that $\mathcal{B}$ knows the trapdoors $T_{i,b}$ to the hash keys $H_{i,b}$ is invisible to $\mathcal{A}$.

From $\mathit{\Gamma}_1$ to $\mathit{\Gamma}_2$. The view of the adversary is identical in both games. The fact that $\mathcal{B}$ uses a different procedure to answer key extraction queries is invisible to $\mathcal{A}$, since the resulting keys have the same distribution.

From $\mathit{\Gamma}_2$ to $\mathit{\Gamma}_3$. The view of the adversary is identical in both games. The fact that $\mathcal{B}$ now only knows half of all hash trapdoors, and is thus forced to use the ones he knows when answering key extractions queries, is again invisible to $\mathcal{A}$.

From $\mathit{\Gamma}_3$ to $\mathit{\Gamma}_4$. The view of the adversary is not identical in both games, but it is indistinguishable under the "Learning With Error" (LWE) hardness assumption. To show this, we consider a reduction from the problem of deciding $1 + m + kl$ samples of LWE to that of distinguishing between games $\mathit{\Gamma}_3$ and $\mathit{\Gamma}_4$. The reduction is as follows:

- At the very beginning, before the game starts, $\mathcal{B}$ receives $1 + m + kl$ samples of the LWE problem, $(a_j, b_j) \in \mathbb{Z}_q^{n+1}$ for $j = 1, ..., 1 + m + kl$, where all $a_j \in \mathbb{Z}_q^n$ are random, and either all $b_j \in \mathbb{Z}_q$ are also random or all are equal to $a_j^T\, s + x_j$ for a (common) uniform secret $s \in \mathbb{Z}_q^n$ and (independent) Gaussian noises x_j drawn from χ. That is, either all b_j are drawn from $\mathfrak{A}_{s,\chi}$ or they are all uniform.
- At the beginning of the game, $\mathcal{B}$ receives from $\mathcal{A}$ the identity $\mathsf{id}^\dagger$ that $\mathcal{A}$ intends to attack.
- In the Setup phase, $\mathcal{B}$ generates $\hat{H}$ as follows.
  - For each $i = 1, ..., k$ and $b \in \{0, 1\}$ such that $b \neq \mathrm{bit}_i(\mathsf{id}^\dagger)$, it executes the GPV generation function as in game $\mathit{\Gamma}_2$ to obtain both a random hash matrix $H_{i,b}$ and its trapdoor $T_{i,b}$, exactly as in games $\mathit{\Gamma}_3$ and $\mathit{\Gamma}_4$.
  - For each $i = 1, ..., k$ and $b \in \{0, 1\}$ such that $b = \mathrm{bit}_i(\mathsf{id}^\dagger)$, it assembles the hash matrix $H_{i,b} \in \mathbb{Z}_q^{n\times l}$ so that the j'-th column of $H_{i,b}$ is copied from the $j = (1 + m + (i - 1)l + j')$-th LWE instance vector a_j. It sets $T_{i,b} = \perp$.

  Again, we let $\bar{H}$ collect all the $H_{i,b}$, and $\bar{T}$ collect the (available) trapdoors $T_{i,b}$.
- To answer private key queries, $\mathcal{B}$ proceeds as in game $\mathit{\Gamma}_3$ or equivalently $\mathit{\Gamma}_4$. It can do this using the available trapdoors.
- To create the challenge ciphertext, $\mathcal{B}$ picks $b^\dagger \in \{0, 1\}$ and sets:

$$\begin{aligned} c_0^\dagger &= b_1 + b\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q \\ c_1^\dagger &= (b_{1+i} : i = 1, ..., m) \in \mathbb{Z}_q^m \\ c_2^\dagger &= (b_{1+m+i} : i = 1, ..., kl) \in \mathbb{Z}_q^{kl} \end{aligned}$$

- At the end of the simulation, when $\mathcal{A}$ outputs a bit $\hat{b}^\dagger$ as its decryption guess, $\mathcal{B}$ returns `genuine` if $\hat{b}^\dagger = b^\dagger$, and `random` if $\hat{b}^\dagger \neq b^\dagger$, as its own answer regarding the LWE instances.

In the view of $\mathcal{A}$, the behavior of $\mathcal{B}$ is identical to both games $\mathit{\Gamma}_3$ and $\mathit{\Gamma}_4$ in all respects excluding the challenge ciphertext. In particular, $\bar{H}$ created using the LWE instances has a uniform distribution whether the LWE instances are genuine or not.

For the challenge ciphertext, it is easy to see that, if the LWE instances are genuine, the components of $C^\dagger$ will have the same distribution as in game $\mathit{\Gamma}_3$; whereas, if the LWE instances are random, so will be the components of $C^\dagger$, as in game $\mathit{\Gamma}_4$. Should $\mathcal{A}$ exhibit a different success probability in either case, $\mathcal{B}$ will have successfully distinguished between $1 + m + kl$ genuine and random instances of the LWE problem.

4 A Hierarchical IBE without Random Oracles

It is easy to extend the previous IBE scheme into a hierarchical HIBE system. To do so, we essentially translate the framework from [4], which amounts to "re-encrypting" the master key for the various identity components as we delegate it down the hierarchy tree. Each time the key is re-encrypted, it becomes more specialized and thus less powerful.

Here, the master key is the trapdoor for the top-level root or "identity-less" matrix (denoted A in the IBE scheme, but here to be renamed H_0 for notational convenience). Hence, we need a mechanism for specializing trapdoors of this form to "identity-specific" matrices H_{id} down the hierarchy. Since in the basic IBE scheme the full identity-specific matrices, from which the ciphertexts are created, are the horizontal concatenation of the root matrix A and a string of identity-dependent sub-matrices H_{i,b_i}, the delegation problem in our case is merely that of finding a way to create a random trapdoor T_{AB} for any given matrix $[A|B]$ given a trapdoor T_A for the matrix A.

This problem has an almost immediate solution for the kind of hard random lattices used in our IBE scheme, as we now show.

4.1 Trapdoor Delegation

Recall that for a lattice defined by a matrix $A \in \mathbb{Z}_q^{n\times m}$, the trapdoor T_A is simply a full-rank set of short vectors $e \in \Lambda^\perp(A)$, which is to say a full-rank matrix T_A of vectors e such that $Ae = 0$.

Hence, given a trapdoor T_A for A, one can construct a random trapdoor $T_{[A|A']}$ for the "wider" concatenated matrix $[A|A']$ very easily: by picking n short random vectors e'_i in the domain of $f_{A'}(\cdot)$, and for each of them finding a short preimage e_i under $f_A(\cdot)$ that will cause mutual "cancellation", i.e., such that $f_A(e_i) + f_{A'}(e'_i) = 0$. The algorithm is as follows:

SubTrapdoor(A, A', T_A)**:** Given two matrices $A \in \mathbb{Z}_q^{n\times m}$ and $A' \in \mathbb{Z}_q^{n\times m'}$, and an arbitrary trapdoor T_A for A, do the following to generate a random trapdoor $T_{[A|A']}$ for $[A|A']$:

```
1. For $i = 1, ..., n$, do:
   (a) Sample a short Gaussian-distributed random vector $e'_i \in \mathbb{Z}_q^{m'}$ from the domain of $B$ using
       the SampleDom algorithm.
   (b) Compute $u_i = A' \, e'_i \in \mathbb{Z}_q^n$.
   (c) Using the trapdoor $T_A$, sample a short Gaussian-distributed random preimage $e_i \in \mathbb{Z}_q^m$
       of $-u_i$ under $f_A(\cdot)$. I.e., find a short vector $e_i$ such that $A\, e_i = -u_i$.
   (d) Assemble the $i$-th trapdoor vector in $T_{[A|A']}$ to be the column vector $t_i = \begin{bmatrix} e_i \\ e'_i \end{bmatrix} \in \mathbb{Z}_q^{m+m'}$.
2. Output the trapdoor matrix $T_{[A|A']} = [t_1|t_2|...|t_n] \in \mathbb{Z}_q^{(m+m')\times n}$.
```

As long as SampleDom produces integer Gaussian samples in $\mathbb{Z}_q^m$ with standard deviation $\sqrt{m}\,\sigma$, and as SamplePre given T_A produces integer Gaussian in $\mathbb{Z}_q^{m'}$ with standard deviation $\sqrt{m'}\,\sigma$, the synthesized vectors t_i will also be integer Gaussian in $\mathbb{Z}_q^{m+m'}$ with standard deviation $\sqrt{m+m'}\,\sigma$.

4.2 Hierarchical Construction

We shall now describe the basic HIBE construction for a hierarchy of identities comprising L levels. An identity id at level $\ell \leq L$ is represented as an ℓ-vector $\mathsf{id} = (id_1, ..., id_\ell) \in \{0,1\}^{\ell k}$ where each identity component $id_l \in \{0,1\}^k$ is a k-bit string. [3]

Setup$(1^\lambda, L)$**:** Fix a suitable prime modulus q, dimensions n and m, and smoothing parameter σ, as a function of the security parameter λ.
Choose a random vector $u_0 \in \mathbb{Z}_q^n$. Invoke Ajtai's construction from Theorem 2 to choose a random matrix $H_0 \in \mathbb{Z}_q^{n\times m}$ along with a trapdoor T_0 which is a short basis for $\Lambda^\perp(H_0)$. We denote by $f_0 : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ the function defined by $f_0 : r_0 \mapsto H_0\, r_0 \mod q$. Moreover, independently select, for each $l = 1, ..., L$, each $i = 1, ..., k$, and each $b = 0, 1$, a random matrix $H_{l,i,b} \in \mathbb{Z}_q^{n\times m}$. Denote by $\bar{H} = \{H_0\} \cup \{(l, i, b, H_{i,b}) : 1 \leq l \leq L, 1 \leq i \leq k, 0 \leq b \leq 1\}$ the ordered set comprising H_0 and all the $H_{l,i,b}$.
The HIBE public parameters are the vector u_0 and the matrices in $\bar{H}$.
The HIBE master secret is the trapdoor T_0 (corresponding to H_0).

Extract$(u_0, \bar{H}, \mathsf{id}, T_0)$**:** To extract an identity-based decryption key corresponding to the ℓ-level identity $\mathsf{id} \in \{0,1\}^{\ell k}$ using the master secret T_0:

1. For each $l = 1, ..., \ell$ and $i = 1, ..., k$, let $b_{l,i} = \mathrm{bit}_i(id_l)$ be the i-th bit of the l-th component of the full identity id.
2. Assemble the $n \times (1 + \ell k)m$ matrix $H_{\mathsf{id}} = [H_0|H_{1,1,b_{1,1}}|...|H_{1,k,b_{1,k}}|...|...|H_{\ell,k,b_{\ell,k}}] \in \mathbb{Z}_q^{n\times \ell km}$ as the concatenation of $1 + \ell k$ matrices chosen from $\bar{H}$ according to the bits of id.
3. Assemble the $n \times \ell km$ matrix H_{id}^Δ in the same manner, but omitting the matrix H_0.
4. Use the $\mathsf{SubTrapdoor}(H_0, H_{\mathsf{id}}^\Delta, T_A)$ algorithm to produce a random trapdoor T_{id} for the lattice defined by the concatenated matrix $H_{\mathsf{id}} = [H_0|H_{\mathsf{id}}^\Delta]$.
5. Output the identity-based private key for id consisting of the subordinate trapdoor T_{id}.

Derive$(\bar{H}, \mathsf{id}, T)$**:** To derive a private key for a delegatee identity $\mathsf{id} = (id_1, ..., id_\ell) \in \{0,1\}^{\ell k}$ given a private key $K = (r, T)$ for a delegator identity $\mathsf{id}' = (id_1, ..., id_{\ell'}) \in \{0,1\}^{\ell' k}$, prefix of id, such that $0 < \ell' < \ell$:

1. For each $l = 1, ..., \ell$ and $i = 1, ..., k$, let $b_{l,i} = \mathrm{bit}_i(id_l)$ be the i-th bit of the l-component of id.
2. Assemble the $n\times(1+\ell k)m$ matrix $H_{\mathsf{id}} = [H_0|H_{1,1,b_{1,1}}|...|H_{1,k,b_{1,k}}|...|...|H_{\ell,k,b_{\ell,k}}] \in \mathbb{Z}_q^{n\times(1+\ell k)m}$ as the concatenation of $1 + \ell k$ matrices including H_0 matching the delegatee's identity id.
3. Build the $n \times (1 + \ell' k)m$ matrix $H'_{\mathsf{id}} = [H_0|H_{1,1,b_{1,1}}|...|H_{1,k,b_{1,k}}|...|...|H_{\ell',k,b_{\ell',k}}] \in \mathbb{Z}_q^{n\times(1+\ell' k)m}$ as the concatenation of the $1 + \ell' k$ matrices corresponding to the delegator's identity id'.
4. Fix the $n \times (\ell - \ell')km$ matrix $H_{\mathsf{id}}^\Delta = [H_{\ell'+1,1,b_{\ell'+1,1}}|...|H_{\ell'+1,k,b_{\ell'+1,k}}|...|...|H_{\ell,k,b_{\ell,k}}] \in \mathbb{Z}_q^{n\times km}$ as the concatenation of the $(\ell - \ell')k$ matrices corresponding to the suffix of id not in id'.

– Notice that $H_{\mathsf{id}} = [H'_{\mathsf{id}}|H_{\mathsf{id}}^\Delta]$ and that the given trapdoor T should apply exactly to H'_{id}.

5. Use the $\mathsf{SubTrapdoor}(H'_{\mathsf{id}}, H_{\mathsf{id}}^\Delta, T)$ algorithm with the trapdoor T to produce a subordinate random trapdoor T_{id} for the concatenated matrix $H_{\mathsf{id}} = [H'_{\mathsf{id}}|H_{\mathsf{id}}^\Delta]$.
6. Output the identity-based private key consisting of the trapdoor T_{id} corresponding to H_{id}.

[3] A word about notation: in the HIBE, the variable l will be used as an iterator over the hierarchy levels. In the basic IBE scheme given in an earlier section, $l = m$ was a dimension parameter; we now use m for l's earlier purpose. Moreover, for notation uniformity, we shall rename the earlier matrix A and its trapdoor T_A respectively by H_0 and T_0, and also substitute for the short private-key vector e a new short vector denoted r_0.

Notice that Extract is a special case of Derive when the delegator identity is the empty string, and could have been specified that way.

Encrypt$(u_0, \bar{H}, \mathsf{id}, b)$**:** To encrypt a bit $b \in \{0,1\}$ for some recipient's full hierarchical identity $\mathsf{id} = (id_1, ..., id_\ell) \in \{0,1\}^{\ell k}$:

1. Let $H_{\mathsf{id}} = [H_0|H_{1,1,b_{1,1}}|...|H_{1,k,b_{1,k}}|...|...|H_{\ell,k,b_{\ell,k}}] \in \mathbb{Z}_q^{n\times(1+\ell k)m}$ where as before $b_{l,i} = \mathrm{bit}_i(id_l)$ is the i-th bit of the l-th component of the recipient identity id.
2. Choose a uniformly random vector $s \in \mathbb{Z}_q^n$.
3. Draw a scalar $x \in \mathbb{Z}_q$, a vector $y = (y_1, ..., y_m) \in \mathbb{Z}_q^m$, and a vector $z = (z_1, ..., z_{\ell km}) \in \mathbb{Z}_q^{\ell km}$, all respectively sampled from the "noise" distributions χ, χ^m, and $\chi^{\ell km}$, parameterized as in the Regev public-key cryptosystem.
4. Let $c_0 = u_0^T\, s + x + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$.
5. Let $c_1 = H_{\mathsf{id}}^T\, s + [y^T|z^T]^T \in \mathbb{Z}_q^{(1+\ell k)m}$.
6. Output the ciphertext $C = (c_0, c_1)$.

Decrypt$(u_0, C, T_{\mathsf{id}})$**:** To decrypt a ciphertext $C = (c_0, c_1) \in \mathbb{Z}_q^{1+((1+\ell k)m)}$ given a trapdoor T_{id} for the lattice defined by the matrix H_{id} corresponding to identity id:

1. Invoke the function SamplePre$(T_{\mathsf{id}}, \sigma, u_0)$ using the trapdoor T_{id}. This produces a short preimage $r \in \mathbb{Z}_q^{(1+\ell k)m}$ of u_0 under the map $f_{\mathsf{id}}(\cdot) : r \mapsto H_{\mathsf{id}}\, r \mod q$. In other words, this samples r from the discrete Gaussian distribution $D_{\mathbb{Z}^{1+\ell k)m},\sigma}$ conditioned on the event $u_0 = H_{\mathsf{id}}\, r \in \mathbb{Z}_q^n$.
2. Compute $v = c_0 - r^T\, c_1 \in \mathbb{Z}_q$.
3. Compare v and $\lfloor \frac{q}{2} \rfloor$ in $\mathbb{Z}$.
4. If they are close, i.e., if the difference $|v - \lfloor \frac{q}{2} \rfloor| \le \frac{q}{4}$, output 1. Otherwise, output 0.

Observe that the first step, that involving SamplePre, can be performed once and for all by the recipient with identity id upon receiving his private-key trapdoor T_{id}. In other words, we can decompose Decrypt into two algorithms, PreDecrypt and PostDecrypt, where the former is expensive but executed once only, and the latter is cheap and executed on every ciphertext needing decryption:

PreDecrypt(u_0, T_{id})**:** To create a permanent decryption key $r_{\mathsf{id}} \in \mathbb{Z}_q^{(1+\ell k)m}$ given a trapdoor T_{id} for the lattice defined by the matrix H_{id} corresponding to identity id:

1. Conpute $r_{\mathsf{id}} \leftarrow$ SamplePre$(T_{\mathsf{id}}, \sigma, u_0)$. We get $r_{\mathsf{id}} \in \mathbb{Z}_q^{(1+\ell k)m}$ such that $H_{\mathsf{id}}\, r = u_0 \in \mathbb{Z}_q^n$.

PostDecrypt$(u_0, C, r_{\mathsf{id}})$**:** To decrypt a ciphertext $C = (c_0, c_1) \in \mathbb{Z}_q^{1+((1+\ell k)m)}$ given a decryption key $r_{\mathsf{id}} \in \mathbb{Z}_q^{(1+\ell k)m}$ previously created by PreDecrypt:

1. Compute $v = c_0 - r_{\mathsf{id}}^T c_1 \in \mathbb{Z}_q$ and compare it to $\lfloor \frac{q}{2} \rfloor$ in $\mathbb{Z}$.
2. If the two are close, i.e., if the difference $|v - \lfloor \frac{q}{2} \rfloor| \le \frac{q}{4}$, output 1. Otherwise, output 0.

We remark finally that the sole purpose of providing private keys of the form T_{id} rather than r_{id} as in the basic IBE scheme, is to enable delegation further down the hierarchy tree. Otherwise, it would be more efficient for the Extract procedure to output r_{id} directly, as in the basic IBE scheme.

4.3 Security Proof

We can show a tight security reduction from the task of deciding the LWE problem on the basis of $(1 + (1 + \ell k)m)$ samples from (or non-interactive oracle to) the LWE oracle, to the advantage over random of guessing the decryption of a random bit in an IND-sHID-CPA attack. The reduction is essentially the same as in the IBE case given earlier. Rather than to give a full proof, we merely note the following changes or additions:

- The adversary $\mathcal{A}$ announces before the Setup step the full hierarchical identity $\mathsf{id}^\dagger$ that it intends to attack.
- The simulator $\mathcal{B}$ will, as in the IBE game, construct the matrices in $\bar{H}$ so that, for each bit of each identity component, it ignores the trapdoor for the bit value that is the same as in the target identity $\mathsf{id}^\dagger$. It also makes sure to ignore the trapdoor for the matrix H_0 (the analogue to A in the basic IBE scheme).
- With this setup, the simulator $\mathcal{B}$ is still able to answer every Extract query for identities id that are neither equal to, or prefix of, the target identity $\mathsf{id}^\dagger$. Indeed, by construction, $\mathcal{B}$ knows at least one trapdoor $T_{l,i,b_{l,i}}$ for some matrix $H_{l,i,b_{l,i}}$ included in the concatenation H_{id}, and based on this it can evaluate $\mathsf{SubTrapdoor}$ and obtain a subordinate trapdoor T_{id} for H_{id}. The subordinate trapdoor will have the same (integer Gaussian) distribution as in the real scheme.
- To construct the challege ciphertext for $\mathsf{id}^\dagger$ in the final game, the simulator proceeds in a completely analogous way as the IBE simulator. The only difference is that $\mathcal{B}$ will need not only $1 + (1 + k)m$ samples from the LWE oracle, but as many $1 + (1 + \ell k)m$ such samples, owing to the increased length of the ciphertext. Observe that the simulator is unable to decrypt the challenge himself, since he will know none of the trapdoors for the matrices $H_{l,i,b_{l,i}}$ that constitute the target identity's encryption matrix $H_{\mathsf{id}^\dagger}$.

5 Further Extensions

We briefly and informally mention a few extensions and properties of note that apply to our systems.

5.1 Ciphertext Anonymity and Indistinguishability from Random

Although the preceding proofs focused on the usual notion of semantic security, we observe that our IBE and HIBE schemes provide the strongest possible notion of privacy: complete indistinguishability of the ciphertext from random (for the basic system: under a selective-identity chosen-plaintext attack, in the computational sense under the LWE assumption).

This very strong notion of indistinguishability subsumes in particular the usual notion of IBE anonymity, denoted ANON-sID-CPA, which requires that it should not be feasible to distinguish a ciphertext created for recipient id_1 from one created for id_2 (without necessarily decrypting either). The notion of IBE and HIBE ciphertext anonymity has many useful applications, e.g., for searching on public-key-encrypted data [6, 1, 10].

5.2 From Selective to Adaptive Security

We briefly discuss two standard methods by which our schemes can be made IND-ID-CPA secure.

Exponential Reduction Very generally, let us say that a cryptographic scheme is (ϵ, τ)-secure for security objective OBJ against adversarial capabilities ADV, if every probabilistic algorithm conducting an attack ADV against the system has a probability of winning under criterion OBJ with probability $\leq \epsilon$ in time $\leq \tau$.

It is well known from [4] that any selective-identity secure IBE scheme $\mathcal{E}$ can be transformed generically into an adaptive-identity secure IBE scheme $\mathcal{E}'$ for the same security objective (e.g., indistinguishability), at the cost of an increase in the attack's success probability proportional to the number N of allowed identities. Specifically:

Theorem 5 *[4] Every IBE system $\mathcal{E}$ with (ϵ, τ)-IND-sID-CPA security can be transformed generically into an IBE system $\mathcal{E}'$ with $(N\epsilon, \tau)$-IND-ID-CPA security, where N is the total number of allowed identities in $\mathcal{E}'$.*

Since N must be exponential for the transformed IBE scheme $\mathcal{E}'$ to be universally useful, this transformation generally introduces an exponential loss of security; but one can easily and cheaply compensate for such loss by increasing the security parameter of the initial scheme $\mathcal{E}$ by an additive constant $\log_2 N$. For general use with hashed identities, it is customary to require $N = 2^n$ or $N = 2^{2n}$, where $n = n_{\mathcal{E}'}$ is the desired security parameter of the final system. In these conditions, the generic transformation from selective to adaptive identity will result for the final system in a security parameter $n_{\mathcal{E}'} = n$ if one sets the initial system's parameter $n_{\mathcal{E}} = 2n$ or $3n$. Because a scheme's security parameter exponentially affects its security, but only polynomially its complexity, the overall cost of the transformation is expected to remain quite small (typically a constant multiplicative factor).

Combinatorial Transformation If one wishes to avoid changing the underlying "number-theoretic context" (i.e., the choices of prime order q, lattice dimension n, vector space dimension m, etc.) when transforming the selective-ID scheme into an adaptive-ID scheme, one can instead use the semi-generic transformation from [5] based on "admissible biased binary hash functions".

The structure of our basic scheme is very well suited to the semi-generic transformation from [5], owing to the construction of $\bar{H}$ as an assembly of left-or-right sub-matrices depending on the value of each bit of the identity. The idea, borrowed from [5], is to expand the identity id using some code with large enough minimum distance, and design a set of trapdoors such that no trapdoor is known for either value 0 or 1 of certain bits of the expanded identities. This way, the simulator will be able to answer private key extraction queries for all identities except a polynomially small fraction of them, and conversely be able to make use of the adversary's response to the challenge ciphertext for any one of those remaining identities. We refer to [5] for more information about this technique. The details of this transformation will be added in a subsequent version.

Per on our current understanding of the hardness of lattice problems, the "exponential reduction" should result in much more efficient fully-secure IBE schemes than the "combinatorial transformation", for identical values of the (resulting) security parameter.

Acknowledgments

The authors wish to thank Dan Boneh for many stimulating discussions on related lattice problems, that were in many ways instrumental to the development of the present work.

References

1. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *Advances in Cryptology—CRYPTO 2005*, LNCS, pages 205–22. Springer-Verlag, 2005.
2. Miklos Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, New York, NY, USA, 1996. ACM.
3. Miklos Ajtai. Generating hard instances of the short basis problem. In *ICALP*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.
4. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238, 2004.
5. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459, 2004.
6. Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–22. Springer-Verlag, 2004.
7. 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.
8. Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.
9. Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *Proceedings of FOCS 2007*, pages 647–657, 2007.
10. 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.
11. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *LNCS*. Springer-Verlag, 2003.
12. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–22. Springer-Verlag, 2004.
13. David Cash, Dennis Hofheinz, and Eike Kiltz. How to delegate a lattice basis. Manuscript, 19 July 2009.
14. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 26–8, 2001.
15. Giovanni Di Crescenzo and Vishal Saraswat. Public key encryption with searchable keywords based on jacobi symbols. In *Proceedings of INDOCRYPT 2007*, pages 282–296, 2007.
16. Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2006*, LNCS. Springer-Verlag, 2006.
17. 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.
18. Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In *Advances in Cryptology—ASIACRYPT 2002*, LNCS. Springer-Verlag, 2002.
19. Jeremy Horwitz and Ben Lynn. Towards hierarchical identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2002*, LNCS, pages 466–81. Springer-Verlag, 2002.
20. 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.
21. Chris Peikert. Limits on the hardness of lattice problems in p norms. In *IEEE Conference on Computational Complexity*, pages 333–346, 2007.
22. Chris Peikert. Bonsai trees (or, arboriculture in lattice-based cryptography). Manuscript, 19 July 2009.
23. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
24. Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology—CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag, 1984.
25. Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*. Springer-Verlag, 2005.