Trapdoors for Hard Lattices
and New Cryptographic Constructions

Craig Gentry*
Stanford University
cgentry@cs.stanford.edu

Chris Peikert†
SRI International
cpeikert@alum.mit.edu

Vinod Vaikuntanathan‡
MIT
vinodv@mit.edu

November 18, 2007

Abstract

We show how to construct a variety of "trapdoor" cryptographic tools assuming the worst-case hardness of standard lattice problems (such as approximating the shortest nonzero vector to within small factors). The applications include trapdoor functions with *preimage sampling*, simple and efficient "hash-and-sign" digital signature schemes, universally composable oblivious transfer, and identity-based encryption.

A core technical component of our constructions is an efficient algorithm that, given a basis of an arbitrary lattice, samples lattice points from a Gaussian-like probability distribution whose standard deviation is essentially the length of the longest vector in the basis. In particular, the crucial security property is that the output distribution of the algorithm is oblivious to the particular geometry of the given basis.

*Supported by the Herbert Kunzel Stanford Graduate Fellowship.

†This material is based upon work supported by the National Science Foundation under Grants CNS-0716786 and CNS-0749931. Any opinions, findings, and conclusions or recommedations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

‡The majority of this work was performed while at SRI International.

1 Introduction

Ever since the seminal work of Ajtai [Ajt96] connecting the *average-case* complexity of lattice problems to their complexity in the *worst case*, there has been an intriguing and fruitful effort to base cryptography (which requires security for *randomly* chosen keys) on worst-case lattice assumptions. In addition to their unique theoretical niche, lattice-based schemes enjoy many advantages: first, their asymptotic efficiency and simplicity (usually requiring only linear operations on small integers); second, their resistance so far to cryptanalysis by *quantum* algorithms (as opposed to schemes based on factoring or discrete log); and third, the guarantee that their random instances are "as hard as possible."

To date, the known constructions of such primitives have been limited mainly to one-way and collision-resistant hash functions [Ajt96, GGH96, CN97, Mic04, MR07] and public-key encryption [AD97, Reg04, Reg05]. Of particular note is the lack of any "direct" construction of *digital signatures* having the simplicity and efficiency of other lattice-based primitives, even in the random oracle model.[1] (An early signature proposal by Goldreich, Goldwasser, and Halevi [GGH97] was directly related to a certain lattice problem, but it lacked a security proof, and recently, Nguyen and Regev [NR06] showed how to completely recover the secret key from a transcript of signatures (on random messages). We discuss the GGH scheme in more detail below.)

There have been some recent developments in the construction of stronger cryptographic tools based on lattices (such as chosen ciphertext-secure encryption [PW07]). But despite many advances in the study of lattices in cryptography, it is still unknown how to realize several notions that were long ago constructed based on other number-theoretic problems, such as those related to factoring.

1.1 Overview of Results and Techniques

Our main thesis in this work is that lattices admit natural and innate "trapdoors" that have a number of useful cryptographic applications. Going at least as far back as the GGH proposal [GGH97], is was intuitively believed that a *short basis* of a lattice could serve as such a trapdoor. Our main contribution is to show how to use a trapdoor basis in a theoretically sound and secure way.

As a basic tool, we first construct a collection of trapdoor functions having some special properties. The functions are *many-to-one*, (i.e., every output value has several preimages), and the trapdoor inversion algorithm *samples* from among all the preimages under an appropriate distribution. Building upon this foundation, we demonstrate efficient realizations of several more advanced cryptographic tools, including signature schemes, universally composable oblivious transfer, and identity-based encryption.

The core technique underlying all these results is an efficient algorithm that "obliviously" samples from *Gaussian-like* probability distributions over arbitrary lattices, given a basis (or even just a full-rank set) of appropriate length. This technique also yields simpler and (slightly) tighter worst-case/average-case connections for lattice problems, and may have additional applications in complexity theory and cryptography.

[1] Indirect (but complex and inefficient) constructions are of course possible by a generic transformation from universal one-way hash functions [NY89], or (in the random oracle model) by applying the Fiat-Shamir heuristic [FS86] to lattice-based identification schemes [MV03].

1.1.1 Gaussian Sampling Algorithm

Because it is the main foundation of our cryptographic results, we start with a summary of our sampling algorithm. Given a basis $\mathbf{B} \subset \mathbb{R}^n$ of some arbitrary n-dimensional lattice[2] Λ (or even just a full-rank set $\mathbf{S} \subset \Lambda$), the algorithm chooses a lattice vector at random under a Gaussian-like probability distribution whose standard deviation is essentially the length of the longest vector in $\mathbf{B}$ (or $\mathbf{S}$). Such *discrete Gaussian* distributions over lattices have been used in mathematics to prove tight "transference theorems" for lattices [Ban93, Ban95], and more recently have proved exceedingly useful in studying the computational complexity of lattice problems [AR03, AR05, Pei07], particularly their worst-case/average-case connections (e.g., [Reg04, MR07, Reg05]). Up to this point, however, discrete Gaussians have been used primarily as an *analysis* tool, rather than an *algorithmic* one.

The sampling algorithm can also be viewed as a randomized *decoder* that, given some short lattice vectors as "advice," outputs a lattice vector that is relatively close to any given target point $\mathbf{t} \in \mathbb{R}^n$. The key feature of the decoder is that its output distribution depends only on the *length* of the advice vectors, and is otherwise oblivious to their particular geometry. This will be a crucial security property in our cryptographic applications.

The algorithm itself is actually a simple *randomized* variant of Babai's "nearest-plane" algorithm [Bab86], with one small difference. Instead of always choosing the *nearest* plane, the algorithm chooses a particular plane with a probability determined by its distance from the target point. It turns out that this simple change causes the output distribution to be statistically close to a discrete Gaussian, though the analysis is non-trivial and crucially uses the notion of the *smoothing parameter* introduced by Micciancio and Regev [MR07].[3] The full details are given in Section 3.

1.1.2 Cryptographic Constructions

For all of our cryptographic constructions, we will need to generate a *"hard" public* basis for a random lattice Λ, together with a *short private* basis for Λ, which will be used as advice for the sampling algorithm. Our preferred approach is due to Ajtai [Ajt99], who described a way to generate such bases so that the public basis has *worst-case* hardness. (As far as we know, our results are the first use of Ajtai's generator in cryptography or otherwise.)

Trapdoor functions with *preimage sampling*. The basic object underlying our higher-level cryptographic tools is a collection of one-way (and also collision-resistant) trapdoor functions. Very informally, evaluating a public function f corresponds to choosing a lattice point $\mathbf{v} \in \Lambda$ "uniformly at random" and perturbing it by some short error term $\mathbf{e}$, yielding a point $\mathbf{y} = \mathbf{v} + \mathbf{e}$.[4] Finding a preimage of $\mathbf{y}$ corresponds to *decoding* it to a sufficiently nearby lattice point $\mathbf{v}' \in \Lambda$ (not necessarily the original $\mathbf{v}$). This is easy using our sampling algorithm with the trapdoor, but should otherwise be hard (for the particular distribution of Λ and $\mathbf{y}$).

Our trapdoor functions have two crucial properties for security in cryptographic applications. First, the input (the error term $\mathbf{e}$) is drawn from a *Gaussian* distribution with relatively small

[2] The lattice generated by the basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is the set of all integer combinations of the linearly-independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$.

[3] In private communication, Lyubashevsky has pointed out that Klein [Kle00] also proposed a randomized nearest-plane algorithm in another context, but gave only a limited analysis of its output distribution. See Section 1.3.

[4] Being an infinite set, the lattice Λ cannot support a uniform distribution. In actuality, the function simply reduces the error term $\mathbf{e}$ modulo the public basis for Λ.

standard deviation, and under this distribution, the output $\mathbf{y}$ is statistically close to *uniform* over the range. This equates the inversion problem with decoding a uniformly random point given the public basis, for which we can show worst-case hardness. Second, the trapdoor inversion algorithm does not just find an *arbitrary* preimage of $\mathbf{y}$, but actually *samples* from all its preimages (under the appropriate conditional distribution), using our discrete Gaussian sampling algorithm. In other words, the inverter samples an input $\mathbf{e}$ from the Gaussian input distribution, conditioned on the event that $f(\mathbf{e}) = \mathbf{y}$. Taken together, these two properties imply that there are two (nearly) equivalent ways of choosing a pair $(\mathbf{e}, \mathbf{y} = f(\mathbf{e}))$: either choose $\mathbf{e}$ from the input distribution and compute $\mathbf{y} = f(\mathbf{e})$, or choose $\mathbf{y}$ uniformly at random and sample $\mathbf{e}$ from $f^{-1}(\mathbf{y})$. As we shall see, this makes our trapdoor functions "as good as" trapdoor *permutations* in certain applications.[5]

Further discussion of random lattices, the hard decoding problem, and the full details of our trapdoor functions appear in Section 4.

Signature schemes. There are several (relatively efficient) digital signature schemes in the literature based on trapdoor permutations: full-domain hash-like schemes [DH76, BR93, BR96, Cor02] that follow the so-called "hash-and-sign paradigm" and have security analyses in the random oracle model, and the scheme of Bellare and Micali [BM92] that is secure in the standard model.

We show that all of the above permutation-based signature schemes can be instantiated equally well using trapdoor functions with preimage sampling, and retain their security analyses in their respective models (though subtleties can arise when dealing with multiple queries on the same message). In fact, we can demonstrate *tight* security reductions using the collision-resistance property of our functions, rather than just their one-wayness.

Concretely, our hash-and-sign signatures bear a close resemblance to the original (but insecure) GGH proposal and its variants [GGH97, HHGP+03]: informally, a message is hashed to a random point in space, and its signature is essentially a nearby lattice point, which is found using a short basis. Our schemes have two main differences: first, they are based on random lattices that enjoy worst-case hardness; second and more importantly, the signatures are generated by a *randomized* decoding algorithm, whose output distribution is *oblivious* to the particular geometry of the trapdoor basis. (The original GGH proposal was insecure precisely because its signatures leaked information about the "shape" of the private basis, allowing full key recovery [NR06].)

Signature schemes are constructed and analyzed in Section 5.

Efficient universally composable oblivious transfer. Our next applications are centered around a cryptosystem of Regev that is based on a problem called *learning with error* (LWE) [Reg05]. We demonstrate a trapdoor that applies to an optimized version of the system in which all parties generate their public keys relative to some shared random lattice.[6]

The first application of this trapdoor is *universally composable oblivious transfer*, the topic of a concurrent work by Peikert *et al.* [PVW07]. The central objects of study in that work are called *message-lossy* (or "messy") public keys, whose defining property is that a ciphertext produced under such a key carries *no information* (statistically) about the encrypted message.

[5] It is also possible to use other (non-Gaussian) input distributions, as long as the function's output remains close to uniform and there is an efficient way to sample preimages under the (conditional) input distribution.

[6] The existence of a trapdoor means that users of the system must implicitly trust the creator of the shared lattice, because it can extract secret keys from any properly-generated public keys. We note that this by no means constitutes a "break" of Regev's system, because parties can generate their own individual lattices.

Prior works on lattice-based encryption [AD97, Reg04, Reg05] used probabilistic arguments to show that messy keys are very dense. In this work, we give an *explicit* geometric description of messy keys in (a slight variant of) Regev's cryptosystem. Essentially, a public key is messy if, after adjoining it to the shared lattice, the minimum distance of the lattice remains large. We then show how to recognize such keys, by using our Gaussian sampling algorithm to *efficiently* implement the preprocessing step of an algorithm that identifies points that are far from a given lattice, due to Aharonov and Regev [AR05].

The trapdoor for Regev's cryptosystem is described in further detail in Section 6.

Identity-based encryption. Identity-based encryption (IBE), first proposed by Shamir [Sha84], permits arbitrary strings to serve as public keys in a cryptosystem. Thus far, IBE has been realized using groups with bilinear pairings (e.g., [BF03, BB04, Wat05]) and under the quadratic residuosity (QR) assumption in the random oracle model, or under an "interactive" QR assumption in the standard model [Coc01, BGH07].

Our final application is an efficient IBE based on LWE in the random oracle model, or under an interactive assumption on the hardness of LWE in the standard model. Although it is possible to extract secret keys from public keys (using a trapdoor) in Regev's LWE-based cryptosystem, obtaining IBE is still not entirely straightforward. Essentially, the problem is that well-formed public keys are exponentially *sparse*, because they consist only of points that are very close to the shared lattice. Hence, it is difficult to see how a hash function or a random oracle could map identities to valid public keys. We circumvent this problem by constructing a "dual" of Regev's cryptosystem, in which the key generation and encryption algorithms are effectively swapped. In the resulting system, *every* point in space is a valid public key that has many equivalent secret keys, which are simply the lattice points sufficiently close to the public key.

Due to its use of a trapdoor for extracting secret keys, our IBE is structurally most similar to those based on QR [Coc01, BGH07]. It is remarkably efficient, at least in an asymptotic sense: for $\tilde{\Theta}(n)$-bit messages (where n is the security parameter), the amortized running time of encryption and decryption is only $\tilde{O}(n)$ bit operations per encrypted bit. The ciphertext expansion is $O(1)$, and can even brought arbitrarily close to 1 under a strong enough (but still polynomial) assumption on LWE. One possible drawback of our system is that the master public key and individual secret keys are $\tilde{O}(n^2)$ bits.

The constructions of the dual cryptosystem and IBE appear in Section 7.

1.2 Assumptions

Our trapdoor functions and signatures are built upon an average-case "absolute distance" decoding problem on random lattices. Our "encryption-like" primitives, i.e., oblivious transfer and identity-based encryption, are built on the average-case learning with error (LWE) problem, defined by Regev [Reg05]. LWE can seen as a "unique decoding" problem on a certain family of random lattices, and is a generalization (to larger moduli) of the well-studied *learning parity with noise* problem, which is believed to be hard and has seen many applications in cryptography (e.g., [BKW03, HB01, JW05, KS06]).

Both of the above decoding problems are hard on the average if standard lattice problems (like the shortest vector problem) are hard to approximate to within small polynomial factors in the dimension of the lattice, in the *worst case*. However, the reduction for LWE is *quantum* [Reg05], i.e., LWE is hard if certain lattice problems are hard in the worst case for quantum algorithms. Given the state of the art, a stronger assumption for encryption-like primitives is necessary, because there is no known cryptosystem based on *classical* worst-case hardness for general lattices.[7]

For the lattice problems underlying our schemes, the best known classical (and quantum) algorithms require time and space exponential in the dimension of the lattice [AKS01], and the best known polynomial-time algorithms deliver exponentially-loose approximations [LLL82, Sch87].

1.3 Related Work

Using completely different techniques, Peikert and Waters [PW07] have constructed an entirely complementary collection of *injective* trapdoor functions based on lattice problems (among others). Their TDFs are based on the hardness of LWE, whereas ours are based entirely on *classical* worst-case hardness. Their TDFs are injective with exponentially-sparse images, whereas ours are many-to-one and surjective. Their TDFs imply *chosen ciphertext-secure* encryption, but the sparse images seem to make them less well-suited toward applications like signature schemes and IBE. Finally, from a purely aesthetic point of view, our trapdoor functions correspond more directly to a "natural" decoding problem on lattices.

As mentioned above, Peikert, Vaikuntanathan, and Waters [PVW07] have constructed efficient and universally composable oblivious transfer protocols from worst-case lattice assumptions (among others), using our trapdoor for detecting messy keys. That work predates this one, and provided motivation for our trapdoor techniques and Gaussian sampling algorithm.

In private communication, Lyubashevsky has pointed out that the randomized nearest-plane algorithm we use for Gaussian sampling is essentially the same algorithm proposed by Klein [Kle00] for solving a different problem. Klein considers a restricted variant of the closest vector problem, in which the target point is guaranteed to be "unusually close" to the lattice. His analysis shows that under such a promise, the algorithm outputs the closest lattice vector with noticeable probability. Our contribution is an analysis of the *entire* output distribution of the algorithm for arbitrary target points, which shows that the distribution is oblivious to the advice basis.

2 Preliminaries

2.1 Notation

We denote set of real numbers by $\mathbb{R}$ and the integers by $\mathbb{Z}$. For a positive integer n, $[n]$ denotes $\{1, \ldots, n\}$. We extend any real function $f(\cdot)$ to a countable set A by defining $f(A) = \sum_{x \in A} f(x)$.

By convention, vectors are assumed to be in column form and are written using bold lower-case letters, e.g. $\mathbf{x}$. The ith component of $\mathbf{x}$ will be denoted by x_i. Matrices are written as bold capital letters, e.g. $\mathbf{X}$, and the ith column vector of a matrix $\mathbf{X}$ is denoted $\mathbf{x}_i$. The length of a matrix is the norm of its longest column: $\|\mathbf{X}\| = \max_i \|\mathbf{x}_i\|$. For notational convenience, we sometimes view a matrix as simply the set of its column vectors.

The natural security parameter throughout the paper is n, and all other quantities are implicitly functions of n. We use standard big-O notation to classify the growth of functions, and say that $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n) \cdot \log^c n)$ for some fixed constant c. We let $\mathrm{poly}(n)$ denote an unspecified function $f(n) = O(n^c)$ for some constant c. A *negligible* function, denoted generically

[7]The cryptosystems from [AD97, Reg04] are based on classical worst-case hardness, but for a restricted class of lattices having "unique" shortest nonzero vectors.

by $\mathsf{negl}(n)$, is an $f(n)$ such that $f(n) = o(n^{-c})$ for every fixed constant c. We say that a probability (or fraction) is *overwhelming* if it is $1 - \mathsf{negl}(n)$.

The *statistical distance* between two distributions X and Y over a domain D is defined to be $\frac{1}{2}\sum_{d\in D}|X(d) - Y(d)|$. We say that two distributions (formally, two ensembles of distributions indexed by n) are *statistically close* if their statistical distance is negligible in n.

Two ensembles of distributions $\{X_n\}$ and $\{Y_n\}$ are *computationally indistinguishable* if for every probabilistic poly-time machine $\mathcal{A}$, $|\Pr[\mathcal{A}(1^n, X_n) = 1] - \Pr[\mathcal{A}(1^n, Y_n) = 1]|$ is negligible (in n). The definition can be extended to non-uniform families of poly-sized circuits in the standard way.

2.2 Standard Definitions

TODO: strong unforgeability under chosen-message attack

TODO: (ANON-)IND-CPA for IBE.

2.3 Lattices

An n-dimensional *lattice* of rank $k \leq n$ is

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{\mathbf{Bc} \ : \ \mathbf{c} \in \mathbb{Z}^k\right\}, \ \mathbf{B} \in \mathbb{R}^{n\times k}$$

where the k columns $\mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{R}^n$ of the *basis* $\mathbf{B}$ are linearly independent. The *dual lattice* of Λ, denoted Λ^*, is defined as $\Lambda^* = \{\mathbf{x} \in \mathrm{span}(\Lambda) \ : \ \forall\, \mathbf{v} \in \Lambda, \langle \mathbf{x}, \mathbf{v}\rangle \in \mathbb{Z}\}$.

The *minimum distance* $\lambda_1(\Lambda)$ of a lattice Λ is the length (in some norm, implicitly the Euclidean ℓ_2 norm) of its shortest nonzero element: $\lambda_1(\Lambda) = \min_{0\neq\mathbf{x}\in\Lambda}\|\mathbf{x}\|$. More generally, the *ith successive minimum* $\lambda_i(\Lambda)$ is the smallest radius r such that Λ contains i linearly independent vectors of norm at most r. We will write λ_1^∞ to denote the minimum distance measured in the ℓ_∞ norm (which is defined as $\|\mathbf{x}\|_\infty = \max|x_i|$).

A lattice is a discrete additive subgroup of $\mathbb{R}^n$. Therefore for lattices $\Lambda' \subseteq \Lambda$, the quotient group Λ/Λ' (also written $\Lambda \bmod \Lambda'$) is well-defined as the group of distinct *cosets* $\mathbf{v} + \Lambda'$ for $\mathbf{v} \in \Lambda$, with addition of cosets defined in the usual way. In particular, if both Λ' and Λ are of the same rank k, then Λ/Λ' is finite.

For completeness, we recall two standard worst-case approximation problems on lattices (though we will not need to refer to their specific definitions at any point). In both problems, $\gamma = \gamma(n)$ is the approximation factor as a function of the dimension.

Definition 2.1 (Shortest Vector Problem (Decision Version))**.** An input to GapSVP_γ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ a basis for a full-rank n-dimensional lattice and $d \in \mathbb{R}$. It is a YES instance if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$, and is a NO instance if $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n)\cdot d$.

Definition 2.2 (Shortest Independent Vectors Problem)**.** An input to SIVP_γ is a full-rank basis $\mathbf{B}$ of an n-dimensional lattice. The goal is to output a set of n linearly independent lattice vectors $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{S}\| \leq \gamma(n)\cdot\lambda_n(\mathcal{L}(\mathbf{B}))$.

2.4 Gaussians on Lattices

Our review of Gaussian measures over lattices follows the development by prior works [Reg04, AR05, MR07]. For any $s > 0$ define the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with parameter s:

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

The subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$ (respectively) when omitted.

For any $\mathbf{c} \in \mathbb{R}^n$, real $s > 0$, and n-dimensional lattice Λ of rank $k \leq n$, define the *discrete Gaussian distribution over* Λ as:

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

(As above, we may omit the parameters s or $\mathbf{c}$.) Note that the denominator in the above expression is simply a normalization factor; the probability $D_{\Lambda,s,\mathbf{c}}(\mathbf{x})$ is simply proportional to $\rho_{s,\mathbf{c}}(\mathbf{x})$. Therefore for any $\mathbf{c} \in \mathbb{R}^n$, the distributions $D_{\Lambda,s,\mathbf{c}}$ and $D_{\Lambda,s,\mathbf{c}'}$ are identical, where $\mathbf{c}'$ is the orthogonal projection of $\mathbf{c}$ onto $\mathrm{span}(\Lambda)$.

Micciancio and Regev [MR07] proposed a lattice quantity called the *smoothing parameter*:

Definition 2.3 ([MR07])**.** For any n-dimensional lattice Λ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest real $s > 0$ such that $\rho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$.

We recall two bounds on the smoothing parameter. The first relates the smoothing parameter of a rank-k lattice to its kth successive minimum λ_k (in ℓ_2 norm).

Lemma 2.4 ([MR07])**.** *For any n-dimensional lattice Λ of rank k and real $\epsilon > 0$, we have*

$$\eta_\epsilon(\Lambda) \leq \lambda_k(\Lambda) \cdot \sqrt{\log(2k(1+1/\epsilon))/\pi}.$$

Then for any $\omega(\sqrt{\log n})$ function, there is a negligible $\epsilon(n)$ for which $\eta_\epsilon(\Lambda) \leq \lambda_k(\Lambda) \cdot \omega(\sqrt{\log n})$.

The second bound relates the smoothing parameter of a lattice to the minimum distance of its dual lattice, *in ℓ_∞ norm*. We note that the smoothing parameter can also be related to the dual minimum distance in ℓ_2 norm (as shown in [MR07]), but the ℓ_∞ norm turns out to be easier to analyze for random lattices, and will also yield the tightest bounds on their smoothing parameters.

Lemma 2.5 ([Pei07])**.** *For any n-dimensional lattice Λ of rank k and real $\epsilon > 0$, we have*

$$\eta_\epsilon(\Lambda) \leq \frac{\sqrt{\log(2k/(1+1/\epsilon))/\pi}}{\lambda_1^\infty(\Lambda^*)}.$$

Then for any $\omega(\sqrt{\log n})$ function, there is a negligible $\epsilon(n)$ for which $\eta_\epsilon(\Lambda) \leq \omega(\sqrt{\log n})/\lambda_1^\infty(\Lambda^*)$.

We now state some central facts regarding discrete Gaussians that apply when the Gaussian parameter s exceeds the smoothing parameter of the lattice. The following lemma states that the total Gaussian measure on any *translate* of the lattice is essentially the same.

Lemma 2.6 ([MR07], implicit)**.** *Let Λ be any n-dimensional lattice of rank k. Then for any $s \geq \eta_\epsilon(\Lambda)$, real $\epsilon \in (0,1)$, and $\mathbf{c} \in \mathrm{span}(\Lambda)$, we have*

$$\rho_{s,\mathbf{c}}(\Lambda) \in [\tfrac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_s(\Lambda).$$

A corollary is that a Gaussian sample over Λ is distributed almost-uniformly modulo a sublattice Λ' (of the same rank), for $s \geq \eta_\epsilon(\Lambda')$.

Corollary 2.7. *Let Λ, Λ' be n-dimensional lattices of rank k, with $\Lambda' \subseteq \Lambda$. Then for any $\epsilon \in (0, \frac{1}{2})$, any $s \geq \eta_\epsilon(\Lambda')$, and any $\mathbf{c} \in \mathbb{R}^n$, the distribution of $(D_{\Lambda,s,\mathbf{c}} \bmod \Lambda')$ is within statistical distance at most 2ϵ of uniform over $(\Lambda \bmod \Lambda')$.*

Proof. We can assume $\mathbf{c} \in \text{span}(\Lambda)$ without loss of generality. Consider the marginal distribution of ($\mathbf{z} \bmod \Lambda'$) where $\mathbf{z} \leftarrow D_{\Lambda,s,\mathbf{c}}$. Then for any coset $\mathbf{v}+\Lambda'$ of Λ/Λ', the probability that $\mathbf{z} = \mathbf{v} \bmod \Lambda'$ is proportional to

$$\rho_{s,\mathbf{c}}(\mathbf{v} + \Lambda') = \rho_{s,\mathbf{c}-\mathbf{v}}(\Lambda') \in [\tfrac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_s(\Lambda')$$

by Lemma 2.6. By a routine calculation, it follows that for every $\mathbf{v}$, the probability that $\mathbf{z} = \mathbf{v} \bmod \Lambda'$ is within $(1 \pm 4\epsilon)/ |\Lambda/\Lambda'|$, which yields the claim. □

Another fact we need says that a sample from a discrete Gaussian with parameter s is at most $s\sqrt{n}$ away from its center (in ℓ_2 norm), with overwhelming probability.

Lemma 2.8 ([MR07]). *For any n-dimensional lattice Λ of rank k, $\mathbf{c} \in \text{span}(\Lambda)$, real $\epsilon \in (0, 1)$, and $s \geq \eta_\epsilon(\Lambda)$, we have*

$$\Pr_{\mathbf{x}\leftarrow D_{\Lambda,s,\mathbf{c}}} \left[\|\mathbf{x} - \mathbf{c}\| > s\sqrt{n}\right] \leq \tfrac{1+\epsilon}{1-\epsilon} \cdot 2^{-k}.$$

The final fact we need for certain applications is an upper bound on the probability of the *mode* (the most likely element) of a discrete Gaussian; equivalently, it is a lower bound on the *min-entropy* of the distribution. A bound that applies when $s \geq 2\eta_\epsilon(\Lambda)$ was first proved in [PR06]; here we give a different style of proof that works for $s \geq \eta_\epsilon(\Lambda)$ (the difference is mainly aesthetic).

Lemma 2.9. *For any n-dimensional lattice Λ of rank k, center $\mathbf{c} \in \mathbb{R}^n$, positive $\epsilon < \exp(-4\pi)$, and $s \geq \eta_\epsilon(\Lambda)$, and for every $\mathbf{x} \in \Lambda$, we have*

$$D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) \leq \tfrac{1+\epsilon}{1-\epsilon} \cdot 2^{-k}.$$

In particular, the min-entropy of $D_{\Lambda,s,\mathbf{c}}$ is at least $k - 1$.

Proof. We can assume without loss of generality that $\mathbf{c} \in \text{span}(\Lambda)$. For any $\mathbf{x} \in \Lambda$, we have

$$D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)} \leq \frac{1}{\rho_{s,\mathbf{c}}(\Lambda)} \leq \frac{1+\epsilon}{1-\epsilon} \cdot \frac{1}{\rho_s(\Lambda)},$$

where the final inequality is by Lemma 2.6. Therefore it suffices to show that $\rho_s(\Lambda) \geq 2^k$.

By the Poisson summation formula,

$$\rho_s(\Lambda) = \det(\Lambda^*) \cdot s^k \cdot \rho_{1/s}(\Lambda^*) \geq \det(\Lambda^*) \cdot s^k. \tag{2.1}$$

Now by Minkowski's second theorem, we have

$$\det(\Lambda^*) \geq \prod_{i\in[k]} \lambda_i(\Lambda^*) \geq (\lambda_1^*)^k, \tag{2.2}$$

where we write $\lambda_1^* = \lambda_1(\Lambda^*)$. Let $\mathbf{v} \in \Lambda^*$ be such that $\|\mathbf{v}\| = \lambda_1^*$. Now because $s \geq \eta_\epsilon(\Lambda)$, we have

$$\epsilon \geq \rho_{1/s}(\Lambda^* \backslash \{\mathbf{0}\}) \geq \rho_{1/s}(\mathbf{v}) = \exp(-\pi \cdot (s\lambda_1^*)^2).$$

Therefore $s \geq \sqrt{\ln(1/\epsilon)/\pi}/\lambda_1^* \geq 2/\lambda_1^*$. Combining this with Inequalities (2.1) and (2.2), we conclude that $\rho_s(\Lambda) \geq 2^k$ as desired. □

2.5 Learning with Error

We start by introducing the *learning with error* (LWE) problems, for the most part following [Reg05].

For $x \in \mathbb{R}$, $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$ denotes a nearest integer to x. Denote $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ as the group of reals $[0, 1)$ with mod 1 addition.

Probability distributions. The *normal (Gaussian) distribution* with mean 0 and variance σ^2 (or standard deviation σ) is the distribution on $\mathbb{R}$ having density function $\frac{1}{\sigma \cdot \sqrt{2\pi}} \exp(-x^2/2\sigma^2)$. The sum of two independent normal variables with mean 0 and variances σ_1^2 and σ_2^2 (respectively) is a normal variable with mean 0 and variance $\sigma_1^2 + \sigma_2^2$. We will also need a standard tail inequality: a normal variable with variance σ^2 is within distance $t \cdot \sigma$ (i.e., t standard deviations) of its mean, except with probability at most $\frac{1}{t} \cdot \exp(-t^2/2)$.

For $\alpha \in \mathbb{R}^+$, Ψ_α is defined to be the distribution on $\mathbb{T}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1. For any probability distribution ϕ over $\mathbb{T}$ and an integer $q \in \mathbb{Z}^+$ (often implicit) its *discretization* $\bar{\phi}$ is the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q \cdot X_\phi \rceil \bmod q$, where X_ϕ has distribution ϕ.

For an integer $q \geq 2$ and some probability distribution χ over $\mathbb{Z}_q$, an integer dimension $n \in \mathbb{Z}^+$ and a vector $\mathbf{s} \in \mathbb{Z}_q^n$, define $A_{\mathbf{s},\chi}$ as the distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ of the variable $(\mathbf{a}, \mathbf{a}^T\mathbf{s} + x)$ where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ is uniform and $x \leftarrow \chi$ are independent, and all operations are performed in $\mathbb{Z}_q$.

Learning with error (LWE). For an integer $q = q(n)$ and a distribution χ on $\mathbb{Z}_q$, the goal of the (average-case) *learning with error* problem $\mathsf{LWE}_{q,\chi}$ is to distinguish (with nonnegligible probability) between the distribution $A_{\mathbf{s},\chi}$ for some uniform (secret) $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ (via oracle access to the given distribution). In other words, if LWE is hard, then the collection of distributions $A_{\mathbf{s},\chi}$ is pseudorandom.

Regev demonstrated that for certain moduli q and Gaussian error distributions χ, $\mathsf{LWE}_{q,\chi}$ is as hard as solving several standard *worst-case* lattice problems *using a quantum algorithm.*

Proposition 2.10 ([Reg05])**.** *Let $\alpha = \alpha(n) \in (0, 1)$ and let $q = q(n)$ be a* prime *such that $\alpha \cdot q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that solves $\mathsf{LWE}_{q,\bar{\Psi}_\alpha}$, then there exists an efficient* quantum *algorithm for approximating SIVP and GapSVP in the ℓ_2 norm, in the worst case, to within $\tilde{O}(n/\alpha)$ factors.*

Peikert [Pei07] extended this result to hold for SIVP and GapSVP in *any* ℓ_p norm, $2 \leq p \leq \infty$, for essentially the same $\tilde{O}(n/\alpha)$ approximation factors.

3 Sampling from Discrete Gaussians

Here we show how to efficiently sample from a discrete Gaussian probability distribution $D_{\Lambda,s,\mathbf{c}}$, given any basis (or full-rank set) of length bounded by about s. As a first attempt, we might consider an algorithm that first samples from a *continuous* Gaussian, and then uses the short basis to "round off" the sampled point to a nearby lattice point. In fact, Regev used this exact strategy for the "bootstrapping" step of his reduction [Reg05], using an LLL-reduced basis and a Gaussian parameter s that was *exponentially* larger than the basis length. Unfortunately, it is not clear that this strategy works as well when s is a smaller (e.g., $\text{poly}(n)$) multiple of the basis length. The problem is that within the region of space that is mapped (via rounding) to a particular lattice point, the density function of the continuous Gaussian may vary significantly. This makes it difficult to analyze the total probability mass assigned to each lattice point under the rounding scheme, and to compare it to the desired discrete Gaussian distribution.[8]

The sampling algorithm below avoids continuous Gaussians altogether, and samples "directly" from the lattice in a tight way. It can be seen as a randomized version of Babai's nearest plane algorithm [Bab86], where instead of always choosing the nearest plane, the algorithm chooses a plane at random with a probability related to its distance from the target point. The rest of this section is devoted to formally defining and analyzing the sampling algorithm.

3.1 Sampling Integers

We first develop a core subroutine, which samples from a discrete Gaussian distribution over a particular one-dimensional lattice, namely, the integers $\mathbb{Z}$. This subroutine will depend on a tail inequality for the distribution:

Lemma 3.1. *For any $\epsilon > 0$, any $s \geq \eta_\epsilon(\mathbb{Z})$, and any $t > 0$, we have*

$$\Pr_{x \leftarrow D_{\mathbb{Z},s,c}}[|x - c| \geq t \cdot s] \leq 2e^{-\pi t^2} \cdot \tfrac{1+\epsilon}{1-\epsilon}.$$

Then for $\epsilon \in (0, \frac{1}{2})$ and $t \geq \omega(\sqrt{\log n})$, the probability that $|x - c| \geq t \cdot s$ is negligible (in n).

Proof. Let $\mathcal{B} = (-1, 1)$. We will use the fact (proved more generally in [Ban95]) that

$$\rho_s((\mathbb{Z} - c) \backslash t \cdot s \cdot \mathcal{B}) \quad \leq \quad 2e^{-\pi t^2} \cdot \rho_s(\mathbb{Z}).$$

Now consider the total probability assigned by $D_{\mathbb{Z},s,c}$ to all integers outside $t \cdot s \cdot (\mathcal{B} + c)$. This is

$$D_{\mathbb{Z},s,c}(\mathbb{Z} \backslash (s \cdot t \cdot (\mathcal{B} + c))) = \frac{\rho_s((\mathbb{Z} - c) \backslash t \cdot s \cdot \mathcal{B})}{\rho_{s,c}(\mathbb{Z})} \leq \frac{2e^{-\pi t^2} \cdot \rho_s(\mathbb{Z})}{\rho_{s,c}(\mathbb{Z})} \leq \frac{2e^{-\pi t^2} \cdot \rho_s(\mathbb{Z})}{\frac{1-\epsilon}{1+\epsilon} \cdot \rho_s(\mathbb{Z})},$$

where we have used Lemma 2.6 for the last inequality. This completes the proof. □

There are several possible methods for efficiently sampling from $D_{\mathbb{Z},s,c}$ (for large enough s); the easiest to describe works by the standard method of rejection sampling. Let $t(n) \geq \omega(\sqrt{\log n})$ be some fixed function. The algorithm $\mathsf{SampleD}$ works as follows: on input (s, c) and (implicitly) the security parameter n, choose an integer $x \leftarrow Z = \mathbb{Z} \cap [c - s \cdot t, c + s \cdot t]$ uniformly at random. Then with probability $\rho_s(x - c)$, output x, otherwise repeat.

Lemma 3.2. *For any $\epsilon \in (0, 1/2)$ and any $s \geq \eta_\epsilon(\mathbb{Z})$, $\mathsf{SampleD}$ runs in time polynomial in n and the size of its input (s, c), and its output distribution is statistically close to $D_{\mathbb{Z},s,c}$.*

Proof. First define a probability distribution D on $\mathbb{Z}$ in which $D(x)$ is proportional to $\rho_s(x - c)$ for every $x \in Z$, and $D(x) = 0$ otherwise. By rejection sampling, the output distribution of $\mathsf{SampleD}$ is identical to D. Furthermore, the D and $D_{\mathbb{Z},s,c}$ are statistically close, by Lemma 3.1.

[8] By using an additional step of rejection sampling, one can effectively "flatten" the continuous distribution over each region that is mapped to a particular lattice point. However, the resulting algorithm is still somewhat "loose," working for a parameter s that is a $\sqrt{n}$ factor of the *diameter* the basis (which may be up to an n factor larger than the basis length).

We now analyze the running time. Each iteration of SampleD picks a uniformly random integer x from Z. The probability that the chosen integer x falls in the set $\mathbb{Z} \cap [c-s, c+s]$ is at least $(2s-1)/(2s \cdot t) \geq 1/2t$, because $s \geq \eta_\epsilon(\mathbb{Z}) \geq 1$. Once chosen, x is then output with probability $\rho_s(x-c) \geq \exp(-\pi)$, a positive constant. By a standard repetition argument, the running time is $O(t(n))$ with all but an exponentially small probability (which can be made zero by terminating after a certain number of iterations, without significantly altering the output distribution). $\square$

3.2 Sampling from Arbitrary Lattices

Overloading notation, we now describe an algorithm SampleD that samples from a discrete Gaussian over any lattice. The algorithm works exactly like Babai's "nearest plane" algorithm, except that it chooses a plane at random with a probability given by a discrete Gaussian over the integers $\mathbb{Z}$.

The SampleD algorithm takes some n-dimensional basis $\mathbf{B} \in \mathbb{Z}^{n \times k}$ of rank k, a sufficiently large Gaussian parameter s (related to the length $\|\mathbf{B}\|$ of the basis), a center $\mathbf{c} \in \mathbb{R}^n$, and efficiently outputs a sample from (a distribution close to) $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$. It proceeds as follows: on input $(\mathbf{B}, s, \mathbf{c})$,

1. If $k = 0$ (i.e., if $\mathbf{B}$ is empty), return $\mathbf{0}$.
2. Compute $\tilde{\mathbf{b}}_k$, the (nonzero) component of $\mathbf{b}_k$ orthogonal to $\text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{k-1})$.
3. Compute $\mathbf{t}$, the projection of $\mathbf{c}$ onto $\text{span}(\mathbf{B})$, and the scalar value $t = \langle \mathbf{t}, \tilde{\mathbf{b}}_k \rangle / \langle \tilde{\mathbf{b}}_k, \tilde{\mathbf{b}}_k \rangle \in \mathbb{R}$.
4. Choose an integer $z \leftarrow D_{\mathbb{Z}, s/\|\tilde{\mathbf{b}}_k\|, t}$, using the SampleD algorithm for $\mathbb{Z}$ described above. (This is the only step that differs from the standard nearest plane algorithm.)
5. Output $z\mathbf{b}_k + \text{SampleD}(\mathbf{B}', s, \mathbf{t} - z\mathbf{b}_k)$ via recursion, where $\mathbf{B}' = [\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}]$.

As described, the sampler runs in time $O(n^2)$ times the representation length (in bits) of $\mathbf{B}, \mathbf{t}$. As an optimization, the Gram-Schmidt vectors $\tilde{\mathbf{b}}_k$ (and their lengths) and the scalar values $\langle \mathbf{t}, \tilde{\mathbf{b}}_k \rangle$ can be pre-computed if the sampler is called multiple times on the same basis $\mathbf{B}$ and center $\mathbf{t}$. With these optimizations, the sampler runs in linear $O(n)$ *amortized* time (times the bit length of $\mathbf{B}, \mathbf{t}$) to produce $\Omega(n)$ samples.

Theorem 3.3. *For any lattice basis $\mathbf{B} \in \mathbb{Z}^{n \times k}$, any real $s \geq \|\mathbf{B}\| \cdot \omega(\sqrt{\log n})$, and any $\mathbf{c} \in \mathbb{R}^n$, the output distribution of* SampleD$(\mathbf{B}, s, \mathbf{c})$ *is within negligible statistical distance of $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$. The running time of* SampleD *is polynomial in n and the size of its input $(\mathbf{B}, s, \mathbf{c})$.*

The main strategy for proving the theorem is to analyze the desired distribution $D = D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$, specifically, the total probability mass it assigns to the lattice points in each distinct plane. Each plane contains some *translate* of the same sublattice of rank $k-1$ (generated by $\mathbf{B}'$). The smoothing parameter will guarantee that the mass assigned by D to the translated sublattice is essentially *independent* of the translation, and therefore only depends on the distance between the plane and the center $\mathbf{c}$. Therefore it suffices for the algorithm to choose a plane with a probability related to this distance, and proceed recursively.

Before proving Theorem 3.3, we develop some technical facts about the distribution $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$ that make the above intuition more precise. Let $\mathbf{B}$, $\mathbf{B}'$, and $\tilde{\mathbf{b}}_k \neq \mathbf{0}$ be as in the algorithm above. Define $\mathbf{b}'_k = \mathbf{b}_k - \tilde{\mathbf{b}}_k \in \text{span}(\mathbf{B}')$, and partition $\mathcal{L}(\mathbf{B})$ into a collection of sets $\{H_z\}_{z \in \mathbb{Z}}$, where

$$H_z = z\mathbf{b}_k + \mathcal{L}(\mathbf{B}') = z\tilde{\mathbf{b}}_k + (\mathcal{L}(\mathbf{B}') + z\mathbf{b}'_k).$$

For every $\mathbf{w} \in \mathcal{L}(\mathbf{B})$, define $\zeta(\mathbf{w}) = z$ where $\mathbf{w} \in H_z$.

Lemma 3.4. *Adopt the notation from above, let $s > 0$, and let $\mathbf{t} \in \operatorname{span}(\mathbf{B})$ be arbitrary. Let $\mathbf{w} \leftarrow D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}}$. Then for every $z \in \mathbb{Z}$,*

1. *The probability that $\zeta(\mathbf{w}) = z$ is proportional to*

$$\rho_{s/\|\tilde{\mathbf{b}}_k\|,t}(z) \cdot \rho_{s,\mathbf{t}'-z\mathbf{b}'_k}(\mathcal{L}(\mathbf{B}')),$$

   where $t = \langle \mathbf{t}, \tilde{\mathbf{b}}_k \rangle / \langle \tilde{\mathbf{b}}_k, \tilde{\mathbf{b}}_k \rangle$ and $\mathbf{t}' = \mathbf{t} - t\tilde{\mathbf{b}}_k \in \operatorname{span}(\mathbf{B}')$.

2. *The conditional distribution of $\mathbf{w} - z\mathbf{b}_k$, given $\zeta(\mathbf{w}) = z$, is exactly $D_{\mathcal{L}(\mathbf{B}'),s,\mathbf{t}-z\mathbf{b}_k}$.*

Proof. We start with the first claim. The probability that $\zeta(\mathbf{w}) = z$ for any fixed $z \in \mathbb{Z}$ is simply $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}}(H_z)$, which is proportional to $\rho_{s,\mathbf{t}}(H_z)$. Now we have

$$\begin{aligned}
\rho_{s,\mathbf{t}}(H_z) &= \rho_s((z-t)\tilde{\mathbf{b}}_k + (\mathcal{L}(\mathbf{B}') + z\mathbf{b}'_k - \mathbf{t}')) \\
&= \rho_s((z-t)\tilde{\mathbf{b}}_k) \cdot \rho_s(\mathcal{L}(\mathbf{B}') + z\mathbf{b}'_k - \mathbf{t}') \qquad (3.1) \\
&= \rho_{s/\|\tilde{\mathbf{b}}_k\|,t}(z) \cdot \rho_{s,\mathbf{t}'-z\mathbf{b}'_k}(\mathcal{L}(\mathbf{B}')) \qquad (3.2)
\end{aligned}$$

where (3.1) follows because because $\tilde{\mathbf{b}}_k$ is orthogonal to $\operatorname{span}(\mathbf{B}')$, and (3.2) follows by definition of ρ. This establishes the first claim.

For the second claim, consider the conditional distribution D of $\mathbf{w}' = \mathbf{w} - z\mathbf{b}_k \in \mathcal{L}(\mathbf{B}')$, given $\mathbf{w} \in H_z$. This distribution is

$$D(\mathbf{w}') = \frac{\rho_{s,\mathbf{t}}(\mathbf{w})}{\rho_{s,\mathbf{t}}(z\mathbf{b}_k + \mathcal{L}(\mathbf{B}'))} = \frac{\rho_{s,\mathbf{t}-z\mathbf{b}_k}(\mathbf{w} - z\mathbf{b}_k)}{\rho_{s,\mathbf{t}-z\mathbf{b}_k}(\mathcal{L}(\mathbf{B}'))} = D_{\mathcal{L}(\mathbf{B}'),s,\mathbf{t}-z\mathbf{b}_k}(\mathbf{w}'). \qquad \square$$

We now prove the main theorem.

Proof of Theorem 3.3. First recall that $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$ and $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}}$ are identical, where $\mathbf{t}$ is the projection of $\mathbf{c}$ onto $\operatorname{span}(\mathbf{B})$. Therefore it enough to prove that the algorithm samples from (a distribution close to) $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}}$.

The running time of the algorithm may be easily verified. The proof of correctness proceeds by induction on the rank k of the lattice. When $k = 0$ (i.e, when $\mathbf{B}$ is empty), $\mathcal{L}(\mathbf{B}) = \{\mathbf{0}\}$ and $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$ assigns probability 1 to $\mathbf{0}$. Therefore the base case is correct.

For $k > 0$, the inductive hypothesis is that the distribution of $z\mathbf{b}_k + \mathsf{SampleD}(\mathbf{B}', s, \mathbf{t} - z\mathbf{b}_k)$ is statistically close to $z\mathbf{b}_k + D_{\mathcal{L}(\mathbf{B}'),s,\mathbf{t}-z\mathbf{b}_k}$, which by Lemma 3.4 is the conditional distibution of a sample $\mathbf{w} \leftarrow D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}}$ given $\zeta(\mathbf{w}) = z$. Therefore it suffices to show that the algorithm samples from the appropriate marginal distribution on $\zeta(\mathbf{w})$.

For every $z \in \mathbb{Z}$, by Lemma 3.4 the marginal probability that $\zeta(\mathbf{w}) = z$ is proportional to

$$\rho_{s/\|\tilde{\mathbf{b}}_k\|,t}(z) \cdot \rho_{s,\mathbf{t}'-z\mathbf{b}'_k}(\mathcal{L}(\mathbf{B}')),$$

where $t \in \mathbb{R}$ and $\mathbf{t}', \mathbf{b}'_k \in \operatorname{span}(\mathbf{B}')$ are as in the lemma. Now because $s \geq \|\mathbf{B}'\| \cdot \omega(\sqrt{\log n})$, there is a negligible $\epsilon(n)$ such that $s \geq \eta_\epsilon(\mathbf{B}')$ by Lemma 2.4. Thus, for every $z \in \mathbb{Z}$ we have

$$\rho_{s,\mathbf{t}'-z\mathbf{b}'_k}(\mathcal{L}(\mathbf{B}')) \in \left[\tfrac{1-\epsilon}{1+\epsilon}, 1\right] \cdot \rho_s(\mathcal{L}(\mathbf{B}'))$$

by Lemma 2.6. Therefore the distribution of $\zeta(\mathbf{w})$ is statistically close to $D_{\mathbb{Z},s/\|\tilde{\mathbf{b}}_k\|,t}$.

Now $s/\|\tilde{\mathbf{b}}_k\| \geq s/\|\mathbf{b}_k\| \geq \omega(\sqrt{\log n}) \geq \eta_\epsilon(\mathbb{Z})$ for some negligible $\epsilon(n)$ and by hypothesis on s. Therefore by Lemma 3.2, the distribution of z chosen by $\mathsf{SampleD}$ is statistically close to $D_{\mathbb{Z},s/\|\tilde{\mathbf{b}}_k\|,t}$, as desired. $\square$

3.3 Sampling with Independent Vectors

In certain applications, we will have a short set $\mathbf{S}$ of *linearly independent* lattice vectors that do not necessarily form a *basis* of the lattice. While one can efficiently convert such a set into a basis (using any basis $\mathbf{B}$ of arbitrary length), the resulting basis length may be up to a $\sqrt{n}$ factor larger than $\|\mathbf{S}\|$. Therefore, it will be more convenient to design an algorithm that can sample from the discrete Gaussian distribution, given only a basis $\mathbf{B}$ (of arbitrary length) and a short set $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ of the same rank as $\mathbf{B}$.

Overloading notation once again, we now describe an algorithm $\mathsf{SampleD}$ that, given some n-dimensional basis $\mathbf{B} \in \mathbb{Z}^{n \times k}$ of rank k, a set $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ of rank k, a sufficiently large Gaussian parameter s (related to $\|\mathbf{S}\|$), and a center $\mathbf{c} \in \mathbb{R}^n$, efficiently produces a sample from (a distribution very close to) $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$. It proceeds as follows: on input $(\mathbf{B}, \mathbf{S}, s, \mathbf{c})$,

1. Choose a lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\mathbf{v} \bmod \mathcal{L}(\mathbf{S})$ is distributed uniformly in $\mathcal{L}(\mathbf{B})/\mathcal{L}(\mathbf{S})$. (This can be done in time polynomial in the representation of $\mathbf{B}$ and $\mathbf{S}$; see [MG02].)

2. Choose $\mathbf{y} \leftarrow D_{\mathcal{L}(\mathbf{S}),s,\mathbf{c}-\mathbf{v}}$, by calling $\mathsf{SampleD}(\mathbf{S}, s, \mathbf{c} - \mathbf{v})$ with the algorithm described above.

3. Output $\mathbf{y} + \mathbf{v}$.

Lemma 3.5. *For $\mathbf{B}$, $\mathbf{S}$, and $\mathbf{c}$ as above, and any $s \geq \|\mathbf{S}\| \cdot \omega(\sqrt{\log n})$, the output distribution of $\mathsf{SampleD}(\mathbf{B}, \mathbf{S}, s, \mathbf{c})$ is within negligible statistical distance of $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$. The running time of $\mathsf{SampleD}$ is polynomial in n and the size of its input $(\mathbf{B}, \mathbf{S}, s, \mathbf{c})$.*

Proof. The running time of the algorithm is apparent. It suffices to show that the algorithm samples from (close to) $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$ when $\mathbf{y}$ is chosen exactly according to the distribution $D_{\mathcal{L}(\mathbf{S}),s,\mathbf{c}-\mathbf{v}}$; it then follows by Theorem 3.3 that the output distribution changes by only a negligible amount when $\mathbf{y}$ is chosen according to $\mathsf{SampleD}(\mathbf{S}, s, \mathbf{c} - \mathbf{v})$.

Suppose that $\mathbf{z}$ is chosen from $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$. Because $s \geq \|\mathbf{S}\| \cdot \omega(\sqrt{\log n})$, by Lemma 2.4 we have $s \geq \eta_\epsilon(\mathcal{L}(\mathbf{S}))$ for some negligible $\epsilon(n)$. Therefore by Lemma 2.7, the marginal distribution of $\mathbf{z} \bmod \mathcal{L}(\mathbf{S})$ is statistically close to uniform over $\mathcal{L}(\mathbf{B})/\mathcal{L}(\mathbf{S})$.

Now for any fixed $\mathbf{v} \in \mathcal{L}(\mathbf{B})$, consider the conditional distribution D of $\mathbf{z}$, given $\mathbf{z} = \mathbf{v} \bmod \mathcal{L}(\mathbf{S})$. For each $\mathbf{z} \in \mathbf{v} + \mathcal{L}(\mathbf{S})$, this conditional distribution is

$$D(\mathbf{z}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{z})}{\rho_{s,\mathbf{c}}(\mathbf{v} + \mathcal{L}(\mathbf{S}))} = \frac{\rho_{s,\mathbf{c}-\mathbf{v}}(\mathbf{z} - \mathbf{v})}{\rho_{s,\mathbf{c}-\mathbf{v}}(\mathcal{L}(\mathbf{S}))} = D_{\mathcal{L}(\mathbf{S}),s,\mathbf{c}-\mathbf{v}}(\mathbf{z} - \mathbf{v}).$$

Therefore $\mathbf{z}$ is distributed as $\mathbf{v} + D_{\mathcal{L}(\mathbf{S}),s,\mathbf{c}-\mathbf{v}}$.

In summary, the algorithm samples from (close to) both the appropriate marginal and conditional distributions of $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$, and the claim follows. $\square$

4 Trapdoors for Hard Random Lattices

In this section, we demonstrate trapdoors for certain random lattices that are, informally speaking, "as hard as" worst-case lattices. We then develop some foundational tools and primitives that our cryptographic applications will build upon.

4.1 Random Modular Lattices

We start by describing two kinds of *modular lattices* that have appeared in recent works. A modular lattice is an integer lattice that is invariant under shifts by q in each of the coordinates, for some fixed modulus q. In other words, whether a vector $\mathbf{x} \in \mathbb{Z}^m$ is an element of the lattice is determined entirely by $\mathbf{x} \bmod q$. For all of our applications, we will need q to be prime.

More formally, let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for some positive integers n, m, q. In prior works (as in this one), n is the natural security parameter and all other variables are functions of n; for example, $m = m(n)$ is typically $O(n \log n)$, and the modulus $q = q(n)$ is some small polynomial, e.g., $O(n^2)$. We consider two kinds of m-dimensional modular lattices defined by $\mathbf{A}$. The first lattice is generated by the (transposed) rows of $\mathbf{A}$, and is defined as

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

The second lattice consists of those integer vectors that are "orthogonal" (modulo q) to the rows of $\mathbf{A}$, and is defined as

$$\Lambda^{\perp}(\mathbf{A}, q) = \{\mathbf{e} \in \mathbb{Z}^m \ : \ \mathbf{A}\mathbf{e} = \mathbf{0} \bmod q\}.$$

In the parlance of coding theory, $\mathbf{A}^T$ is the "generator matrix" for the lattice $\Lambda(\mathbf{A}, q)$, and $\mathbf{A}$ is the "parity check" matrix for the lattice $\Lambda^{\perp}(\mathbf{A}, q)$. When $\mathbf{A}$ and q are implicit from context, we sometimes omit them as arguments and just write Λ and $\Lambda^{\perp}$.

Both lattices are modular and full-rank, because they contain the vectors $q \cdot \mathbf{e}_i$ for each $i \in [m]$, where $\{\mathbf{e}_i\}$ is the standard basis of $\mathbb{Z}^m$. It can be seen directly from their definitions that Λ and $\Lambda^{\perp}$ have a dual relationship; specifically, $\Lambda^{\perp} = q \cdot \Lambda^*$ and $\Lambda = q \cdot (\Lambda^{\perp})^*$. Another useful fact is that the quotient group $(\mathbb{Z}^m/\Lambda^{\perp})$ and the set of *syndromes* $\{\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q \ : \ \mathbf{e} \in \mathbb{Z}^m\} \subseteq \mathbb{Z}_q^n$ are in bijective correspondence, via the mapping $(\mathbf{e} + \Lambda^{\perp}) \mapsto \mathbf{A}\mathbf{e} \bmod q$.

Ajtai showed that for appropriate parameters, solving SVP on random instances of $\Lambda^{\perp}(\mathbf{A}, q)$ is as hard as approximately solving certain problems (e.g., SIVP and GapSVP) on *any* lattice of dimension n to within $\text{poly}(n)$ factors [Ajt96]. The first cryptographic application of this result was a candidate collection of one-way functions; further works constructed candidate collision-resistant hash functions, and improved the approximation factors in the underlying worst-case assumptions to as small as $\tilde{O}(n)$ [GGH96, CN97, Mic04, MR07].

Regev [Reg05] proposed the *learning with error* (LWE) problem, which can be phrased in terms of a *unique* decoding problem on a random lattice $\Lambda(\mathbf{A}, q)$. Regev showed that solving LWE on the average is as hard as approximating SIVP and GapSVP on *any* lattice of dimension n, to within $\tilde{O}(n)$ factors, using a quantum algorithm. He also constructed a public-key cryptosystem based on the LWE problem.

We now state a few important facts about these random lattices that will be used throughout the rest of the paper. The first is a lemma on additive groups going back to [Ajt96]; the tightest version we know of was proved in [Reg05].

Lemma 4.1. *Let q be prime and let $m \geq 2n \lg q$. Then for all but an at most q^{-n} fraction of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the subset-sums of the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$; i.e., for every $\mathbf{u} \in \mathbb{Z}_q^n$ there is a $\mathbf{e} \in \{0, 1\}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{u}$.*[9]

[9] In fact, the lemma is actually stronger, saying that a random subset-sum of $\mathbf{A}$'s columns is statistically close to uniform over $\mathbb{Z}_q^n$ for almost all $\mathbf{A}$.

The next lemma says that an integer error vector taken from an appropriate discrete Gaussian over $\mathbb{Z}^m$ corresponds to a nearly-uniform syndrome. It also characterizes the conditional distribution of the error vector, given its syndrome.

Lemma 4.2. *Assume the columns of* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *generate* $\mathbb{Z}_q^n$*, and let* $\epsilon \in (0, \frac{1}{2})$ *and* $s \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}, q))$*. Then for* $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$*, the distribution of the syndrome* $\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q$ *is within statistical distance* 2ϵ *of uniform over* $\mathbb{Z}_q^n$*.*

Furthermore, fix $\mathbf{u} \in \mathbb{Z}_q^n$ *and let* $\mathbf{t} \in \mathbb{Z}^m$ *be an* arbitrary *solution to* $\mathbf{A}\mathbf{t} = \mathbf{u} \bmod q$*. Then the conditional distribution of* $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$ *given* $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ *is exactly* $\mathbf{t} + D_{\Lambda^\perp,s,-\mathbf{t}}$*.*

Proof. By hypothesis, the set of all syndromes $\{\mathbf{A}\mathbf{e} \bmod q : \mathbf{e} \in \mathbb{Z}^m\} = \mathbb{Z}_q^n$. Now by Lemma 2.7, for $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$ the distribution of $\mathbf{e} \bmod \Lambda^\perp$ is within statistical distance 2ϵ of uniform over the quotient group $(\mathbb{Z}^m/\Lambda^\perp)$. Because this quotient group is isomorphic to the set of syndromes $\mathbb{Z}_q^n$ via the mapping $(\mathbf{e} + \Lambda^\perp) \mapsto \mathbf{A}\mathbf{e} \bmod q$, the first claim follows.

For the second claim, fix $\mathbf{u} \in \mathbb{Z}_q^n$ and consider the distribution D of $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$ given $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$. The support of D is $\mathbf{t} + \Lambda^\perp$, and the distribution is

$$D(\mathbf{e}) = \frac{\rho_s(\mathbf{e})}{\rho_s(\mathbf{t} + \Lambda^\perp)} = \frac{\rho_{s,-\mathbf{t}}(\mathbf{e} - \mathbf{t})}{\rho_{s,-\mathbf{t}}(\Lambda^\perp)} = D_{\Lambda^\perp,s,-\mathbf{t}}(\mathbf{e} - \mathbf{t}).$$

Writing $\mathbf{e} = \mathbf{t} + \mathbf{v}$ for $\mathbf{v} = \mathbf{e} - \mathbf{t}$ distributed as $D_{\Lambda^\perp,s,-\mathbf{t}}$, the claim follows. □

We now show that a random lattice $\Lambda(\mathbf{A}, q)$ has large minimum distance (in ℓ_∞ norm) with overwhelming probability; this implies that $\Lambda^\perp(\mathbf{A}, q)$ has a very small smoothing parameter.

Lemma 4.3. *Let* n *and* q *be positive integers with* q *prime, and let* $m \geq 2n \lg q$*. Then for all but an at most* q^{-n} *fraction of* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$*, we have* $\lambda_1^\infty(\Lambda) \geq q/4$*.*

In particular, for such $\mathbf{A}$ *and any* $\omega(\sqrt{\log m})$ *function, we have* $\eta_\epsilon(\Lambda^\perp(\mathbf{A}, q)) \leq \omega(\sqrt{\log m})$ *for some negligible function* $\epsilon(m)$*.*

Proof. The second claim follows by Lemma 2.5 and the fact that $\Lambda = q \cdot (\Lambda^\perp)^*$.

For the first claim, consider the open ℓ_∞ "cube" $\mathcal{C}$ of radius $q/4$ (hence edge length $q/2$). The set $Z = \mathcal{C} \cap \mathbb{Z}^m$ contains at most $(q/2)^m$ points. Therefore for any fixed nonzero $\mathbf{s} \in \mathbb{Z}_q^n$, the probability over the uniform choice of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ that $\mathbf{A}^T\mathbf{s} = \mathbf{v} \bmod q$ for some $\mathbf{v} \in Z$ is at most $(q/2)^m/q^m = 2^{-m} \leq q^{-2n}$. Taking a union bound over all nonzero $\mathbf{s} \in \mathbb{Z}_q^n$, we conclude that the probability that Λ contains any nonzero point in Z is at most q^{-n}. □

Combining the previous lemmas, we get the following main corollary:

Corollary 4.4. *Let* n *and* q *be positive integers with* q *prime, and let* $m \geq 2n \lg q$*. Then for all but a* $2q^{-n}$ *fraction of all* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and for any* $s \geq \omega(\sqrt{\log m})$*, the distribution of the syndrome* $\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q$ *is statistically close to uniform over* $\mathbb{Z}_q^n$*, where* $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$*.*

Proof. By Lemmas 4.1 and 4.3, for all but a $2q^{-n}$ fraction of all $\mathbf{A}$, the columns of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ generate $\mathbb{Z}_q^n$, and $s \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}, q))$ for some negligible function $\epsilon(m)$. Now by Lemma 4.2, the distribution of $\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. □

4.2 Hard-on-Average Problems

The hard-on-average problem first proposed by Ajtai [Ajt96] is to find a short nonzero integer solution $\mathbf{e} \in \mathbb{Z}^m$ to the homogeneous linear system $\mathbf{A}\mathbf{e} = \mathbf{0} \bmod q$ for uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. This is syntactically equivalent to finding some short nonzero vector in $\Lambda^\perp(\mathbf{A}, q)$. The problem was formalized as follows in [MR07].

Definition 4.5. The *small integer solution* problem SIS (in the ℓ_2 norm) is as follows: given an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and a real β, find a *nonzero* integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{0} \bmod q$ and $\|\mathbf{e}\|_2 \leq \beta$.

For functions $q(n)$, $m(n)$, and $\beta(n)$, $\mathsf{SIS}_{q,m,\beta}$ is the ensemble over instances $(q(n), \mathbf{A}, \beta(n))$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m(n)}$ is uniformly random.

We now define a variant problem, which is to find a short solution to a random *inhomogeneous* system, specifically, $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ (where both $\mathbf{A}$ and $\mathbf{u}$ are uniformly random). As we will see, the difference between the problems is analogous to the difference between SVP and CVP.

Definition 4.6. The *inhomogeneous small integer solution* problem ISIS (in the ℓ_2 norm) is as follows: given an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a syndrome $\mathbf{u} \in \mathbb{Z}_q^n$, and a real β, find an integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ and $\|\mathbf{e}\|_2 \leq \beta$.

The average-case problem $\mathsf{ISIS}_{q,m,\beta}$ is defined similarly, where $\mathbf{A}$ and $\mathbf{u}$ are uniformly random and independent.

The ISIS problem is phrased as a *syndrome decoding* problem, and is equivalent to the problem of decoding an arbitrary integer target point $\mathbf{t} \in \mathbb{Z}^m$ to within distance β on the lattice $\Lambda^\perp = \Lambda^\perp(\mathbf{A}, q)$. Specifically, the target point's syndrome is $\mathbf{u} = \mathbf{A}\mathbf{t} \bmod q$, and solving ISIS on this syndrome yields a short error vector $\mathbf{e} \in \mathbb{Z}^m$ having the same syndrome $\mathbf{u}$. This error vector yields a lattice point $\mathbf{v} = \mathbf{t} - \mathbf{e} \in \Lambda^\perp$, because $\mathbf{A}\mathbf{v} = \mathbf{A}\mathbf{t} - \mathbf{A}\mathbf{e} = \mathbf{0} \bmod q$; furthermore, $\mathbf{v}$ is within distance β of $\mathbf{t}$. Conversely, decoding $\mathbf{t}$ to some $\mathbf{v} \in \Lambda^\perp$ within distance β yields an error vector $\mathbf{e} = \mathbf{t} - \mathbf{v}$ such that $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$. In this work it will be more convenient to work with syndrome decoding, because the lattice $\Lambda^\perp$ is "modded out" of the instances. In other words, an instance of ISIS refers only to a particular coset of $\Lambda^\perp$, and carries no "extraneous" information (unlike target points $\mathbf{t} \in \mathbb{Z}^m$).

Of course, the SIS and ISIS problems are only meaningful if they admit valid solutions for the particular choices of q, m, β. For $\beta \geq \sqrt{m}$ and $m \geq 2n \lg q$ (for some prime q), Lemma 4.1 implies that with overwhelming probability over the choice of $\mathbf{A}$, there is an $\mathbf{e} \in \{0,1\}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ for any $\mathbf{u} \in \mathbb{Z}_q^n$. Then because $\|\mathbf{e}\| \leq \sqrt{m} \leq \beta$, we see that a random instance of $\mathsf{ISIS}_{q,m,\beta}$ has a solution with overwhelming probability. For the same parameters, a pigeonhole argument shows that SIS *always* admits a *nonzero* solution (even for non-prime q, though we will not need this fact). From now on, q, m, and β will always implicitly satisfy the above contraints.

Using Gaussian techniques, Micciancio and Regev [MR07] showed that the $\mathsf{SIS}_{q,m,\beta}$ problem is as hard (on the average) as approximating certain worst-case problems on lattices to within small factors. We give a simpler and slightly tighter proof (also showing hardness for ISIS) that employs our discrete Gaussian sampling algorithm, and which works for a smaller modulus q.

Proposition 4.7. *For any poly-bounded $m, \beta = \mathrm{poly}(n)$ and for any prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the average-case problems $\mathsf{SIS}_{q,m,\beta}$ and $\mathsf{ISIS}_{q,m,\beta}$ are as hard as approximating the problems SIVP and GapSVP (among others) in the worst case to within certain $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$ factors.*[10]

[10] It is possible to base the hardness of ISIS solely on the assumed hardness of SIS, but we only know of such a

Note that Proposition 4.7 gives a "sliding scale" of hardness (and modulus q) depending on the value of β. For the tightest value of $\beta = \sqrt{m}$, we can take $q = \tilde{O}(n)$ and obtain approximation factors $\gamma = \tilde{O}(n)$ for the worst-case problems. However, for our trapdoor functions and other cryptographic primitives, we will need to assume hardness of ISIS for larger values of β (e.g., $\beta \approx m^{1.5} = \tilde{O}(n^{1.5})$). This is because our trapdoor inversion algorithm is only able to produce preimages of length approximately $\sqrt{m}$ times the length of the trapdoor for $\mathbf{A}$; the shorter the trapdoor, the smaller we may take β to be, and the weaker our underlying assumptions can be.

The proof of Proposition 4.7 appears in Section 8.

4.3 Trapdoor Functions

4.3.1 Definitions

We start by defining some enhanced variants of trapdoor functions with *preimage sampling*, which are given by a tuple of probabilistic polynomial-time algorithms ($\mathsf{TrapGen}, \mathsf{SampleDom}, \mathsf{SamplePre}$).

A collection of *trapdoor one-way functions with preimage sampling* satisfies the following:

1. *Generating a function with trapdoor*: $\mathsf{TrapGen}(1^n)$ outputs (a, t), where a is the description of an efficiently-computable function $f_a : D_n \to R_n$ (for some efficiently-recognizable domain D_n and range R_n depending on n), and t is some trapdoor information for f_a.

   For the remaining properties, fix some $(a, t) \leftarrow \mathsf{TrapGen}(1^n)$.

2. *Domain sampling with uniform output*: $\mathsf{SampleDom}(1^n)$ samples an x from some (possibly non-uniform) distribution over D_n, for which the distribution of $f_a(x)$ is uniform over R_n.

3. *Preimage sampling with trapdoor*: for every $y \in R_n$, $\mathsf{SamplePre}(t, y)$ samples from the conditional distribution of $x \leftarrow \mathsf{SampleDom}(1^n)$, given $f_a(x) = y$.

4. *One-wayness without trapdoor*: for any probabilistic poly-time algorithm $\mathcal{A}$, the probability that $\mathcal{A}(1^n, a, y) \in f_a^{-1}(y) \subseteq D_n$ is negligible, where the probability is taken over the choice of a, the target value $y \leftarrow R_n$ chosen uniformly at random, and $\mathcal{A}$'s random coins.[11]

Note that trapdoor *permutations* (with uniform distribution over the domain) satisfy this definition, becase the output distribution is uniform and every point has a unique inverse.

A collection of *trapdoor collision-resistant hash functions with preimage sampling* satisfies the above properties, plus the following:

5. *Preimage min-entropy*: for every $y \in R_n$, the conditional min-entropy of $x \leftarrow \mathsf{SampleDom}(1^n)$ given $f_a(x) = y$ is at least $\omega(\log n)$.

6. *Collision-resistance without trapdoor*: for any probabilistic poly-time algorithm $\mathcal{A}$, the probability that $\mathcal{A}(1^n, a)$ outputs distinct $x, x' \in D_n$ such that $f_a(x) = f_a(x')$ is negligible, where the probability is taken over the choice of a and $\mathcal{A}$'s random coins.

reduction for slightly looser values of β. Because ISIS is interesting on its own, and might even be harder than SIS, we elect to treat it as a separate problem and give a direct, tight reduction from worst-case lattice problems.

[11] This property can be easily adapted to non-uniform adversaries modelled as families of circuits.

We point out that these two additional properties together imply one-wayness (Property 4). For if not, then given a function f_a one could find a collision as follows: choose an $x \leftarrow \mathsf{SampleDom}(1^n)$, and obtain a preimage x' of $f_a(x)$ from the adversarial inverter. Then because x has large min-entropy given $f_a(x)$, we have $x' \neq x$ with overwhelming probability, so x, x' are a collision.

It is also possible to define a trapdoor variant of *universal one-wayness* [NY89], which is implied by collision-resistance. Because our constructions will be collision-resistant anyway, we omit a precise definition.

A collection of *claw-free* pairs of one-way/collision-resistant trapdoor functions (with preimage sampling) is defined similarly, with the following differences: $\mathsf{TrapGen}$ outputs a pair a, a' describing functions $f_a, f_{a'} : D_n \to R_n$ (respectively), and their respective trapdoors t, t'. The preimage sampler works the same way for both f_a (given t) and $f_{a'}$ (given t'). The hardness condition is that no PPT algorithm $\mathcal{A}$, given a, a', can find a pair $x, x' \in D_n$ such that $f_a(x) = f_{a'}(x')$. Each function $f_a, f_{a'}$ may itself also be collision-resistant in the usual way.

A needed relaxation. To be completely precise, the trapdoor functions we construct will actually satisfy a slightly relaxed definition in which the properties are satisfied only *statistically.* First, the properties will hold only with overwhelming probability over the choice of a. Additionally, $\mathsf{SampleDom}(1^n)$ will output an $x \in D_n$ only with overwhelming probability, and the distribution of $f_a(x)$ will only be statistically close to uniform. Finally, $\mathsf{SamplePre}$ will sample from a distribution over the preimages that is statistically close to the prescribed conditional distribution. None of these relaxations will affect security in our applications.

4.3.2 Constructions

Before giving concrete constructions, we need to recall the result of Ajtai [Ajt99] that shows how to sample an essentially uniform $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, along with a short full-rank "trapdoor" set of lattice vectors $\mathbf{S} \subset \Lambda^\perp(\mathbf{A}, q)$.

Proposition 4.8 ([Ajt99])**.** *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 $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a full-rank set $\mathbf{S} \subset \Lambda^\perp(\mathbf{A}, q)$, where the distribution of $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ and the length $\|\mathbf{S}\| \leq L = m^{2.5}$.*

By optimizing the construction and its analysis, the bound L on the length $\|\mathbf{S}\|$ of the trapdoor can be improved to $L = m^{1+\epsilon}$ for any $\epsilon > 0$; we defer the details.

We can now define a collection of trapdoor functions with preimage sampling based on the average-case hardness of SIS and/or ISIS. Let q, m, and L be as in Proposition 4.8. The collection is parameterized by some Gaussian parameter $s \geq L \cdot \omega(\sqrt{\log m})$.

- The function generator uses the algorithm from Proposition 4.8 to choose $(\mathbf{A}, \mathbf{S})$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is statistically close to uniform and $\mathbf{S} \subset \Lambda^\perp(\mathbf{A}, q)$ is a full-rank set with $\|\mathbf{S}\| \leq L$. The matrix $\mathbf{A}$ (and implicitly q) defines the function $f_{\mathbf{A}}(\cdot)$, and the matrix $\mathbf{S}$ is its trapdoor.
- The function $f_{\mathbf{A}}$ is defined as $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \bmod q$, with domain $D_n = \{\mathbf{e} \in \mathbb{Z}^m : \|\mathbf{e}\| \leq s\sqrt{m}\}$ and range $R_n = \mathbb{Z}_q^n$. The input distribution is $D_{\mathbb{Z}^m, s}$, which can be sampled using $\mathsf{SampleD}$ with the standard basis for $\mathbb{Z}^m$.

- The trapdoor inversion algorithm $\mathsf{SampleISIS}(\mathbf{A}, \mathbf{S}, s, \mathbf{u})$ samples from $f_{\mathbf{A}}^{-1}(\mathbf{u})$ as follows: first, choose via linear algebra an arbitrary $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{At} = \mathbf{u} \bmod q$ (such a $\mathbf{t}$ exists for all but an at most q^{-n} fraction of $\mathbf{A}$, by Lemma 4.1). Then sample $\mathbf{v} \leftarrow D_{\Lambda^\perp, s, -\mathbf{t}}$ using the $\mathsf{SampleD}$ algorithm with the trapdoor $\mathbf{S}$, and output $\mathbf{e} = \mathbf{t} + \mathbf{v}$.

Theorem 4.9. *The algorithms described above give a collection of trapdoor* one-way *functions with preimage sampling, if* $\mathsf{ISIS}_{q,m,s\sqrt{m}}$ *is hard on the average.*

Furthermore, they give a collection of trapdoor collision-resistant *hash functions with preimage sampling, if* $\mathsf{SIS}_{q,m,2s\sqrt{m}}$ *is hard on the average.*

Proof. First we note that $s \geq L \cdot \omega(\sqrt{\log m}) \geq \eta_\epsilon(\Lambda^\perp)$ for some negligible $\epsilon(n)$ by Lemma 2.4, because $L \geq \|\mathbf{S}\| \geq \lambda_m(\Lambda^\perp)$.

We start by showing domain sampling. A sample $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, s}$ lands in the domain D_n (except with exponentially small probability), by Lemma 2.8. Furthermore, for all but an exponentially small fraction of $\mathbf{A}$, $f_{\mathbf{A}}(\mathbf{e})$ is statistically close to uniform over R_n, by Corollary 4.4.

We now demonstrate preimage sampling. Because $s \geq \|\mathbf{S}\| \cdot \omega(\sqrt{\log m})$, Lemma 3.5 implies that $\mathsf{SampleD}$ samples from a distribution that is statistically close to $D_{\Lambda^\perp, s, -\mathbf{t}}$. Then by the second claim of Lemma 4.2, $\mathsf{SampleISIS}$ samples from the appropriate conditional distribution.

For one-wayness, inverting a random function $f_{\mathbf{A}}$ on a uniform output $\mathbf{u} \in R_n = \mathbb{Z}_q^n$ is syntactically equivalent to solving $\mathsf{ISIS}_{q,m,s\sqrt{m}}$.

The preimage min-entropy is at least $m-1$; this follows immediately from the fact that preimages are distributed according to a discrete Gaussian (the second claim of Lemma 4.2), and by the min-entropy of the discrete Gaussian (Lemma 2.9).

Finally, for collision-resistance, a collision $\mathbf{e}, \mathbf{e}' \in D_n$ for $f_{\mathbf{A}}$ implies $\mathbf{A}(\mathbf{e} - \mathbf{e}') = \mathbf{0} \bmod q$. Because $\|\mathbf{e} - \mathbf{e}'\| \leq 2s\sqrt{m}$ by the triangle inequality and $\mathbf{e} - \mathbf{e}' \neq \mathbf{0}$ because $\mathbf{e}, \mathbf{e}'$ are distinct, finding a collision in a random $f_{\mathbf{A}}$ implies solving $\mathsf{SIS}_{q,m,2s\sqrt{m}}$. ☐

Claw-free pairs. Constructing a collection of *claw-free* pairs of trapdoor functions is very similar. The function generator produces $(\mathbf{A}, \mathbf{S})$ as above, as well as a uniform $\mathbf{w} \in \mathbb{Z}_q^n$. It outputs a pair of functions $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{Ae} \bmod q$ and $f_{\mathbf{A},\mathbf{w}}(\mathbf{e}) = \mathbf{Ae} + \mathbf{w} \bmod q$. The domains and input distributions are the same as above. The preimage sampler for $f_{\mathbf{A}}^{-1}(\mathbf{u})$ works exactly as above, and the preimage sampler for $f_{\mathbf{A},\mathbf{w}}^{-1}(\mathbf{u})$ samples from the solutions to the system $\mathbf{Ae} = (\mathbf{u} - \mathbf{w}) \bmod q$.

Claw-freeness is based on the average-case hardness of $\mathsf{ISIS}_{q,m,2s\sqrt{m}}$. Given a claw $(\mathbf{e}, \mathbf{e}') \in D_n^2$ for the pair of functions $f_{\mathbf{A}}, f_{\mathbf{A},\mathbf{w}}$, we have $\mathbf{A}(\mathbf{e} - \mathbf{e}') = \mathbf{w} \bmod q$. Because $\|\mathbf{e} - \mathbf{e}'\| \leq 2s\sqrt{m}$ by the triangle inequality, $(\mathbf{e} - \mathbf{e}')$ is a solution to the random ISIS instance $(q, \mathbf{A}, \mathbf{w}, 2s\sqrt{m})$. As above, the functions $f_{\mathbf{A}}$ and $f_{\mathbf{A},\mathbf{w}}$ are both collision-resistant assuming the hardness of $\mathsf{SIS}_{q,m,2s\sqrt{m}}$.

Alternate domains. For some applications, the definition of the domain D_n in terms of the ℓ_2 norm may be inconvenient. In such a case, the domain can also be defined in terms of the ℓ_∞ norm as $D_n = \{\mathbf{e} : \|\mathbf{e}\|_\infty \leq s \cdot \omega(\sqrt{\log m})\}$ for some arbitrary $\omega(\sqrt{\log m})$ function. It can be shown (e.g., using the tail inequality in Section 3.1) that a sample from $D_{\mathbb{Z}^m, s}$ falls in this new domain with overwhelming probability. (Note, though, that inputs still must be chosen from the Gaussian $D_{\mathbb{Z}^m, s}$ over the integers.)

5 Signature Schemes

The hash-and-sign paradigm for signature schemes, first suggested in [DH76], works as follows: the public verification key is a trapdoor function f and the signing key is f^{-1}. To sign a message m, first hash m to some point $y = H(m)$ in the range of a trapdoor function f, then output the signature $\sigma = f^{-1}(y)$. To verify (m, σ), simply check that $f(\sigma) = H(m)$. Bellare and Rogaway [BR93] formalized this notion and showed that this basic scheme, called Full-Domain Hash (FDH), is existentially unforgeable under chosen-message attacks when f is a trapdoor *permutation* and the hash function H is modelled as a random oracle. Many variations on this theme have been proposed, such as the Probabilistic Full-Domain Hash (PFDH) scheme of Coron [Cor00] and the Probabilistic Signature Scheme (PSS) of Bellare and Rogaway [BR96]. These were proposed in part to improve upon the *exact* security of FDH, which is quite loose when instantiated with a black-box trapdoor permutation (the success probability of the reduction is a Q_{hash} factor smaller than that of the forger, where Q_{hash} is the number of hash queries made by the forger).

All of the above schemes were intended to be instantiated with a collection of trapdoor *permutations*, such as RSA. In this section, we show that they can be instantiated equally well using our various notions of trapdoor functions with *preimage sampling*, and retain their tight security reductions in the random oracle model. In fact, we are even able to give a *tight* security reduction for FDH by exploiting *collision resistance*; this stands in constrast to the best known reductions for FDH using trapdoor permutations (for black-box trapdoor permutations, the reduction *must* lose a factor of Q_{hash} [DR02]; for RSA and claw-free permutations, the known reductions still lose a factor of Q_{sign} [Cor00, DR02]). In addition, all of our instantiations are *strongly* unforgeable.

5.1 Full-Domain Hash Scheme

We start with a version of the FDH signature scheme using trapdoor *collision-resistant* hash functions with preimage sampling; recall that such a collection can be constructed assuming that SIS is hard on the average for appropriate parameters. In order for our security reduction to work, the signer must give out *at most one* preimage of a given point. This can be implemented by making the signer *stateful*, or by using a pseudorandom function to implement "repeatable randomness" in a standard way. The PRF is used to generate the random coins of the preimage sampler, so if the sampler's randomness complexity is large, this solution may be impractical and the PFDH scheme below may be a better option. For simplicity, we describe the stateful version of the scheme.

The scheme is built upon a collection of trapdoor collision-resistant hash functions given by $(\mathsf{TrapGen}, \mathsf{SampleDom}, \mathsf{SamplePre})$, and operates relative to a function $H = H_n : \{0,1\}^* \to R_n$ that is modelled as a random oracle (recall that D_n and R_n are the domain and range, respectively, of the collection for security parameter n).

- $\mathsf{SigKeyGen}(1^n)$: let $(a, t) \leftarrow \mathsf{TrapGen}(1^n)$, where a describes a function f_a and t is its trapdoor. The verification key is a and the signing key is t.
- $\mathsf{Sign}(t, m)$: if (m, σ_m) is in local storage, output σ_m. Else, let $\sigma_m \leftarrow \mathsf{SamplePre}(t, H(m))$, store (m, σ_m), and output σ_m.
- $\mathsf{Verify}(a, m, \sigma)$: if $\sigma \in D_n$ and $f_a(\sigma) = H(m)$, accept. Else, reject.

Proposition 5.1. *The scheme described above is SUF-CMA-secure.*

Proof. It is clear that the scheme is complete, by the properties of the trapdoor collection.

Assume, for contradiction, that there is an adversary $\mathcal{A}$ that breaks the existential unforgeability of the signature scheme with probability $\epsilon = \epsilon(n)$. We construct a poly-time adversary $\mathcal{S}$ that breaks the trapdoor collision-resistant hash function with probability negligibly close to ϵ. Given an index a describing a function f_a, $\mathcal{S}$ runs $\mathcal{A}$ on public key a, and simulates the random oracle H and signing oracle as follows. Without loss of generality, assume that $\mathcal{A}$ queries H on every message m before making a signing query on m.

- For every query to H on a distinct $m \in \{0,1\}^*$, $\mathcal{S}$ lets $\sigma_m \leftarrow \mathsf{SampleDom}(1^n)$, stores (m, σ_m), and returns $f_a(\sigma_m)$ to $\mathcal{A}$. (If H was previously queried on m, $\mathcal{S}$ looks up (m, σ_m) and returns $f_a(\sigma_m)$.)
- Whenever $\mathcal{A}$ makes a signing query on m, $\mathcal{S}$ looks up (m, σ_m) in its local storage and returns σ as the signature.

Now without loss of generality, assume that before outputting its attempted forgery (m^*, σ^*), $\mathcal{A}$ queries H on m^*. When $\mathcal{A}$ produces (m^*, σ^*), $\mathcal{S}$ looks up (m^*, σ_{m^*}) in its local storage and outputs (σ^*, σ_{m^*}) as a collision in f_a.

We now analyze the reduction. First, we claim that the view of $\mathcal{A}$ in the real chosen-message attack is identical to its view as provided by $\mathcal{S}$. (This assumes that the trapdoor function properties from Section 4.3.1 are *perfect*; if they are only statistical, the views are statistically close.) For each distinct query m to H, the value returned by $\mathcal{S}$ is $f_a(\sigma_m)$ where $\sigma_m \leftarrow \mathsf{SampleDom}(1^n)$; by the "uniform output" property of the collection, this is identical to the uniformly random value of $H(m) \in R_n$ in the real system. Now fix the value $H(m)$. Then for every signature query on the message m, $\mathcal{S}$ returns a single value σ_m which is distributed as $\mathsf{SampleDom}(1^n)$, given $f_a(\sigma_m) = H(m)$. In the real system, signature queries on m (even repeated ones) are answered by a single value having the same distribution, by the preimage sampling property of $\mathsf{SamplePre}$.

Therefore $\mathcal{A}$ outputs a valid forgery (m^*, σ^*) with probability (negligibly close to) ϵ. Because σ^* is a valid signature on m^*, we have $\sigma^* \in D_n$ and $f_a(\sigma^*) = H(m^*) = f_a(\sigma_{m^*})$. It simply remains to check that $\sigma^* \neq \sigma_{m^*}$, i.e., that they form a collision in f_a. There are two cases to consider:

1. If $\mathcal{A}$ made a signature query on m^*, it received back the signature σ_{m^*}. Because (m^*, σ^*) is considered a forgery, we have $\sigma^* \neq \sigma_{m^*}$.
2. If $\mathcal{A}$ did *not* make a signature query on m^*, then for the query to H on m^*, $\mathcal{S}$ stored a tuple (m^*, σ_{m^*}) for $\sigma_{m^*} \leftarrow \mathsf{SampleDom}(1^n)$, and returned $f_a(\sigma_{m^*})$ to $\mathcal{A}$. By the preimage min-entropy property of the hash family, the min-entropy of σ_{m^*} given $f_a(\sigma_{m^*})$ (and the rest of the view of $\mathcal{A}$, which is independent of σ_{m^*}) is $\omega(\log n)$. Thus, the signature $\sigma^* \neq \sigma_{m^*}$, except with negligible probability $2^{-\omega(\log n)}$.

We conclude that $\mathcal{S}$ outputs a valid collision in f_a with probability negligibly close to ϵ. □

5.2 Probabilistic FDH

The PFDH scheme replaces the statefulness of FDH with a random "salt" for each signature, and is parameterized by the length k of the salt (for simplicity, we can set $k = n$, though any $k = \omega(\log n)$ will suffice for asymptotic security).

- $\mathsf{SigKeyGen}(1^n)$: let $(a, t) \leftarrow \mathsf{TrapGen}(1^n)$, where a describes a function f_a and t is its trapdoor. The verification key is a and the signing key is t.
- $\mathsf{Sign}(t, m)$: choose $r \leftarrow \{0,1\}^k$ at random, let $\sigma \leftarrow \mathsf{SamplePre}(t, H(m\|r))$, and output (r, σ).
- $\mathsf{Verify}(a, m, (r, \sigma))$: if $\sigma \in D_n$ and $r \in \{0,1\}^k$ and $f_a(\sigma) = H(m\|r)$, then accept. Else, reject.

Proposition 5.2. *The scheme described above is SUF-CMA-secure.*

Proof. The proof is almost identical the prior one, so we only give the main idea. Security can be based on either *collision-resistance* as above (which can be based on the hardness of SIS), or on *claw-free pairs* (which can be based on the hardness of ISIS). The essential idea is that repeated signature queries on the same message m will all have distinct salts r (except with negligible probability $Q_{\text{sign}}^2/2^k$), so the signer will provide a preimage for independent hash values $H(m\|r)$. $\square$

6 Trapdoor for Regev's Cryptosystem

In this section we demonstrate a trapdoor for the cryptosystem of Regev based on LWE [Reg05]. We show that this trapdoor enables two powerful applications: *efficient and universally composable oblivious transfer*, and *identity-based encryption*. The first application is the focus of a concurrent work by Peikert *et al.* [PVW07], which relies on the results of this section. The second application requires some additional machinery, which we develop in Section 7.

6.1 Variant Cryptosystem

We now describe a slight variant of Regev's cryptosystem [Reg05] that will be easier to analyze in our applications. Our version differs only in the encryption algorithm. The original algorithm chooses a uniformly random 0-1 vector $\mathbf{e} \in \mathbb{Z}_q^m$ (corresponding to a random subset of $\{1, \ldots, m\}$) and computes the corresponding subset sum $\mathbf{Ae}$ of the columns of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Our encryption algorithm instead chooses a random vector $\mathbf{e}$ from the discrete Gaussian $D_{\mathbb{Z}^m,r}$ over the integer lattice $\mathbb{Z}^m$ (using the SampleD algorithm from Section 3), and computes the syndrome $\mathbf{Ae}$. The particular value of the Gaussian parameter r will be a parameter of the scheme, and for our trapdoor applications will correspond to the length of the trapdoor for $\mathbf{A}$.

The cryptosystem is defined below. We define an optimized version of the system (also described in [Reg05]), in which all users share a common matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ chosen uniformly at random. All operations are performed over $\mathbb{Z}_q$.

- $\mathsf{LWEKeyGen}$: choose a secret decryption key $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random. The public key is the vector $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x} \in \mathbb{Z}_q^m$, where each x_i is chosen independently from the error distribution χ for $i \in [m]$.

  Note that the $\mathbf{p}$ component of the public key can be viewed as a uniform lattice point $\mathbf{A}^T\mathbf{s} \in \Lambda(\mathbf{A}, q) \bmod q$, perturbed by some (small) random error vector $\mathbf{x}$.

- $\mathsf{LWEEnc}(\mathbf{p}, b)$: to encrypt a bit $b \in \{0,1\}$, choose a vector $\mathbf{e} \in \mathbb{Z}^m$ from the distribution $D_{\mathbb{Z}^m,r}$ (using the SampleD algorithm with the standard basis for $\mathbb{Z}^m$), and output the ciphertext $(\mathbf{u}, c) = (\mathbf{Ae}, \mathbf{p}^T\mathbf{e} + b \cdot \lfloor q/2 \rceil) \in \mathbb{Z}_q^{n+1}$.

- $\mathsf{LWEDec}(\mathbf{s}, (\mathbf{u}, c))$: compute $b' = c - \mathbf{s}^T\mathbf{u} \in \mathbb{Z}_q$. Output 0 if b' is closer to 0 than to $\lfloor q/2 \rfloor$ modulo q, otherwise output 1.

Proposition 6.1. *For parameters r, q, m, and α satisfying the hypotheses of Lemmas 6.2 and 6.4, the cryptosystem above is IND-CPA-secure, assuming that $\mathsf{LWE}_{q,\bar{\Psi}_\alpha}$ is hard.*

One possible choice of parameters is to let $m = 6n \lg n$, $r = \log m$, $q \in [\frac{n^2}{2}, n^2]$ be prime, and $\alpha = 1/(\sqrt{m} \cdot \log^2 m)$. It can be verified that these choices satisfy the needed hypotheses. In addition, we have $q \cdot \alpha \geq n$ for all sufficiently large n, so Proposition 2.10 implies that $\mathsf{LWE}_{q,\bar{\Psi}_\alpha}$ is hard unless there are poly-time quantum algorithms for approximating GapSVP and SIVP to within $\tilde{O}(n^{1.5})$ factors.

Proof. The proof is based on a similar line of reasoning as in prior works [AD97, Reg04, Reg05]. For completeness of the cryptosystem, we show in Lemma 6.2 that the decryption algorithm is correct with overwhelming probability over the randomess of $\mathsf{LWEKeyGen}$ and LWEEnc.

We prove semantic security in three steps: first, by assumption on $\mathsf{LWE}_{q,\chi}$, it is immediately apparent that $\mathbf{A}' = (\mathbf{A}, \mathbf{p}^T) \in \mathbb{Z}_q^{(n+1)\times m}$, where $\mathbf{p}$ is the public key generated by $\mathsf{LWEKeyGen}$, is pseudorandom (i.e., indististinguishable from uniform). Second, we describe below the notion of "message-lossy" public keys, whose defining property is that the encryption algorithm *statistically* hides the message bit when encrypting under such keys. Lastly, in Lemma 6.3 we describe an explicit geometric condition that makes $\mathbf{A}'$ message-lossy, and in Lemma 6.4 we show that a uniformly random choice of $\mathbf{A}'$ meets this condition with overwhelming probability. (The last step is the novel part of our proof, and is the only part that uses our modified encryption algorithm.)

Putting everything together, we see that no efficient adversary can distinguish between public keys generated by $\mathsf{LWEKeyGen}$ and those that are message-lossy. Therefore encrypting under keys generated by $\mathsf{LWEKeyGen}$ hides the encrypted bit *computationally*. $\square$

Lemma 6.2 (Completeness)**.** *Let $q \geq 5rm$, let $\alpha \leq 1/(r\sqrt{m} \cdot \omega(\sqrt{\log n}))$, and let $\chi = \bar{\Psi}_\alpha$. Then* LWEDec *decrypts correctly with overwhelming probability (over the random choices of* $\mathsf{LWEKeyGen}$ *and* LWEEnc*).*

Proof. Consider some secret key $\mathbf{s} \in \mathbb{Z}_q^n$ and its public key $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$, for $\mathbf{x} \leftarrow \chi^m$. Now consider a ciphertext

$$(\mathbf{u}, c) = (\mathbf{A}\mathbf{e}, \mathbf{p}^T\mathbf{e} + b \cdot \lfloor q/2 \rfloor) = (\mathbf{A}\mathbf{e}, \mathbf{s}^T\mathbf{A}\mathbf{e} + \mathbf{x}^T\mathbf{e} + b \cdot \lfloor q/2 \rfloor)$$

of a bit b, where $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$. The decryption algorithm computes $c - \mathbf{s}^T\mathbf{u} = \mathbf{x}^T\mathbf{e} + b \cdot \lfloor q/2 \rfloor$, so it outputs b if $\mathbf{x}^T\mathbf{e}$ is at distance at most (say) $q/5$ from 0 (modulo q).

By Lemma 2.8, we have $\|\mathbf{e}\| \leq r\sqrt{m}$ (except with exponentially small probability). Now by definition of $\chi = \bar{\Psi}_\alpha$, we have $x_i = \lfloor q \cdot y_i \rceil \bmod q$, where the y_i are independent normal variables with mean 0 and variance α^2. Then $\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{m}/2$, and by the Cauchy-Schwarz inequality, $\mathbf{x}^T\mathbf{e}$ is at most $rm/2 \leq q/10$ away from $q \cdot (\mathbf{y}^T\mathbf{e} \bmod 1)$. Therefore it suffices to show that $\left|\mathbf{y}^T\mathbf{e}\right| < 1/10$ with overwhelming probability.

Because the y_i are independent, $\mathbf{y}^T\mathbf{e}$ is distributed as a normal variable with mean 0 and standard deviation $\|\mathbf{e}\| \cdot \alpha \leq r\sqrt{m} \cdot \alpha \leq 1/\omega(\sqrt{\log n})$. Therefore by the tail inequality on normal variables, the probability that $\left|\mathbf{y}^T\mathbf{e}\right| > 1/10$ is negligible, and we are done. $\square$

Message-lossy public keys. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be the fixed common matrix and $\mathbf{p} \in \mathbb{Z}_q^m$ be some fixed public key, forming $\mathbf{A}' \in \mathbb{Z}_q^{(n+1)\times m}$ as described above. Define $\delta(\mathbf{p})$ to be the statistical distance between the uniform distribution over $\mathbb{Z}_q^{n+1}$ and the distribution of $\mathbf{A}'\mathbf{e} = (\mathbf{A}\mathbf{e}, \mathbf{p}^T\mathbf{e}) \in \mathbb{Z}_q^{n+1}$, where $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$. If $\delta(\mathbf{p})$ is negligible, we call $\mathbf{p}$ a *message-lossy* public key, because encrypting under $\mathbf{p}$ *loses essentially all information* (statistically) about the encrypted bit. More formally, $\mathsf{LWEEnc}(\mathbf{p}, 0)$ and $\mathsf{LWEEnc}(\mathbf{p}, 1)$ are statistically close, because both distributions are statistically close to uniform. (Of course, the correctness of LWEDec implies that the public keys $\mathbf{p}$ generated by $\mathsf{LWEKeyGen}$ typically have large $\delta(\mathbf{p})$.)

Message-lossy keys (though not named as such) have played a role in the security proofs for several prior lattice-based cryptosystems, e.g., [AD97, Reg04, Reg05]. In all these works, it sufficed to demonstrate that *most* keys are message-lossy, without necessarily identifying their particular characteristics. This was always done via a non-constructive, probabilistic argument.

In the proof of security for the oblivious transfer protocol in [PVW07], the simulator needs to *efficiently identify* message-lossy keys (with the help of a trapdoor). This calls for an *explicit* condition that identifies a public key as message-lossy. A two-part condition suffices: first, the columns of $\mathbf{A}'$ should generate all of $\mathbb{Z}_q^{n+1}$; second, the modular lattice $\Lambda(\mathbf{A}', q)$ should have large minimum distance λ_1^∞ (in the ℓ_∞ norm).

Lemma 6.3 (Message-Lossy Description)**.** *Let* $\mathbf{A}$*,* $\mathbf{p}$*, and* $\mathbf{A}'$ *be as above, and suppose that the columns of* $\mathbf{A}'$ *generate* $\mathbb{Z}_q^{n+1}$*. Then for any* $\epsilon \in (0, \frac{1}{2})$ *and any Gaussian parameter* $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}', q))$ *used by* LWEEnc*, we have* $\delta(\mathbf{p}) \leq 2\epsilon$*.*

In particular, if $r \geq q \cdot \omega(\sqrt{\log m})/\lambda_1^\infty(\Lambda(\mathbf{A}', q))$*, then* $\mathbf{p}$ *is message-lossy.*

Proof. The first claim is a direct consequence of Lemma 4.2 (for dimension $n+1$ instead of n), which implies that $\mathbf{A}'\mathbf{e} = (\mathbf{A}\mathbf{e}, \mathbf{p}^T\mathbf{e})$ is statistically close to uniform over $\mathbb{Z}_q^{n+1}$ for $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$.

The second claim follows directly from Lemma 2.5 and the duality between $\Lambda^\perp$ and Λ. □

Lemma 6.4 (Density of Message-Lossy Keys)**.** *Let* $m \geq 2(n+1)\lg q$ *and let* $r \geq \omega(\sqrt{\log m})$*. Then for all but an at most* $2q^{-n}$ *fraction of* $(\mathbf{A}, \mathbf{p})$*, the public key* $\mathbf{p}$ *is message-lossy for the cryptosystem with common matrix* $\mathbf{A}$*.*

Proof. Let $\mathbf{A}' \in \mathbb{Z}_q^{(n+1)\times m}$ be comprised of $\mathbf{A}$ and $\mathbf{p}$ as above. By an argument identical to the one from Lemma 4.1, the columns of $\mathbf{A}'$ generate $\mathbb{Z}_q^{n+1}$ for all but an at most q^{-n} fraction of all $\mathbf{A}'$. Likewise, by an argument identical to the one from Lemma 4.3, we have $\lambda_1^\infty(\Lambda(\mathbf{A}', q)) \geq q/4$ for all but an at most q^{-n} fraction of all $\mathbf{A}'$. Therefore, for such $\mathbf{A}'$ and for $r \geq \omega(\sqrt{\log m})$, Lemma 6.3 implies that $\mathbf{p}$ is a message-lossy key. □

6.2 Identifying Message-Lossy Keys

We now present an algorithm that identifies (most) message-lossy keys for the above cryptosystem, using a trapdoor for the common matrix $\mathbf{A}$. We start with a high-level overview of the intuition behind the algorithm.

Consider a shared matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and public key $\mathbf{p} \in \mathbb{Z}_q^m$ that together form $\mathbf{A}' \in \mathbb{Z}_q^{(n+1)\times m}$ as above, and recall that $\mathbf{p}$ is message-lossy if: (1) the columns of $\mathbf{A}'$ generate $\mathbb{Z}_q^{n+1}$, and (2) the minimum ℓ_∞ distance $\lambda_1^\infty(\Lambda(\mathbf{A}', q))$ is large enough, so that the Gaussian parameter r used in encryption exceeds the smoothing parameter of the lattice $\Lambda^\perp(\mathbf{A}', q)$. The first condition is easy to check by computing the determinant of the lattice $\Lambda^\perp(\mathbf{A}', q)$. The second condition can be checked using a trapdoor for $\mathbf{A}$ in the following way: first, by Lemma 4.3 we know that with high probability the lattice $\Lambda = \Lambda(\mathbf{A}, q)$ has large minimum distance λ_1^∞. Adjoining the vector $\mathbf{p}$ to Λ yields the lattice $\Lambda' = \Lambda(\mathbf{A}', q)$, which still has large minimum distance λ_1^∞ as long as every nonzero multiple $k \cdot \mathbf{p}$ is far from Λ in ℓ_∞ norm, for $k \in \{1, \dots, q-1\}$. Using techniques of Aharonov and Regev [AR05] (extended to arbitrary ℓ_p norms by Peikert [Pei07]), one can efficiently check that each of the multiples $k \cdot \mathbf{p}$ is far from Λ (in ℓ_∞ norm), using samples from a discrete Gaussian over $\Lambda^* = \Lambda^\perp(\mathbf{A}, q)/q$. The Gaussian sampling algorithm with the trapdoor for $\mathbf{A}$ produces such samples efficiently.

The actual algorithm and its analysis depend crucially on the proper relationship between the cryptosystem's Gaussian parameter r and the length L of the trapdoor. A tighter bound on L means that we can use a smaller value of r and still correctly identify message-lossy keys. A smaller r means that the cryptosystem remains complete for larger values of the LWE error parameter α, which in turn corresponds to weaker assumptions on lattice problems.

We also point out that we only know how to identify message-lossy keys for the *single-bit* cryptosystem, as opposed to the more efficient multi-bit system constructed in [PVW07]. The reason is that a public key for the multi-bit system consists of several vectors $\mathbf{p}_i$, and the key is message-lossy (meaning encryption destroys the *entire* message) if the minimum distance remains large after adjoining *all* the $\mathbf{p}_i$s to the common lattice. Testing for this condition (at least naively) seems to require checking all of the exponentially-many combinations of the $\mathbf{p}_i$s.

We start by recalling the algorithm of [AR05] that distinguishes between points that are far from a lattice and those that are close to it, given access to Gaussian samples over the dual lattice. The meanings of "far" and "close" are determined by the Gaussian parameter s.

Proposition 6.5 ([AR05, Pei07])**.** *There is a deterministic polynomial-time oracle machine $\mathcal{V}^\cdot$ that, on input a basis $\mathbf{B}$ for a full-rank lattice $\Lambda = \mathcal{L}(\mathbf{B}) \subset \mathbb{R}^m$ and a point $\mathbf{x} \in \mathbb{R}^m$, and given access to an oracle that samples from the distribution $D_{\Lambda^*,s}$ for some $s > 0$, has the following behavior:*

- *If* $\mathrm{dist}^\infty(\mathbf{x}, \Lambda) \leq 1/(10s\sqrt{m})$, *then $\mathcal{V}$ outputs "close" with overwhelming probability.*
- *If* $\mathrm{dist}^\infty(\mathbf{x}, \Lambda) \geq \sqrt{\log m}/s$, *then $\mathcal{V}$ outputs "far" with overwhelming probability.*

The probabilities are taken over the samples produced by the oracle.

We now describe our algorithm $\mathsf{IsLossy}$ that identifies message-lossy public keys, given a trapdoor for $\mathbf{A}$. The algorithm has the following properties:

1. For almost all shared matrices $\mathbf{A}$, if $\mathsf{IsLossy}$ outputs "lossy" on a public key $\mathbf{p}$, then $\mathbf{p}$ is indeed message-lossy with overwhelming probability. More precisely, a non-lossy key will only be declared "lossy" with negligible probability over $\mathsf{IsLossy}$'s randomness.
2. For all $\mathbf{A}$, $\mathsf{IsLossy}$ outputs "lossy" on an overwhelming fraction of all public keys $\mathbf{p}$.

Note that $\mathsf{IsLossy}$ has two very different kinds of error. Suppose that $\mathbf{A}$ is "unexceptional," and fix some particular public key $\mathbf{p}$. If $\mathbf{p}$ is "leaky" (non-lossy), then $\mathsf{IsLossy}$ will mistakenly declare it to be lossy with only negligible probability (over its own randomness). On the other hand, if $\mathbf{p}$ is indeed message-lossy, we still have no guarantee — $\mathsf{IsLossy}$ might still might output "not sure" on

$\mathbf{p}$ most of the time. Nevertheless, *most* message-lossy keys are identified as such, and this will be good enough for the applications in [PVW07].

The $\mathsf{IsLossy}$ algorithm is parameterized by a Gaussian parameter $s = L \cdot \omega(\sqrt{\log m})/q$ for some $\omega(\sqrt{\log m})$ function, and is defined as follows: on input $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a full-rank set $\mathbf{S} \subset \Lambda^\perp(\mathbf{A}, q)$ of length $\|\mathbf{S}\| \leq L$, and a public key $\mathbf{p} \in \mathbb{Z}_q^m$,

1. Let $\Lambda = \Lambda(\mathbf{A}, q)$ and $\Lambda^\perp = \Lambda^\perp(\mathbf{A}, q)$. (Recall that $\Lambda^* = \Lambda^\perp/q$.) Below we abuse notation and use Λ and $\Lambda^\perp$ as arguments to algorithms, rather than (arbitrary) bases for them.
2. For each $k \in \{1, \ldots, q-1\}$, run $\mathcal{V}(\Lambda, k \cdot \mathbf{p})$. Whenever $\mathcal{V}$ asks for a sample from $D_{\Lambda^*,s}$, provide it with the output of an independent execution of $\mathsf{SampleD}(\Lambda^\perp/q, \mathbf{S}/q, s, \mathbf{0})$.
3. If every execution of $\mathcal{V}$ outputs "far," then output "lossy." Otherwise, output "not sure."

Lemma 6.6. *Let $r \geq q \cdot s\sqrt{m} \cdot \omega(\sqrt{\log m})$ be the Gaussian parameter used by* LWEEnc*, for some arbitrary $\omega(\sqrt{\log m})$ function. Then for all but a q^{-n} fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, if $\mathbf{p}$ is* not *a message-lossy public key for* LWEEnc*, then $\mathsf{IsLossy}(\mathbf{A}, \mathbf{S}, \mathbf{p})$ errs (i.e., outputs "lossy") with only negligibly small probability.*

Proof. As above, let $\mathbf{A}' \in \mathbb{Z}_q^{(n+1) \times m}$ have the rows of $\mathbf{A}$ as its first n rows and $\mathbf{p}^T$ as its $(n+1)$st row. Let $\Lambda = \Lambda(\mathbf{A}, q)$, $\Lambda' = \Lambda(\mathbf{A}', q)$, and note that $\Lambda' \subseteq \Lambda$.

Now if $\mathbf{p}$ is not message-lossy, by the contrapositive of Lemma 6.3 we have

$$\lambda_1^\infty(\Lambda') < \frac{q \cdot O(\sqrt{\log m})}{r} \leq \frac{1}{10s\sqrt{m}}$$

for all large enough m.

Now for all but a q^{-n} fraction of $\mathbf{A}$, by Lemma 4.3 we have $\lambda_1^\infty(\Lambda) \geq q/4 > \lambda_1^\infty(\Lambda')$. Let $\mathbf{A}$ be such a matrix. Then it must be the case that for some integer k,

$$\mathrm{dist}^\infty(k \cdot \mathbf{p}, \Lambda) = \lambda_1^\infty(\Lambda').$$

Furthermore, because $q \cdot \mathbf{p} \in \Lambda$, such k must be nonzero modulo q, and the above distance is determined by $k \bmod q$. We conclude that there must be some $k \in \{1, \ldots, q-1\}$ such that

$$\mathrm{dist}^\infty(k \cdot \mathbf{p}, \Lambda) \leq \frac{1}{10s\sqrt{m}}.$$

By Proposition 6.5, $\mathcal{V}$ outputs "close" for that choice of k (except with negligible probability), assuming its oracle samples from $D_{\Lambda^*,s}$. Because $\mathbf{S}/q$ is a full-rank set of vectors from $\Lambda^* = \Lambda^\perp(\mathbf{A}, q)/q$ and $s \geq \|\mathbf{S}/q\| \cdot \omega(\sqrt{\log n})$, by Lemma 3.5 the algorithm $\mathsf{SampleD}$ samples from a distribution that is statistically close to $D_{\Lambda^*,s}$, and the claim follows. ∎

We now show that $\mathsf{IsLossy}$ almost always outputs "lossy" for a random choice of public key $\mathbf{p}$ (and arbitrary $\mathbf{A}$).

Lemma 6.7. *Let q be prime, let $m \geq (n+1) \lg q$, and let $\mathbf{A}$, $\mathbf{S}$ be as above where $\|\mathbf{S}\| \leq L$. Then the fraction of $\mathbf{p} \in \mathbb{Z}_q^m$ such that $\mathsf{IsLossy}(\mathbf{A}, \mathbf{S}, \mathbf{p})$ outputs "not sure" with some non-negligible probability is at most*

$$\left(\frac{4\sqrt{\log m}}{q \cdot s}\right)^m.$$

In particular, for any $s = L \cdot \omega(\sqrt{\log m})/q$, IsLossy *outputs "lossy" with overwhelming probability on all but a negligible fraction of $\mathbf{p} \in \mathbb{Z}_q^m$.*

Proof. We proceed by a counting argument. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be arbitrary and let $\Lambda = \Lambda(\mathbf{A}, q)$. Let $\mathcal{C}$ be the open ℓ_∞ cube in $\mathbb{R}^m$ of radius $\sqrt{\log m}/s$. Let $Z = (\Lambda + \mathcal{C}) \cap \mathbb{Z}_q^m$ be the set of integer points in $\mathbb{Z}_q^m$ that are close to Λ in ℓ_∞ norm. By Proposition 6.5, any $\mathbf{p} \in \mathbb{Z}_q^m$ such that IsLossy outputs "not sure" with non-negligible probability is containted in $(k^{-1} \cdot Z) \bmod q$ for some $k \in \{1, \ldots, q-1\}$. (For if not, every multiple $k \cdot \mathbf{p}$ is of ℓ_∞ distance at least $\sqrt{\log m}/s$ from Λ, so IsLossy outputs "lossy" with overwhelming probability.)

Now the number of integer points in $\Lambda \bmod q$ is bounded from above by q^n, and the number of integer points in any translate of $\mathcal{C}$ is bounded from above by $(2\sqrt{\log m}/s)^m$. Therefore the total number of points in $(k^{-1} \cdot Z) \bmod q$ for some k is at most

$$q^{n+1} \cdot \left(\frac{2\sqrt{\log m}}{s}\right)^m \leq \left(\frac{4\sqrt{\log m}}{s}\right)^m.$$

The claim follows by taking the probability over a uniform choice of $\mathbf{p} \in \mathbb{Z}_q^m$. $\square$

6.3 Extracting Secret Keys

We also show that it is possible to extract the secret key $\mathbf{s}$ from a properly-generated public key $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$ for Regev's cryptosystem, using a trapdoor for the shared matrix $\mathbf{A}$. Extracting secret keys is equivalent to solving a bounded-distance (unique) decoding problem on the lattice $\Lambda(\mathbf{A}, q)$. Whereas distinguishing message-lossy keys from properly-generated ones is a *decision* problem, extracting the secret key from a public key is essentially the correspoding *search* problem.

Building on the techniques of Aharonov and Regev [AR05], Liu, Lyubashevsky, and Micciancio [LLM06] gave a deterministic "hill-climbing" algorithm that solves the bounded-distance decoding problem on any lattice, given samples from a discrete Gaussian over the dual lattice. Their algorithm was generalized to all ℓ_p norms in [Pei07].

Proposition 6.8 ([LLM06, Pei07])**.** *There is a deterministic poly-time oracle algorithm that, given:*

- *access to an oracle that samples from the distribution $D_{\Lambda^*,s}$ for some $s > 0$,*
- *a basis $\mathbf{B}$ for a full-rank lattice $\Lambda = \mathcal{L}(\mathbf{B}) \subset \mathbb{R}^m$ such that $\lambda_1^\infty(\Lambda) \geq 10\sqrt{\log m}/s$,*
- *a point $\mathbf{x} \in \mathbb{R}^m$ such that $\mathrm{dist}^\infty(\mathbf{x}, \Lambda) \leq 1/(10s\sqrt{m})$, and*

outputs the unique $\mathbf{y} \in \Lambda$ closest (in ℓ_∞ norm) to $\mathbf{x}$.

As we have already seen, for a random lattice $\Lambda = \Lambda(\mathbf{A}, q)$ with $\Lambda^* = \Lambda^\perp(\mathbf{A}, q)/q$, we can implement the oracle in the above lemma for any $s = L \cdot \omega(\sqrt{\log m})/q$ using the SampleD algorithm with any full-rank set $\mathbf{S} \subset \Lambda^\perp(\mathbf{A}, q)$ such that $\|\mathbf{S}\| \leq L$.

Lemma 6.9. *Let $s = L \cdot \omega(\sqrt{\log m})/q$ as above, and let $\chi = \bar{\Psi}_\alpha$ for $\alpha \leq 1/(q \cdot s\sqrt{m} \cdot \omega(\sqrt{\log m}))$. Let $\Lambda = \Lambda(\mathbf{A}, q)$, and let $\mathbf{p}$ denote a public key produced by* LWEKeyGen*. Then with overwhelming probability over the choice of $\mathbf{A}$ and the randomness of* LWEKeyGen*,*

- *$\lambda_1^\infty(\Lambda) \geq 10\sqrt{\log m}/s$, and*

- $\mathrm{dist}^\infty(\mathbf{p}, \Lambda) \leq 1/(10s\sqrt{m})$

In particular, by Proposition 6.8 one can efficiently extract the secret key $\mathbf{s}$ *from the public key* $\mathbf{p}$ *using a full-rank trapdoor* $\mathbf{S} \subset \Lambda^\perp(\mathbf{A}, q)$ *such that* $\|\mathbf{S}\| \leq L$.

Proof. By Lemma 4.3, for all but an at most q^{-n} fraction of all $\mathbf{A}$, we have $\lambda_1^\infty(\Lambda) \geq q/4 \geq 10\sqrt{\log m}/s$ for sufficiently large m.

Now say $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$ where $\mathbf{x} \leftarrow \chi^m$ is the noise term. By the tail inequality on normal variables and the definition of $\chi = \bar{\Psi}_\alpha$, each coordinate x_i of $\mathbf{x}$ is at distance at most $q \cdot \alpha \cdot g(m)$ from 0 modulo q, with overwhelming probability for any $g(m) = \omega(\sqrt{\log m})$. Therefore for an appropriate choice of $g(m)$ we get $\mathrm{dist}^\infty(\mathbf{p}, \Lambda) \leq 1/(10s\sqrt{m})$. □

Remarks. We point out that Lemma 6.9 applies to both Regev's original scheme and our variant, as it depends only on the distribution of public keys, and not on the distribution of the randomness used in encryption.

We also remark that the above results imply another kind of trapdoor function. Define $f'_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \mathbf{A}^T\mathbf{s} + \mathbf{x}$, where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{x} \leftarrow \chi^m$ is the input distribution. Then for almost all $\mathbf{A}$, the function $f'_{\mathbf{A}}$ has the following properties: (1) with overhwhelming probability over the input $\mathbf{s}, \mathbf{x}$, the output $f_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$ has a unique preimage, in the sense that all other preimages are exponentially unlikely under the input distribution; (2) the unique preimage can be recovered with an appropriate trapdoor for $\mathbf{A}$; (3) assuming that LWE is hard, $f'_{\mathbf{A}}$ is hard to invert, and $(\mathbf{A}, f'_{\mathbf{A}}(\mathbf{s}, \mathbf{x}))$ is indististinguishable from uniform over the choice of $\mathbf{A}$, $\mathbf{s}$, $\mathbf{x}$.

7 Identity-Based Encryption

In this section, we construct an identity-based encryption (IBE) system whose security is based on the hardness of the LWE problem, in the random oracle model. More precisely, it relies on a non-standard "interactive" assumption on the hardness of LWE, in the presence of a signing oracle for our full domain hash signature scheme from Section 5.

In the previous section, we saw that a single lattice $\Lambda(\mathbf{A}, q)$ can support the public keys of many users, and that a trapdoor for $\mathbf{A}$ allows one to extract the secret key from any well-formed public key. At first glance, it may seem that this is all that is needed for an identity-based cryptosystem. However, the situation is not so simple. Well-formed public keys in that system are *exponentially sparse*, because they correspond to points very close to the lattice $\Lambda(\mathbf{A}, q)$. It is not at all clear how a hash function or random oracle could securely map identities to valid public keys.

To remedy this situation, we construct a "dual" of the cryptosystem from the previous section, in which the generation and encryption algorithms are essentially swapped. More specifically, in the dual system, the secret key is a vector $\mathbf{e}$ distributed according to $D_{\mathbb{Z}^m,r}$, and the corresponding public key is its syndrome $\mathbf{u} = f_{\mathbf{A}}(\mathbf{e}) \in \mathbb{Z}_q^n$. The encryption algorithm chooses a pseudorandom LWE vector $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$ (for a uniform secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error vector $\mathbf{x} \leftarrow \chi^m$), and uses the syndrome $\mathbf{u}$ to generate one more LWE instance $p = \mathbf{u}^T\mathbf{s} + x$ as a "pad" to hide the message (where $x \leftarrow \chi$). Because the public key syndrome $\mathbf{u}$ is statistically close to uniform, the adversary's view in the dual system is indistinguishable from uniform, under the hardness of LWE. For the same reason, the scheme (and its identity-based version below) is also *anonymous*; that is, a ciphertext hides the identity to which it was encrypted.

The crucial feature of the dual cryptosystem is that its public keys are *dense*; in fact, every syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ is a valid public key that has many essentially equivalent decryption keys $\mathbf{e} \in \mathbb{Z}_q^m$. We therefore have all the properties we need to implement an IBE system. Identities are hashed to syndromes in $\mathbb{Z}_q^n$, with the assurance that every such syndrome is a well-defined public key for the dual scheme. Furthermore, a trapdoor for $\mathbf{A}$ allows a trusted authority to efficiently sample a secret key $\mathbf{e}$ corresponding to any syndrome $\mathbf{u}$, under the same distribution as in the dual cryptosystem.

We first describe the dual cryptosystem and prove that it is secure based on the hardness of LWE. We then show how to use this to construct an IBE system.

7.1 Dual Cryptosystem

Our public-key dual cryptosystem is defined below. It is parameterized by some $r \geq \omega(\sqrt{\log m})$, which specifies the discrete Gaussian distribution $D_{\mathbb{Z}^m,r}$ from which secret keys are chosen. As in Section 6.1, all users share a common matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ (an implicit input to all algorithms) chosen uniformly at random, which is the index of the function $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \bmod q$ defined in Section 4.3. (Each user may also generate her own matrix $\mathbf{A}$, included in the public key). The trapdoor for $\mathbf{A}$ will not be needed here, and is only used in the IBE below. All operations are performed over $\mathbb{Z}_q$.

- $\mathsf{DualKeyGen}$: Choose an error vector $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ (i.e., the input distribution to $f_{\mathbf{A}}$), which is the secret key. The public key is the syndrome $\mathbf{u} = f_{\mathbf{A}}(\mathbf{e})$.

- $\mathsf{DualEnc}(\mathbf{u}, b)$: to encrypt a bit $b \in \{0, 1\}$, choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly and $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x} \in \mathbb{Z}_q^m$, where $\mathbf{x} \leftarrow \chi^m$. Output the ciphertext $(\mathbf{p}, c = \mathbf{u}^T\mathbf{s} + x + b \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$, where $x \leftarrow \chi$.

- $\mathsf{DualDec}(\mathbf{e}, (\mathbf{p}, c))$: compute $b' = c - \mathbf{e}^T\mathbf{p} \in \mathbb{Z}_q$. Output 0 if b' is closer to 0 than to $\lfloor q/2 \rfloor$ modulo q, otherwise output 1.

Before proving security, we point out an easy extension of the above scheme (which will also apply to our IBE) to encrypt k-bit messages, for any $k = \text{poly}(n)$. The idea (which is very similar to one from [PVW07] for Regev's original system) is to amortize a single $\mathbf{s}$ in the encryption algorithm over *several* syndromes in the public key, thereby encrypting several message bits. The key generator simply chooses k independent secrets $\mathbf{e}_i \leftarrow D_{\mathbb{Z}^m,r}$ for each $i \in [k]$, and publishes all the syndromes $\mathbf{u}_i = f_{\mathbf{A}}(\mathbf{e}_i)$ as the public key. The encryption algorithm chooses $\mathbf{s}$ and $\mathbf{p}$ as above, and encrypts each bit b_i of a k-bit message independently as above using the syndrome $\mathbf{u}_i$.

Setting $k = \Theta(n \log n)$, we get secret/public keys (including $\mathbf{A}$) of size $\tilde{O}(n^2)$ bits, encryption/decryption time of only $\tilde{O}(n)$ bit operations per encrypted bit, and a ciphertext expansion factor of $O(\log n)$. As shown in [KTX07, PVW07], message bits can also be replaced with elements of $\mathbb{Z}_p$, where $p = n^c < q$ for some $c > 0$, at the expense of a smaller noise parameter α (and thus a stronger, though still polynomial, assumption on LWE). This yields an $O(1)$ ciphertext expansion, which can be brought arbitrarily close to 1 for large enough k and constant c. (These measures of complexity are asymptotically the same as those achieved in [PVW07].)

Theorem 7.1. *Let $q \geq 5r(m + 1)$, $\alpha \leq 1/(r\sqrt{m + 1} \cdot \omega(\sqrt{\log n}))$ and $\chi = \bar{\Psi}_\alpha$, and $m \geq 2n \lg q$. Then the above system is IND-CPA-secure and anonymous, assuming that $\mathsf{LWE}_{q,\chi}$ is hard.*

Furthermore, for all but a negligible fraction of shared matrices $\mathbf{A}$, the distribution of public keys generated by $\mathsf{DualKeyGen}$ is statistically close to uniform over $\mathbb{Z}_q^n$.

Proof. The claim on the distribution of public keys follows directly from Corollary 4.4.

We show that $\mathsf{DualDec}$ is correct with overwhelming probability (over the randomness of $\mathsf{DualKeyGen}$ and $\mathsf{DualEnc}$). Consider a ciphertext

$$(\mathbf{p}, c) = (\mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{e}^T\mathbf{A}^T\mathbf{s} + x + b \cdot \lfloor q/2 \rfloor)$$

of a bit b under a public key $\mathbf{u} = \mathbf{A}\mathbf{e}$. The decryption algorithm computes $c - \mathbf{e}^T\mathbf{p} = x - \mathbf{e}^T\mathbf{x} + b \cdot \lfloor q/2 \rfloor$, so it outputs b if $x - \mathbf{e}^T\mathbf{x}$ is at distance at most (say) $q/5$ from 0 (modulo q). By an essentially identical argument as in Lemma 6.2, this occurs with overwhelming probability for our choice of q and α.

Semantic security follows almost immediately from the presumed hardness of LWE. We claim that for a ciphertext $(\mathbf{p}, c)$ of either $b = 0$ or 1, the entire view $(\mathbf{A}, \mathbf{p}, \mathbf{u}, c)$ of the adversary is indistinguishable from uniform, assuming hardness of $\mathsf{LWE}_{q,\chi}$. Indeed, for almost all fixed choices of $\mathbf{A}$ and because $m \geq 2n \lg q$, the public key syndrome $\mathbf{u} = f_{\mathbf{A}}(\mathbf{e})$ is statistically close to uniform by Theorem 4.9. Then the view $(\mathbf{A}, \mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}, \mathbf{u}, c = \mathbf{u}^T\mathbf{s} + x + b \cdot \lfloor q/2 \rfloor)$ simply consists of $m+1$ samples from the LWE distribution $A_{\mathbf{s},\chi}$ (for $\mathbf{s} \leftarrow \mathbb{Z}_q^n$), which are indistinguishable from uniform assuming the hardness of $\mathsf{LWE}_{q,\chi}$. For anonymity, it is enough to see that a ciphertext $(\mathbf{p}, c)$ alone is indistinguishable from uniform, and as such, it computationally hides the particular public key $\mathbf{u}$ under which it was generated.

The proof easily generalized to the multi-bit extension, because each syndrome $\mathbf{u}_i$ is independent and statistically close to uniform (for almost all choices of $\mathbf{A}$). $\square$

7.2 IBE System

Our IBE system uses a random orace $H : \{0,1\}^* \to \mathbb{Z}_q^n$ that maps identities to public keys of the dual cryptosystem, which is instantiated with a Gaussian parameter $r \geq L \cdot \omega(\sqrt{\log m})$ so as to guarantee the preimage sampling property as proved in Theorem 4.9. As with the full-domain hash signature scheme from Section 5, we describe a *stateful* secret key extractor (to prevent a re-querying attack), which can be made stateless via pseudorandom functions in a standard way.

- $\mathsf{IBESetup}(1^n)$: generate $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and trapdoor $\mathbf{S} \subset \Lambda^\perp(\mathbf{A}, q)$ according to the trapdoor generator from Section 4.3. The master public key is $\mathbf{A}$, which is taken as the shared matrix for the dual cryptosystem, and the master secret key is $\mathbf{S}$.

- $\mathsf{IBEExtract}(\mathbf{A}, \mathbf{S}, id)$: if the pair $(id, \mathbf{e})$ is in local storage (from a prior query on id), then return $\mathbf{e}$. Otherwise, let $\mathbf{u} = H(id)$ and choose a decryption key $\mathbf{e} \leftarrow f_{\mathbf{A}}^{-1}(\mathbf{u})$ using the preimage sampler with trapdoor $\mathbf{S}$. Store $(id, \mathbf{e})$ locally and return $\mathbf{e}$.

- $\mathsf{IBEEnc}(\mathbf{A}, id, b)$: to encrypt a bit $b \in \{0,1\}$ to identity id, let $\mathbf{u} = H(id) \in \mathbb{Z}_q^n$, and output a ciphertext $(\mathbf{p}, c) \leftarrow \mathsf{DualEnc}(\mathbf{u}, b)$.

- $\mathsf{IBEDec}(\mathbf{e}, (\mathbf{p}, c))$: Output $\mathsf{DualDec}(\mathbf{e}, (\mathbf{p}, c))$.

A multi-bit IBE follows in the same way from the multi-bit extension of the dual cryptosystem, with the same measures of complexity. Identities are simply mapped by H to multiple uniform syndromes in $\mathbb{Z}_q^n$, one for each bit of the message.

Theorem 7.2. *Suppose that Theorem 7.1 holds, i.e., the dual cryptosystem is IND-CPA-secure and anonymous in the standard model, and that its public keys are statistically close to uniform over $\mathbb{Z}_q^n$ for all but a negligible fraction of shared matrices $\mathbf{A}$.*

Then the IBE system described above is ANON-IND-CPA-secure in the random oracle model.

Proof. First we show completeness. Note that for any identity id and its syndrome $\mathbf{u} = H(id)$, $\mathsf{IBEExtract}$ samples from $f_{\mathbf{A}}^{-1}(\mathbf{u})$ and produces a secret key $\mathbf{e}$ whose distribution is statistically close to that of a secret key for the public key $\mathbf{u}$ in the dual cryptosystem. Therefore IBEDec decrypts correctly as long as $\mathsf{DualDec}$ does. Furthermore, the system is anonymous because IBEEnc simply returns the output of $\mathsf{DualEnc}$.

We now show semantic security in the random oracle model. Let $\mathcal{A}$ be a PPT adversary that attacks the IBE scheme and has advantage ϵ using Q_{hash} distinct queries to H. We will construct an adversary $\mathcal{S}$ that attacks the dual cryptosystem by simulating the view of $\mathcal{A}$, and has advantage negligibly close to ϵ/Q_{hash}.

The adversary $\mathcal{S}$ works as follows. On input a shared matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a public key $\mathbf{u}^* \in \mathbb{Z}_q^n$ for the dual cryptosystem, $\mathcal{S}$ chooses an index $i \leftarrow [Q_{\text{hash}}]$ uniformly at random and simulates the view of $\mathcal{A}$ as follows:

- *Hash queries*: on $\mathcal{A}$'s jth distinct query id_j to H, do the following: if $j = i$, then locally store the tuple $(id_j, \mathbf{u}^*, \perp)$ and return $\mathbf{u}^*$ to $\mathcal{A}$. Otherwise for $j \neq i$, generate a public/secret key pair $(\mathbf{u}_j, \mathbf{e}_j) \leftarrow \mathsf{DualKeyGen}$, locally store the tuple $(id_j, \mathbf{u}_j, \mathbf{e}_j)$, and return $\mathbf{u}_j$ to $\mathcal{A}$.
- *Secret key queries*: when $\mathcal{A}$ asks for a secret key for the identity id, assume without loss of generality that $\mathcal{A}$ already queried H on id. Retrieve the unique tuple $(id, \mathbf{u}, \mathbf{e})$ from local storage. If $\mathbf{e} = \perp$, then output a random bit and abort, otherwise return $\mathbf{e}$ to $\mathcal{A}$.
- *Challenge ciphertext*: when $\mathcal{A}$ produces a challenge identity id^* (distinct from all its secret key queries) and messages m_0, m_1, assume without loss of generality that $\mathcal{A}$ already queried H on id^*. If $id^* \neq id_i$, i.e., if the tuple $(id^*, \mathbf{u}^*, \perp)$ is not in local storage, then output a random bit and abort. Otherwise, relay the messages m_0, m_1 to the challenger, receive a challenge ciphertext c^*, and return c^* to $\mathcal{A}$.

When $\mathcal{A}$ terminates with some output, $\mathcal{S}$ terminates with the same output.

We now analyze the reduction. By a standard argument, the probability that $\mathcal{S}$ does not abort during the simulation is $1/Q_{\text{hash}}$ (this is proved by considering a game in which $\mathcal{S}$ can answer all secret key queries, so that the value of i is perfectly hidden from $\mathcal{A}$). Conditioned on $\mathcal{S}$ not aborting, we claim that the view it provides to $\mathcal{A}$ is statistically close to the view of the real IBE system. Indeed, the answers to the hash queries are independent public keys of the dual cryptosystem, which are statistically close to uniform (for almost all $\mathbf{A}$) by assumption. Furthermore, as we have already seen, the answers to the secret key queries in the real system are statistically close to those generated by $\mathsf{DualKeyGen}$. Finally, we observe that $\mathcal{S}$'s advantage is the same as $\mathcal{A}$'s, conditioned on $\mathcal{S}$ not aborting. ∎

Interactive LWE assumption. Instead of an analysis in the random oracle model, we can also construct an IBE and prove its security under an "interactive" assumption about the hardness of LWE in the presence of a signing oracle for the (stateful) FDH signature scheme from Section 5. A similar "interactive quadratic residuosity assumption" was used for the IBE of [BGH07]. We sketch the assumption and proof of security here.

The interactive $\mathsf{LWE}_{q,\chi}$ problem is this: the input is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ chosen uniformly at random, a vector $\mathbf{p} \in \mathbb{Z}_q^m$, a hash function $H : \{0,1\}^* \to \mathbb{Z}_q^n$, and access to an oracle that, on input z, returns a sample from $f_{\mathbf{A}}^{-1}(H(z))$ (the same value is returned for repeated queries on the same z). The goal is to distinguish whether $\mathbf{p}$ is either an LWE instance or uniform, i.e., between the case that $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$ for some $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{x} \leftarrow \chi^m$, and the case that $\mathbf{p} \leftarrow \mathbb{Z}_q^m$ is uniform. When H is modelled as a random oracle, the interactive LWE problem is hard as long as the standard LWE problem is hard.

The main idea for proving security of the IBE under the interactive assumption is as follows: the simulator is given $\mathbf{A}$, $\mathbf{p}$, hash function H, and access to the signing oracle. It simulates an IBE system having public parameter $\mathbf{A}$ and H, and answers secret key queries by simply querying the signing oracle. When given the challenge identity id^* and challenge message m_0, m_1, the simulator additionally queries the signing oracle on id^* and obtains a secret key $\mathbf{e}^*$. It then constructs a challenge ciphertext by encrypting a random m_b "with the secret key $\mathbf{e}^*$." Specifically, the ciphertext is made up of $\mathbf{p}$ and a "pad" $\mathbf{e}^T\mathbf{p} \in \mathbb{Z}_q$ that hides the message m_b in the standard way. If $\mathbf{p}$ is uniform, it can be shown that the pad is essentially uniform and independent of the other variables, thus the adversary has no advantage. If $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$ is an LWE instance, the ciphertext is distributed as in the real IBE system. Note that the simulated pad $\mathbf{e}^T\mathbf{p} = \mathbf{u}^T\mathbf{s} + \mathbf{e}^T\mathbf{x}$ is correlated with the length $\|\mathbf{x}\|$; therefore, the encryption algorithm in the real IBE system will also choose a pad using an error distribution whose standard deviation is determined by $\|\mathbf{x}\|$.

The full proof of security is somewhat subtle, and is more convenient when the error terms in the LWE problem are *continuous* quantities, not rounded off (this is actually the main form of the problem studied in [Reg05]). When "encrypting with the secret key," the simulator also needs to add a small amount of continuous Gaussian error to the pad $\mathbf{e}^T\mathbf{p}$, in order to ensure that its overall distribution is close to a continuous Gaussian; this technique is also used in the main reduction of [Reg05]. We defer the details.

8 Hardness of SIS and ISIS

The results of this section are very similar to the main worst-case to average-case reduction of Micciancio and Regev [MR07] (which in turn inherits from the original work of Ajtai [Ajt96]). The main difference is the use of our discrete Gaussian sampling algorithm to simplify and slightly tighten the reduction. The discrete sampling algorithm avoids certain complications associated with using *continuous* Gaussian distributions, and the looseness that comes with "rounding off" real-valued samples to nearby lattice points. The net effect is that our reduction works for values of the modulus $q = \tilde{O}(n)$ that are almost linear, versus $\tilde{O}(n^2)$ as shown in [MR07].

We start by introducing a worst-case lattice problem that acts as an intermediary between our average-case decoding problem and more standard problems like SIVP and GapSVP. The new problem is similar to the intermediate *incremental guaranteed distance decoding* (IncGDD) problem defined in [MR07], but has a somewhat simpler formulation.

Definition 8.1 (Incremental Independent Vectors Decoding)**.** An input to $\mathsf{IncIVD}_{\gamma,g}^{\phi}$ is a tuple $(\mathbf{B}, \mathbf{S}, \mathbf{t})$, where $\mathbf{B}$ is a basis for a full-rank lattice in $\mathbb{R}^n$, $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ is a full-rank set of lattice vectors such that $\|\mathbf{S}\| \geq \gamma(n) \cdot \phi(\mathbf{B})$, and $\mathbf{t} \in \mathbb{R}^n$ is a target point. The goal is to output a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{S}\| / g$. (The norm $\|\cdot\|$ is arbitrary, and is implicitly taken to be the Euclidean ℓ_2 norm unless otherwise specified.)

As shown in [MR07], the IncGDD problem is as hard as approximating several other worst-case problems (such as SIVP), via standard worst-case to worst-case reductions. All of these reductions can easily be adapted to work for our problem IncIVD as well. Therefore it will suffice to show that IncIVD reduces to the average-case problem SIS or ISIS for appropriate choices of parameters.

Before stating the main theorem, we observe that SIS can be viewed essentially as a special case of ISIS. An instance of ISIS is given by $(q, \mathbf{A}, \mathbf{u}, \beta)$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$ are uniformly random, and the goal is to find an $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ and $\|\mathbf{e}\| \leq \beta$. Note that $\mathbf{u} \neq \mathbf{0}$ except with q^{-n} probability, so without loss of generality we can assume that a valid solution $\mathbf{e} \neq \mathbf{0}$. Now by instead always setting $\mathbf{u} = \mathbf{0}$ and explicitly requiring $\mathbf{e} \neq \mathbf{0}$, we get exactly the homogeneous SIS problem. Therefore, in the reduction we will choose $\mathbf{u} \in \mathbb{Z}_q^n$ to be either uniform or $\mathbf{0}$ as appropriate, and the analysis will only use the hypothesis that the solution $\mathbf{e} \neq \mathbf{0}$.

Theorem 8.2. *For any $g(n) > 1$ and negligible $\epsilon(n)$, there is a probabilistic poly-time reduction from solving $\mathsf{IncIVD}^{\eta_\epsilon}_{\gamma,g}$ in the worst case for $\gamma(n) = g(n) \cdot \beta(n) \cdot \sqrt{n}$ to solving either $\mathsf{SIS}_{q,m,\beta}$ or $\mathsf{ISIS}_{q,m,\beta}$ on the average with non-negligible probability, for any $q(n) \geq \gamma(n) \cdot \omega(\sqrt{\log n})$ and $m(n), \beta(n) = poly(n)$.*

Using analysis techniques of [Pei07], the theorem can be generalized to solve $\mathsf{IncIVD}^{\eta_\epsilon}_{\gamma,g}$ in *any* ℓ_p norm, $1 \leq p < \infty$, for an approximation factor $\gamma(n) = c_p \cdot g(n) \cdot \beta(n) \cdot n^{1/p}$ where c_p is a fixed constant depending only on p. (For $p = \infty$, the proof goes through with $c_\infty = O(\sqrt{\log n})$.)

Proof of Theorem 8.2. Suppose that oracle $\mathcal{O}$ solves either $\mathsf{ISIS}_{q,m,\beta}$ or $\mathsf{SIS}_{q,m,\beta}$ on the average with non-negligible probability (the difference between the two problems is limited to the first step of the reduction). The reduction that solves $\mathsf{IncIVD}^{\eta_\epsilon}_{\gamma,g}$ works as follows: on input $(\mathbf{B}, \mathbf{S}, \mathbf{t})$,

1. *(Setup.)* Choose an index $j \leftarrow [m]$ and $\alpha \leftarrow \{-\beta, \ldots, -1, 1, \ldots, \beta\}$ uniformly at random.

   Let $\mathbf{c}_j = \mathbf{t} \cdot q/\alpha \in \mathbb{R}^n$, and let $\mathbf{c}_i = \mathbf{0} \in \mathbb{R}^n$ otherwise for $i \in [m]$.

   For reducing to ISIS, choose $\mathbf{u} \leftarrow \mathbb{Z}_q^n$ uniformly and independently.
   For reducing to SIS, let $\mathbf{u} = \mathbf{0} \in \mathbb{Z}_q^n$.

   Let $\mathbf{x}_j = \mathbf{u} \cdot \alpha^{-1} \bmod q$, and $\mathbf{x}_i = \mathbf{0}$ otherwise. Define the matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{Z}_q^{n\times m}$.

2. *(Sample lattice points.)* Let $s = \|\mathbf{S}\| \cdot q/\gamma$. For each $i \in [m]$, let $\mathbf{y}_i \leftarrow D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}_i}$, where the sample is produced using $\mathsf{SampleD}(\mathbf{B}, \mathbf{S}, s, \mathbf{c}_i)$. Define the matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m] \in \mathbb{R}^{n\times m}$.

   Let $\mathbf{A} = (\mathbf{B}^{-1}\mathbf{Y} + \mathbf{X}) \bmod q$.

3. *(Invoke oracle and combine lattice points.)* Invoke oracle $\mathcal{O}$ on $(q, \mathbf{A}, \mathbf{u}, \beta)$, yielding $\mathbf{e} \in \mathbb{Z}^m$. Output the vector $\mathbf{v} = \mathbf{Y}\mathbf{e}/q$.

It is apparent that the reduction runs in time polynomial in n and the size of $(\mathbf{B}, \mathbf{S}, \mathbf{t})$. The correctness of the reduction (with non-negligible probability) will follow from the following claims.

Claim 8.3. *For any values of j, α chosen by the reduction, the distribution of $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$. In particular, $\mathcal{O}$ outputs a nonzero solution $\mathbf{e} \in \mathbb{Z}^m$ (i.e., $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ with $\|\mathbf{e}\| \leq \beta$) such that $e_j = \alpha$ with non-negligible probability.*

Proof. We have $s = \|\mathbf{S}\| \cdot q/\gamma \geq \|\mathbf{S}\| \cdot \omega(\sqrt{\log n})$, so by Lemma 3.5, the distribution sampled by $\mathsf{SampleD}$ is statistically close to $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}_i}$.

By hypothesis, we also have $\|\mathbf{S}\| \geq \gamma \cdot \eta_\epsilon(\mathcal{L}(\mathbf{B}))$, so $s \geq q \cdot \eta_\epsilon(\mathcal{L}(\mathbf{B})) = \eta_\epsilon(q\mathcal{L}(\mathbf{B}))$. Therefore by Lemma 2.7, $\mathbf{y}_i \bmod q\mathcal{L}(\mathbf{B})$ is statistically close to uniform over $\mathcal{L}(\mathbf{B})/q\mathcal{L}(\mathbf{B})$. Hence $\mathbf{a}_i = (\mathbf{B}^{-1}\mathbf{y}_i + \mathbf{x}_i) \bmod q$ is statistically close to uniform over $\mathbb{Z}^n/(q\mathbb{Z})^n = \mathbb{Z}_q^n$. Because the $\mathbf{y}_i$ are independent and $m = \mathrm{poly}(n)$, the entire matrix $\mathbf{A}$ is statistically close to uniform by the triangle inequality. Therefore $\mathcal{O}$ outputs a valid solution $\mathbf{e}$ with non-negligible probability.

Finally, by the discussion above we may assume that the solution $\mathbf{e} \neq \mathbf{0}$, so $\mathbf{e}$ has some nonzero coordinate $e_k \in \{-\beta, \ldots, -1, 1, \ldots, \beta\}$ for some $k \in [m]$. Because $\mathcal{O}$'s input is statistically close to uniform for any values of α, j chosen by the reduction, the probability that $j = k$ and $\alpha = e_k$ is negligibly close to $1/(2\beta m) = 1/\mathrm{poly}(n)$. The claim follows. □

Claim 8.4. *If $\mathbf{e}$ is a valid solution and $e_j = \alpha$, then the output $\mathbf{v} \in \mathcal{L}(\mathbf{B})$.*

Proof. It suffices to show that $\mathbf{B}^{-1}\mathbf{Y}\mathbf{e} \in q\mathbb{Z}^m$. By definition, $\mathbf{B}^{-1}\mathbf{Y} = \mathbf{A} - \mathbf{X} \bmod q$, so it suffices to show that $\mathbf{A}\mathbf{e} = \mathbf{X}\mathbf{e} \bmod q$. Indeed, if $e_j = \alpha$ then $\mathbf{X}\mathbf{e} = \alpha \cdot \mathbf{x}_j = \mathbf{u} \bmod q$. If $\mathbf{e}$ is a valid solution, then $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ as well, as desired. □

Claim 8.5. *If $\mathbf{e}$ is a valid solution and $e_j = \alpha$, then $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{S}\|/g(n)$ with overwhelming probability.*

Proof. If $e_j = \alpha$, we have $\mathbf{t} = \mathbf{C}\mathbf{e}/q$ where $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_m]$. Now for each $\mathbf{y}_i$, let $\mathbf{w}_i = \mathbf{y}_i \bmod q\mathcal{L}(\mathbf{B})$ and define $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m]$. Then conditioned on any fixed value of $\mathbf{w}_i$, $\mathbf{y}_i$ is distributed as $\mathbf{w}_i + D_{q\mathcal{L}(\mathbf{B}),s,\mathbf{c}_i-\mathbf{w}_i}$. The input to the oracle $\mathcal{O}$ depends only on $\mathbf{W}$ and $\mathbf{X}$. Therefore for any fixed $\mathbf{e}$ returned by $\mathcal{O}$, the vector $\mathbf{v} - \mathbf{t} = (\mathbf{Y} - \mathbf{C})\mathbf{e}/q$ is distributed as

$$\frac{1}{q}\left((\mathbf{W} - \mathbf{C})\mathbf{e} + \sum_{i \in [m]} e_i \cdot D_{q\mathcal{L}(\mathbf{B}),s,\mathbf{c}_i - \mathbf{w}_i}\right).$$

Because $s \geq \eta_\epsilon(q\mathcal{L}(\mathbf{B}))$, the summation is distributed essentially as a Gaussian centered at $\mathbf{0}$ with parameter

$$\frac{\|\mathbf{e}\| \cdot s}{q} \leq \frac{\beta \cdot \|\mathbf{S}\|}{\gamma} \leq \frac{\|\mathbf{S}\|}{g \cdot \sqrt{n}},$$

which with overwhelming probability will have length at most $\|\mathbf{S}\|/g$, as desired. A more formal analysis (also for arbitrary ℓ_p norms) can be done using the results of [Pei07] on the sums of discrete Gaussians over lattices. □

By standard repetition techniques, the reduction can be made correct with overwhelming probability. This completes the proof of Theorem 8.2. □

9 Acknowledgments

We thank Vadim Lyubashevsky for pointing out Klein's paper [Kle00], Oded Regev for helpful advice, and Brent Waters for useful observations about our IBE system.

References

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In *ICALP*, pages 1–9, 1999.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610, 2001.

[AR03] Dorit Aharonov and Oded Regev. A lattice problem in quantum NP. In *FOCS*, pages 210–219, 2003.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP $\cap$ coNP. *J. ACM*, 52(5):749–765, 2005.

[Bab86] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

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

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[BGH07] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *FOCS*, 2007. Full version at `http://eprint.iacr.org/2007/177`.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BM92] Mihir Bellare and Silvio Micali. How to sign given any trapdoor permutation. *J. ACM*, 39(1):214–233, 1992.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures - how to sign with rsa and rabin. In *EUROCRYPT*, pages 399–416, 1996.

[CN97] Jin-Yi Cai and Ajay Nerurkar. An improved worst-case to average-case connection for lattice problems. In *FOCS*, pages 468–477, 1997.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[Cor00] Jean-Sébastien Coron. On the exact security of full domain hash. In *CRYPTO*, pages 229–235, 2000.

[Cor02] Jean-Sébastien Coron. Optimal security proofs for pss and other signature schemes. In *EUROCRYPT*, pages 272–287, 2002.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[DR02] Yevgeniy Dodis and Leonid Reyzin. On the power of claw-free permutations. In *SCN*, pages 55–73, 2002.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[GGH96] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *CRYPTO*, pages 112–131, 1997.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *ASIACRYPT*, pages 52–66, 2001.

[HHGP+03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In *CT-RSA*, pages 122–140, 2003.

[JW05] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In *CRYPTO*, pages 293–308, 2005.

[Kle00] Philip N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, pages 937–941, 2000.

[KS06] Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the HB and HB^+ protocols. In *EUROCRYPT*, pages 73–87, 2006.

[KTX07] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit cryptosystems based on lattice problems. In *PKC*, pages 315–329, 2007.

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.

[LLM06] Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On bounded distance decoding for general lattices. In *Proceedings of RANDOM 2006*, August 2006.

[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, 2002.

[Mic04] Daniele Micciancio. Almost perfect lattices, the covering radius problem, and applications to ajtai's connection factor. *SIAM J. Comput.*, 34(1):118–169, 2004.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[MV03] Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO*, pages 282–298, 2003.

[NR06] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In *EUROCRYPT*, pages 271–288, 2006.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43, 1989.

[Pei07] Chris Peikert. Limits on the hardness of lattice problems in ℓ_p norms. In *IEEE Conference on Computational Complexity*, pages 333–346, 2007. Full version in ECCC Report TR06-148.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, pages 145–166, 2006. Full version in ECCC TR05-158.

[PVW07] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. Manuscript, 2007.

[PW07] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. Cryptology ePrint Archive, Report 2007/279, 2007. `http://eprint.iacr.org/2007/279`.

[Reg04] Oded Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.