How to Use a Short Basis: 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

August 25, 2008

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 length of the shortest nonzero vector to within certain polynomial factors). Our contributions include a new notion of *preimage sampleable* functions, simple and efficient "hash-and-sign" digital signature schemes, 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 *discrete Gaussian* probability distribution whose standard deviation is essentially the length of the longest Gram-Schmidt vector of the basis. A 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 recommendations 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. Supported in part by NSF Grant CCF-0635297.

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 *random* keys) on worst-case lattice assumptions. In addition to their unique theoretical niche, lattice-based schemes enjoy many potential advantages: their asymptotic efficiency and conceptual simplicity (usually requiring only linear operations on small integers); their resistance so far to cryptanalysis by *quantum* algorithms (as opposed to schemes based on factoring or discrete log); and the guarantee that their random instances are "as hard as possible."

Until very recently, the known constructions of such primitives were limited mainly to one-way and collision-resistant hash functions [Ajt96, GGH96, CN97, Mic04, MR07] and public-key encryption [AD97, Reg04b, Reg05]. In particular, it has been a longstanding open problem to give a "direct" construction of *digital signatures* having the simplicity and efficiency of other lattice-based primitives, even in the random oracle model.[1] The early "GGH" signature proposal of 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 recover the entire secret key (or its equivalent) from a transcript of signatures.

Moreover, despite some recent advances in lattice-based cryptography (e.g., [PW08, LM08]), many other important cryptographic notions (that were long ago attained under other number-theoretic assumptions) still remain unrealized under lattice assumptions.

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, it was intuitively believed that a *short basis* of a lattice (i.e., a basis in which all the vectors are relatively short) could serve as such a trapdoor. Our central contribution is in showing how to use a short 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 surjective and 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 then give direct lattice-based constructions of richer cryptographic notions, such as signature schemes and identity-based encryption.

A core component in all of our constructions is an efficient algorithm that samples from a so-called *discrete Gaussian* probability distribution over an arbitrary lattice, given an appropriate basis. The sampling algorithm also enables simpler and (slightly) tighter worst-case/average-case connections for lattice problems, and may have additional applications in complexity theory and cryptography.

1.1.1 Gaussian Sampling Algorithm

Because it is the foundation of our cryptographic results, we start by summarizing the Gaussian sampler. The distribution from which it samples is called a *discrete Gaussian* over an n-dimensional lattice Λ.[2] Under such a distribution $D_{\Lambda,s,\mathbf{c}}$, the probability of each vector $\mathbf{v} \in \Lambda$ is proportional to $\exp(-\pi\|\mathbf{v}-\mathbf{c}\|^2/s^2)$,

[1] Indirect (but 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].

[2] An n-dimensional lattice is the set of all integer linear combinations $c_1\mathbf{b}_1 + \cdots + c_n\mathbf{b}_n$ (where each $c_i \in \mathbb{Z}$) of some linearly independent *basis* vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$.

where $\mathbf{c} \in \mathbb{R}^n$ and $s > 0$ are parameters of the distribution akin to its mean and standard deviation, respectively. Discrete Gaussians over lattices are standard in mathematics (see, e.g., [Ban93, Ban95]), and have recently proved to be an exceedingly useful analytical tool in studying the computational complexity of lattice problems [AR03, AR05, Pei07], particularly their worst-case/average-case connections (e.g., [Reg04b, MR07, Reg05]).

The sampling algorithm takes as input the desired parameters $\mathbf{c} \in \mathbb{R}^n$ and $s > 0$, and an arbitrary basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ of the lattice Λ. As long as s exceeds the lengths (times a small extra factor) of all the *Gram-Schmidt*[3] vectors $\tilde{\mathbf{b}_i}$ of the basis $\mathbf{B}$, the output of the algorithm is a lattice vector distributed according to $D_{\Lambda,s,\mathbf{c}}$. In other words, the "width" of the sampled Gaussian is determined by the quality of the input basis. As an alternate perspective, one can view the sampler as a randomized *decoder* that outputs a lattice vector relatively close to $\mathbf{c}$. A key property is that the output distribution depends only on the *maximal length* of $\mathbf{B}$'s Gram-Schmidt vectors; it is otherwise oblivious to $\mathbf{B}$'s particular geometry.

The sampling algorithm itself is actually a simple randomized variant of Babai's "nearest-plane" algorithm [Bab86], which was originally proposed by Klein [Kle00] in another context (see Section 1.2 for details). Instead of determinstically rounding to the nearest plane in each iteration, the algorithm simply chooses a plane with a probability determined by its distance from the target point. While the algorithm itself is not new, we present a (nearly) *exact analysis* of its output distribution (for different parameters than were considered in [Kle00]) using a lattice quantity called the *smoothing parameter*, as defined by Micciancio and Regev [MR07]. As a related contribution, we also bound the smoothing parameter in terms of a quantity that we call the *Gram-Schmidt minimum*; this improves upon a prior bound involving the nth successive minimum [MR07].

As an application of independent interest, we also use the sampling algorithm to give conceptually simpler and slightly tighter worst-case to average-case reductions for lattice problems, building on prior Gaussian techniques [MR07]. Our reduction avoids a "rounding" step that arises when using *continuous* Gaussians, which introduces some looseness in the analysis. While we obtain only modest quantitative improvements, the new reduction and its analysis are technically simpler.

1.1.2 Cryptographic Constructions

Our cryptographic results are summarized as follows (we describe each in more detail below):

- We propose a new abstraction called *preimage sampleable (trapdoor) functions* (PSFs), and present constructions whose security is based on the presumed *worst-case* hardness of standard lattice problems (and whose efficiency is comparable to prior lattice-based cryptographic functions).

- We show that our new abstraction can securely serve as a black-box replacement for trapdoor *permutations* in several prior signature schemes, including those that follow the "hash-and-sign" paradigm (in the random oracle model) [BR93, BR96, Cor02], and a construction of Bellare and Micali (in the plain model) [BM92]. In particular, we obtain simple and efficient "hash and sign" lattice-based signatures, in the random oracle model.

- We construct an asymptotically efficient *identity-based cryptosystem* (in the random oracle model, or under an "interactive" assumption) based on *learning with errors* (LWE), a bounded-distance decoding problem on lattices that generalizes the well-known "learning parity with noise" problem. As shown by

[3] The Gram-Schmidt vectors are defined iteratively: $\tilde{\mathbf{b}_1} = \mathbf{b}_1$, and $\tilde{\mathbf{b}_i}$ is the component of $\mathbf{b}_i$ orthogonal to $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$ for $i = 2, \ldots, n$. In particular, note that $\|\tilde{\mathbf{b}_i}\| \leq \|\mathbf{b}_i\|$.

Regev [Reg05], the average-case hardness of LWE can be based on the presumed *worst-case* hardness of standard lattice problems for *quantum* algorithms.

- We present (unconditionally) some trapdoor techniques for the LWE problem and cryptosystems based upon it. A concurrent work [PVW07] applies these techniques to instantiate a general framework for efficient and universally composable oblivious transfer.

The worst-case problems underlying our cryptographic schemes are to approximate the *shortest independent vectors problem* SIVP or the *shortest vector problem* GapSVP (in its decision version) to within small polynomial (in the dimension n) factors. Known classical (and quantum) algorithms for these problems require time and space that are exponential in n [AKS01], and known polynomial-time algorithms obtain approximation factors that are essentially exponential in n [LLL82, Sch87].

In all of our constructions, we need to generate a *"hard" public* basis $\mathbf{B}$ (chosen at random from some appropriate distribution) of some lattice Λ, together with a *"good" trapdoor* basis $\mathbf{T}$ of Λ whose Gram-Schmidt vectors are relatively short (this is used as the advice for the sampling algorithm). Although our techniques are entirely orthogonal to the method for generating such bases, our preferred approach comes from a lesser-known paper of Ajtai [Ajt99], which describes a way to generate such bases so that the random public basis has worst-case hardness. As far as we know, our results are the first applications of Ajtai's generator in cryptography or otherwise.

***Preimage sampleable* (trapdoor) functions.** The basic object underlying our higher-level cryptographic tools is a collection of special one-way (and even collision-resistant) trapdoor functions, which we call *preimage sampleable functions* (PSFs). Intuitively, evaluating a public function $f = f_{\mathbf{B}}$ from the collection (where $\mathbf{B}$ is the public basis for Λ) on a random input corresponds to choosing a lattice point $\mathbf{v} \in \Lambda$ "uniformly at random" and perturbing it by some relatively short error term $\mathbf{e}$, yielding a point $\mathbf{y} = \mathbf{v} + \mathbf{e}$.[4] Inverting $\mathbf{y}$ corresponds to *decoding* it to any sufficiently nearby lattice point $\mathbf{v}' \in \Lambda$, though not necessarily the original $\mathbf{v}$; the error term is large enough that many preimages exist. Given the trapdoor basis $\mathbf{T}$, it is easy to decode $\mathbf{y}$ using the sampling algorithm. But given only the public basis $\mathbf{B}$, the decoding problem is hard (on the average, for the particular distribution of $\mathbf{B}$ and $\mathbf{y}$).

Our trapdoor functions have two crucial properties for security in cryptographic applications. First, the random input (the error term $\mathbf{e}$) is drawn from a relatively narrow Gaussian distribution, and under this distribution, the output $\mathbf{y}$ is statistically close to *uniform* over the range. Second, the trapdoor inversion algorithm does not just find an *arbitrary* preimage of $\mathbf{y}$, but actually *samples* from among all its preimages under the appropriate conditional distribution, i.e., a discrete Gaussian over Λ. In other words, the inverter samples an input $\mathbf{e}$ from the Gaussian input distribution, conditioned on the event $f(\mathbf{e}) = \mathbf{y}$.

The properties described above 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, these properties make PSFs "as good as" trapdoor *permutations* in certain applications.

Signature schemes. The cryptographic literature contains several existentially unforgeable digital signature schemes based on trapdoor *permutations*. Using the "hash-and-sign paradigm" [DH76, RSA78] in the random oracle model, we have the simple and efficient full-domain hash (FDH) scheme [BR93] and its variants [BR96, Cor02]. In the plain model, there is a tree-based scheme of Bellare and Micali [BM92] that,

[4] Of course, as an infinite set, the lattice Λ cannot support a uniform distribution. Formally, f applies the standard technique of reducing a random error term $\mathbf{e}$ modulo the public basis $\mathbf{B}$.

while somewhat inefficient, has significantly shorter signatures than generic constructions based on one-way or universal one-way functions [NY89, Rom90].

We show that all of the above permutation-based signature schemes can also be instantiated using (as a black box) any collection of preimage sampleable functions, and retain their security analyses in their respective models (though subtleties can arise when signing the same message more than once). In fact, by relying on collision-resistant PSFs, we are able to give *tight* security reductions for FDH (and its variants), whereas reductions for plain FDH based on trapdoor *permutations* are inherently loose [Cor02]. Similarly, we also give a much tighter reduction for the scheme of Bellare and Micali.

Concretely, our hash-and-sign schemes represent a more theoretically sound way of instantiating the original (but insecure) GGH proposal [GGH97] and its variants, e.g., NTRUSign [HHGP$^+$03]. Informally, in these schemes a message is hashed to a point in some region of space, and its signature is essentially a nearby lattice point, which is found using a "good" secret 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 geometry of the secret basis. (Recall that the original GGH proposal is insecure precisely because its signatures leak information about the "shape" of the trapdoor basis [NR06].)

Trapdoors for learning with errors. Our next two applications are centered around the *learning with errors* (LWE) problem, as defined by Regev [Reg05]. We observe that LWE is essentially a bounded-distance decoding problem on the *dual lattice* Λ^* of Λ, where as above, Λ is a random lattice having public basis $\mathbf{B}$ (and trapdoor basis $\mathbf{T}$). Using this interpretation, the goal of LWE is to decode a randomly-chosen lattice vector $\mathbf{w} \in \Lambda^*$ that has been perturbed to a point $\mathbf{p}$ by some small amount of noise. The perturbation is small enough that $\mathbf{w}$ is the unique vector closest to $\mathbf{p}$ (with overwhelming probability).

In an optimized version of Regev's LWE-based cryptosystem [Reg05], the same dual lattice Λ^* is shared among all users, and public keys are perturbed points $\mathbf{p}$ as above. Security is demonstrated by showing that such public keys are indistinguishable from so-called "messy" public keys, whose ciphertexts carry *no information* about the encrypted messages. As in prior lattice-based cryptosystems [AD97, Reg04b], this security argument is probabilistic and non-constructive.

A concurrent work of Peikert, Vaikuntanathan, and Waters [PVW07] uses cryptosystems that have messy public keys to instantiate a framework for efficient oblivious transfer. However, the framework requires a way to *identify* messy keys efficiently, given some master trapdoor for the cryptosystem. In this work, we give an *explicit geometric* description of messy keys in Regev's cryptosystem, and a way of efficiently identifying them. Essentially, a public key $\mathbf{p}$ is messy if the minimum distance of the dual lattice Λ^* remains large after adjoining $\mathbf{p}$ to it. To identify such keys, we use the Gaussian sampling algorithm with the trapdoor basis $\mathbf{T}$ of Λ to implement the preprocessing phase of an algorithm of Aharonov and Regev [AR05]. Using an extension of this algorithm due to Liu, Lyubashevsky, and Micciancio [LLM06], we also show how to recover the secret key $\mathbf{w} \in \Lambda^*$ from any properly-generated public key $\mathbf{p}$, i.e., we show how to solve LWE using a master trapdoor.

Identity-based encryption. In identity-based encryption (IBE), proposed by Shamir [Sha84], any string can serve as a public key, and secret keys are administered by an authority who knows some master secret key of the system. Thus far, IBE has been realized under various assumptions relating to groups with bilinear pairings (e.g., [BF03, BB04, Wat05]), and under the quadratic residuosity (QR) assumption in the random oracle model or an "interactive" QR assumption in the plain model [Coc01, BGH07].

Our final application is an efficient (and "anonymous") IBE based on LWE in the random oracle model

(or in the plain model under an interactive LWE assumption). Although secret keys can be extracted from public keys using a master trapdoor as described above, 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 Λ^*. 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 public-key cryptosystem, in which the key generation and encryption algorithms are effectively swapped: public keys belong to the "primal space" containing Λ, and encryption is performed in the "dual space" containing Λ^*. In the resulting system, *every* point of the primal space is a valid public key having many equivalent secret keys, which are simply the nearby lattice points in Λ. Using the Gaussian decoder with the trapdoor basis $\mathbf{T}$ of Λ, the authority can extract a (properly-distributed) secret key from any public key. (In fact, extracting a secret key for an identity is entirely equivalent to signing that identity under the FDH signature scheme.)

Because it uses a trapdoor for extracting secret keys, our IBE is structurally closest to those based on quadratic residuosity [Coc01, BGH07]. It is the most efficient IBE to date, at least in an asymptotic sense: for messages of length $n \log n$ (where n is the security parameter), the amortized encryption and decryption times are only $\tilde{O}(n)$ per message bit, and the ciphertext expansion factor can be made as small as $O(1)$. One possible drawback of our system is that the master public key and individual secret keys are $\tilde{O}(n^2)$ bits. As a point of comparison, the recent QR-based IBE of Boneh, Gentry, and Hamburg [BGH07] has essentially optimal *additive* ciphertext expansion of $O(n)$ bits (where n is the size of the master public modulus $N = pq$), but the encryption and decryption times are $O(n^4)$ and $O(n^3)$ per message bit, respectively.

1.2 Related Work

The randomized nearest-plane algorithm we use for Gaussian sampling was originally proposed by Klein [Kle00] for solving a variant of the closest vector problem, in which the target point is guaranteed to be "unusually close" to the lattice. Klein's analysis is focused on the case where the parameter s is no more than (a small factor times) the length of the *shortest* Gram-Schmidt vector of the input basis; for such parameters, the output distribution is concentrated on the unique closest lattice vector, but may be quite far from a discrete Gaussian. A preliminary version of [NV08] showed that the output distribution is "quasi-Gaussian" when s is at least the length of the *longest* Gram-Schmidt vector; our analysis essentially subsumes that analysis.

Independently of our work, Lyubashevsky and Micciancio [LM08] gave a direct lattice-based construction of a *one-time* signature scheme that can sign $O(n)$-bit messages in $\tilde{O}(n)$ time. The functionality and security of the scheme both rely on special classes of *cyclic/ideal* lattices having algebraic structure, which were studied previously in [Mic07, PR06, LM06, PR07]. A full signature scheme having comparable asymptotic efficiency can be obtained by incorporating the one-time scheme into a tree-based construction.

Several works have given tight security reductions for FDH-like signatures based on variants of trapdoor permutations or specific number-theoretic assumptions. Coron [Cor00] improved the exact security of FDH for its concrete instantiation with RSA. Dodis and Reyzin [DR02] presented tight reductions for probabilistic FDH (PFDH) based on any collection of *claw-free* pairs of trapdoor permutations. Katz and Wang [KW03] gave a tight reduction based on claw-free pairs for PFDH with only one bit of salt. Bernstein [Ber08] recently gave a tight reduction for a concrete instantiation of FDH with Rabin-Williams signatures. We point out that claw-free pairs of trapdoor permutations can be viewed as a special case of collision-resistant PSFs from $n + 1$ bits to n bits, where the extra input bit indicates which of the two permutations is evaluated on the remaining bits.

Using entirely different techniques, Peikert and Waters [PW08] constructed complementary collections of *injective* trapdoor functions based on LWE (among other assumptions). Their TDFs imply several

cryptographic primitives, most notably *chosen ciphertext-secure* encryption, but have exponentially-sparse images that seem less well-suited toward applications like signature schemes and IBE. From a purely aesthetic point of view, our trapdoor functions also correspond more directly to "natural" lattice problems.

1.3 Open Problems

Many interesting questions arise from our work. The most important problem, in our view, is to construct a simple and efficient lattice-based signature scheme without using tree structures or a random oracle. Even under other strong number-theoretic assumptions, only a few such schemes are known (e.g., [GHR99, CS00]), so this problem appears quite challenging. A related problem is to construct an IBE without a random oracle under standard lattice assumptions (recall that our IBE can be based on a non-standard "interactive" LWE assumption in the plain model).

Another important direction is to obtain more efficient cryptographic schemes based on *ideal* (e.g., cyclic) lattices, as in prior works [Mic07, PR06, LM06, PR07, LM08]. Most of our techniques apply equally well to ideal lattices; two main technical hurdles are to generate appropriate random lattices with good trapdoor bases, and to demonstrate a hard decoding problem analogous to LWE.

The concrete security of our schemes (i.e., the approximation factor obtained by the worst-case/average-case reduction) is determined by the Gaussian parameter of the sampling algorithm, which in turn depends on the quality of the trapdoor basis. It is therefore important to optimize Ajtai's trapdoor generator [Ajt99] and its analysis, as well as to seek other Gaussian sampling algorithms that might work for smaller parameters s (perhaps given different advice).

A final interesting problem is to construct a lattice-based IBE having security under *chosen-ciphertext attack* (CCA security). The techniques of [PW08] for obtaining CCA security in lattice-based public-key cryptosystems are quite different from ours, and do not appear to be immediately applicable to our IBE. Combining the two approaches seems to be a worthy goal.

1.4 Organization and Reader's Guide

For the reader interested mainly in our cryptographic constructions (but who may not be familiar with recent work on lattices), we recommend starting with the background in Sections 2.3 and 2.4, and the statements of Lemma 3.1 and Theorem 4.1. We then suggest moving directly to Section 5, which contains background on hard random lattices, our abstract definition of trapdoor functions with preimage sampling, and concrete instantiations. The signature schemes in Section 6 can be fully understood based only on the abstract definition (found in Section 5.3.1). To understand our IBE construction in Section 7, we recommend first reading the LWE background in Section 2.5 and understanding the concrete lattice-based trapdoor functions of Section 5. The trapdoor techniques for LWE in Section 8 are the most technical and rely on results from other recent works, but can be understood after absorbing the background on LWE and the details of Section 5.1.

For the reader interested more in the new analytic and algorithmic results for lattices (but who may not be as interested in the cryptographic applications), we recommend reading (in order) the new smoothing parameter bound in Section 3, the analysis of the Gaussian sampling algorithm in Section 4, and the simplified worst-case to average-case reduction in Section 9.

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 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 countable 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 is extended to non-uniform families of poly-sized circuits in the standard way.

2.2 Cryptographic Notions

For signature schemes, we use the standard notion of existential unforgeability under chosen-message attack due to Goldwasser, Micali, and Rivest [GMR88]. We actually only use the stricter notion of *strong* unforgeability in which an adversary cannot even produce a new signature for any message on which it queried its signing oracle. For public-key encryption, we use the standard definition of indistinguishability under a chosen-plaintext eavesdropping attack (semantic security) [GM84].

For identity-based encryption (IBE), we use the standard definition of security under chosen-plaintext and chosen-identity attack [BF03, ABC$^+$05], which we summarize here. An IBE consists of the following four algorithms.

- A setup algorithm $\mathsf{IBESetup}$ that outputs a master public key mpk and master secret key msk.
- A secret key extraction algorithm $\mathsf{IBEExtract}$ that, given msk and an identity id, outputs a secret key sk for that identity.
- An encryption algorithm IBEEnc that, given the master public key mpk, an identity id, and a message m, outputs a ciphertext c.
- A decryption algorithm IBEDec that, given a secret key sk and a ciphertext c, outputs a message m.

The completeness condition is that for all identities id, IBEDec correctly decrypts a ciphertext encrypted to id, given the sk for id produced by $\mathsf{IBEExtract}$. Security is defined by a game in which the adversary is given mpk and access to an oracle computing $\mathsf{IBEExtract}$. The adversary produces a challenge identity id^*

and two valid messages m_0, m_1, is given an encryption of m_b under id^* for $b \leftarrow \{0,1\}$ chosen uniformly at random, and attempts to guess the value of b without ever querying its oracle on identity id^* (queries can be made both before and after the adversary produces id^*, m_0, m_1). We say that the scheme is secure if every PPT adversary succeeds with probability at most negligibly more than $1/2$.

2.3 Lattices

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$ consist of n linearly independent vectors. The n-dimensional *lattice*[5] Λ generated by the *basis* $\mathbf{B}$ is

$$\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bc} = \sum\nolimits_{i \in [n]} c_i \cdot \mathbf{b}_i \ : \ \mathbf{c} \in \mathbb{Z}^n\}.$$

The *minimum distance* $\lambda_1(\Lambda)$ of a lattice Λ is the length (in the Euclidean ℓ_2 norm, unless otherwise indicated) of its shortest nonzero vector: $\lambda_1(\Lambda) = \min_{0 \neq \mathbf{x} \in \Lambda} \|\mathbf{x}\|$. 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 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 additive group of distinct *cosets* $\mathbf{v} + \Lambda'$ for $\mathbf{v} \in \Lambda$, with addition of cosets defined in the usual way.

For any (ordered) set $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathbb{R}^n$ of linearly independent vectors, let $\tilde{\mathbf{S}} = \{\tilde{\mathbf{s_1}}, \ldots, \tilde{\mathbf{s_n}}\}$ denote its Gram-Schmidt orthogonalization, defined iteratively in the following way: $\tilde{\mathbf{s_1}} = \mathbf{s}_1$, and for each $i = 2, \ldots, n$, $\tilde{\mathbf{s}_i}$ is the component of $\mathbf{s}_i$ orthogonal to $\mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{i-1})$. Clearly, $\|\tilde{\mathbf{s}_i}\| \leq \|\mathbf{s}_i\|$. The following useful lemma says that any full-rank set of vectors in a lattice can be efficiently converted to a *basis* of the lattice, without increasing the lengths of the Gram-Schmidt vectors.

Lemma 2.1 ([MG02, Lemma 7.1, page 129])**.** *There is a deterministic polynomial-time algorithm that, given an arbitrary basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$ and a full-rank set of lattice vectors $\mathbf{S} \subset \Lambda$, outputs a basis $\mathbf{T}$ of Λ such that $\|\tilde{\mathbf{t}_i}\| \leq \|\tilde{\mathbf{s}_i}\|$ for all $i \in [n]$.*

The *dual lattice* of Λ, denoted Λ^*, is defined to be $\Lambda^* = \{\mathbf{x} \in \mathbb{R}^n \ : \ \forall\ \mathbf{v} \in \Lambda, \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}\}$. By symmetry, it can be seen that $(\Lambda^*)^* = \Lambda$. If $\mathbf{B}$ is a basis of Λ, it can be seen that the dual basis $\mathbf{B}^* = (\mathbf{B}^{-1})^T$ is in fact a basis of Λ^*. The following standard fact relates the Gram-Schmidt orthogonalizations of a basis and its dual (a proof can be found in [Reg04a, Lecture 8]).

Lemma 2.2. *Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be an (ordered) basis, and let $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ be its dual basis in reversed order (i.e., $\mathbf{d}_i = \mathbf{b}^*_{n-i+1}$). Then $\tilde{\mathbf{d}_i} = \tilde{\mathbf{b}_i}/\|\tilde{\mathbf{b}_i}\|^2$ for all $i \in [n]$. In particular, $\|\tilde{\mathbf{d}_i}\| = 1/\|\tilde{\mathbf{b}_i}\|$.*

We recall two standard worst-case approximation problems on lattices. In both problems, $\gamma = \gamma(n)$ is the approximation factor as a function of the dimension.

Definition 2.3 (Shortest Vector Problem (Decision Version))**.** An input to GapSVP_γ is a basis $\mathbf{B}$ of a full-rank n-dimensional lattice. It is a YES instance if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq 1$, and is a NO instance if $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n)$.

Definition 2.4 (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 $\|\mathbf{S}\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$.

[5] Technically, this is the definition of a *full-rank* lattice, which is all we will be concerned with in this work.

2.4 Gaussians on Lattices

Our review of Gaussian measures over lattices follows the development by prior works [Reg04b, 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, 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 merely a normalization factor; the probability $D_{\Lambda,s,\mathbf{c}}(\mathbf{x})$ is simply proportional to $\rho_{s,\mathbf{c}}(\mathbf{x})$.

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

Definition 2.5 ([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 $\rho_{1/s}(\Lambda^* \backslash \{\mathbf{0}\}) \leq \epsilon$.

In this paper we use two bounds on the smoothing parameter. The first relates the smoothing parameter of a lattice to the minimum distance of its dual lattice, *in the ℓ_∞ norm*. We note that the smoothing parameter can also be related to the dual minimum distance in the ℓ_2 norm, as shown in [MR07, Lemma 3.2]. However, the ℓ_∞ norm turns out to be easier to analyze for the random lattices we use, and also yields the tightest bounds on their smoothing parameters.

Lemma 2.6 ([Pei07], using [Ban95])**.** *For any n-dimensional lattice Λ and real $\epsilon > 0$, we have*

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

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

The second bound on the smoothing parameter is new to this work; it relates the smoothing parameter to the longest Gram-Schmidt vector in any basis of the lattice. See Section 3 for a precise statement and proof.

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 2.7 ([MR07], implicit in Lemma 4.4)**.** *Let Λ be any n-dimensional lattice. Then for any $\epsilon \in (0,1)$, $s \geq \eta_\epsilon(\Lambda)$, and $\mathbf{c} \in \mathbb{R}^n$, 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', if $s \geq \eta_\epsilon(\Lambda')$.

Corollary 2.8. *Let Λ, Λ' be n-dimensional lattices, 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. 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 $\mathbf{z} \in \mathbf{v} + \Lambda'$ 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 2.7. By a routine calculation, it follows that for every $\mathbf{v} \in \Lambda$, $\Pr_{\mathbf{z}}[\mathbf{z} = \mathbf{v} \bmod \Lambda']$ is in the range $(1 \pm 4\epsilon)/\left|\Lambda/\Lambda'\right|$, 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 the ℓ_2 norm), with overwhelming probability.

Lemma 2.9 ([MR07, Lemma 4.4])**.** *For any n-dimensional lattice Λ, $\mathbf{c} \in \mathrm{span}(\Lambda)$, real $\epsilon \in (0,1)$, and $s \geq \eta_\epsilon(\Lambda)$,*

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

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.

Lemma 2.10 ([PR06])**.** *For any n-dimensional lattice Λ, center $\mathbf{c} \in \mathbb{R}^n$, positive $\epsilon > 0$, and $s \geq 2\eta_\epsilon(\Lambda)$, and for every $\mathbf{x} \in \Lambda$, we have*

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

In particular, for $\epsilon < \frac{1}{3}$, the min-entropy of $D_{\Lambda,s,\mathbf{c}}$ is at least $n - 1$.

2.5 Learning with Errors

We now review the *learning with errors* (LWE) problem, 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 errors (LWE). For an integer $q = q(n)$ and a distribution χ on $\mathbb{Z}_q$, the goal of the (average-case) *learning with errors* 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 2.11 ([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.*

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

3 New Smoothing Parameter Bound

Here we give a new bound on the smoothing parameter relative to a certain lattice quantity. For a lattice Λ, define the *Gram-Schmidt minimum* as

$$\tilde{bl}(\Lambda) = \min_{\mathbf{B}} \|\tilde{\mathbf{B}}\| = \min_{\mathbf{B}} \max_{i \in [n]} \|\tilde{\mathbf{b}_i}\|,$$

where the minimum is taken over all (ordered) bases $\mathbf{B}$ of Λ. The definition is restricted to bases without loss of generality, because Lemma 2.1 implies that for any full-rank set $\mathbf{S} \subset \Lambda$, there is a basis $\mathbf{T}$ of Λ such that $\|\tilde{\mathbf{T}}\| \le \|\tilde{\mathbf{S}}\| \le \|\mathbf{S}\|$.

In this section we prove two lemmas regarding the Gram-Schmidt minimum:

Lemma 3.1. *For any n-dimensional lattice Λ and real $\epsilon > 0$, we have*

$$\eta_\epsilon(\Lambda) \le \tilde{bl}(\Lambda) \cdot \sqrt{\log(2n(1+1/\epsilon))/\pi}.$$

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

Lemma 3.2. *For any n-dimensional lattice Λ,*

$$\lambda_1(\Lambda) \le \tilde{bl}(\Lambda) \le \lambda_n(\Lambda) \le 2\mu(\Lambda) \le \sqrt{n} \cdot \tilde{bl}(\Lambda).$$

Furthermore, the latter inequality is tight up to some constant factor, i.e., there exists a family of lattices $\{\Lambda_n\}_{n \in \mathbb{N}}$ such that Λ_n is an n-dimensional lattice and $\lambda_n(\Lambda_n) \ge \Omega(\sqrt{n}) \cdot \tilde{bl}(\Lambda_n)$.

In particular, because $\tilde{bl}(\Lambda) \le \lambda_n(\Lambda)$ by Lemma 3.2, the bound from Lemma 3.1 on the smoothing parameter is at least as strong as a prior one relating it to λ_n [MR07, Lemma 3.3]. Moreover, by the last part of Lemma 3.2, the new bound can be up to an $\Omega(\sqrt{n})$ factor tighter.

We note that our definition of the Gram-Schmidt minimum is equivalent to the (unnamed) quantity $\tilde{bl}$ defined by Cai [Cai98], who gave an elementary proof of the fact that

$$1 \le \lambda_1(\Lambda^*) \cdot \tilde{bl}(\Lambda) \le O(n)$$

for any n-dimensional lattice Λ. In addition, an ℓ_2 version of $\tilde{bl}$, called the "shortest diagonal" $\sigma(\Lambda)$, was defined in [MG02, Chapter 7] as the minimum of $(\sum_{i\in[n]}\|\tilde{\mathbf{b}_i}\|^2)^{1/2}$ over all bases $\mathbf{B}$ of Λ. By standard relations between the ℓ_2 and ℓ_∞ norms, we have $\tilde{bl}(\Lambda) \le \sigma(\Lambda) \le \sqrt{n}\cdot\tilde{bl}(\Lambda)$.

We now prove the two lemmas.

Proof of Lemma 3.1. Let $\mathbf{B}$ be a basis of Λ such that $\|\tilde{\mathbf{B}}\| = \tilde{bl}(\Lambda)$. By applying rigid rotations and reflections to the lattice Λ (resulting in corresponding transformations of the dual lattice Λ^*), we may assume without loss of generality that the orthogonal Gram-Schmidt vectors $\tilde{\mathbf{b}_i}$ are parallel to the standard basis vectors $\mathbf{e}_i \in \mathbb{R}^n$ (respectively), i.e.,

$$\tilde{\mathbf{b}_i} = \|\tilde{\mathbf{b}_i}\|\cdot\mathbf{e}_i.$$

The transformation does not affect the value of the smoothing parameter $\eta_\epsilon(\Lambda)$, because it is defined with respect to the Gaussian measure $\rho_{1/s}(\Lambda^*\backslash\{\mathbf{0}\})$, which is invariant under rotations and reflections.

By Lemma 2.6, it suffices to show that $\lambda_1^\infty(\Lambda^*) \ge 1/\tilde{bl}(\Lambda)$. Let $\mathbf{v}\in\Lambda^*$ be an arbitrary nonzero dual lattice vector, and let $c_i = \langle\mathbf{v},\mathbf{b}_i\rangle \in \mathbb{Z}$ for all $i\in[n]$. Let i be the smallest index such that $c_i \ne 0$; such an i exists because $\mathbf{v}\ne\mathbf{0}$ and the $\mathbf{b}_i$ are linearly independent. Then $\mathbf{v}$ is orthogonal to $\mathrm{span}(\mathbf{b}_1,\ldots,\mathbf{b}_{i-1})$, so we have

$$c_i = \langle\mathbf{v},\mathbf{b}_i\rangle = \langle\mathbf{v},\tilde{\mathbf{b}_i}\rangle = \|\tilde{\mathbf{b}_i}\|\cdot\langle\mathbf{v},\mathbf{e}_i\rangle = \|\tilde{\mathbf{b}_i}\|\cdot v_i \in \mathbb{Z}\backslash\{\mathbf{0}\}.$$

Therefore we have

$$\|\mathbf{v}\|_\infty \ge |v_i| \ge 1/\|\tilde{\mathbf{b}_i}\| \ge 1/\tilde{bl}(\Lambda),$$

as desired. □

Proof of Lemma 3.2. The first inequality follows from $\|\tilde{\mathbf{B}}\| \ge \|\tilde{\mathbf{b}_1}\| = \|\mathbf{b}_1\| \ge \lambda_1(\Lambda)$ for any basis $\mathbf{B}$ of Λ. The second inequality follows immediately from the above discussion on converting a full-rank set into a basis. The third inequality is from [MG02, Theorem 7.9]. For the final inequality, observe that for any target point $\mathbf{t}\in\mathbb{R}^n$, the nearest plane algorithm [Bab86] on input $\mathbf{t}$ and any basis $\mathbf{B}$ of Λ outputs a $\mathbf{v}\in\Lambda$ such that

$$\|\mathbf{v}-\mathbf{t}\|^2 \le \frac{1}{4}\sum_{i\in[n]}\|\tilde{\mathbf{b}_i}\|^2 \le \frac{n}{4}\cdot\|\tilde{\mathbf{B}}\|^2.$$

Letting $\mathbf{B}$ be such that $\|\tilde{\mathbf{B}}\| = \tilde{bl}(\Lambda)$, we see that the covering radius $\mu(\Lambda) \le \frac{\sqrt{n}}{2}\cdot\tilde{bl}(\Lambda)$, as desired.

TODO: write the tightness proof. □

4 Sampling from Discrete Gaussians

Here we show how to use an arbitrary basis $\mathbf{B}$ to sample efficiently from the discrete Gaussian distribution $D_{\Lambda,s,\mathbf{c}}$, for any s greater than $\|\tilde{\mathbf{B}}\|$ (times a small extra factor). In particular, it suffices to have an appropriately short full-rank set of lattice vectors $\mathbf{S}\subset\Lambda$, because by Lemma 2.1 we can efficiently convert it into a basis $\mathbf{B}$ such that $\|\tilde{\mathbf{B}}\| \le \|\tilde{\mathbf{S}}\| \le \|\mathbf{S}\|$.

As a first attempt, consider an algorithm that first samples from a *continuous* Gaussian with parameter s, and then uses $\mathbf{B}$ to "round off" the sampled point to a relatively nearby lattice point. In fact, Regev applied this exact strategy in the "bootstrapping" step of his reduction [Reg05], using an LLL-reduced basis and a Gaussian parameter s that was an *exponential* factor larger than the basis length $\|\mathbf{B}\|$.

Unfortunately, this strategy does not work so well when s is a *small* multiple of the basis length. The problem can be seen even when Λ is a one-dimensional lattice, e.g., the set of integers $\mathbb{Z}\subset\mathbb{R}^1$. Consider the

distribution D induced by the rounding scheme, using a continuous Gaussian centered at zero with parameter $s = n^c$ for some constant $c > 0$. A routine calculation shows that the probability assigned to zero by D is $\mathrm{erf}(\sqrt{\pi}/2s) = 1/s - \Omega(1/s^3)$, by the Maclaurin series expansion of the error function erf. On the other hand, it can be shown that the probability assigned to zero by the desired distribution $D_{\mathbb{Z},s}$ is negligibly close to $1/s$. Therefore, the statistical distance between the two distributions is at least $\Omega(1/s^3)$, which is non-negligible. For n-dimensional lattices, the distribution induced by the rounding scheme is even more biased, because the distance between the sampled point and its rounded-off lattice point grows with n.[6]

Instead of using continuous distributions, we show how to sample "directly" from a lattice under the desired discrete Gaussian distribution. Even in the one-dimensional case, this requires some care: the support of the distribution is infinite, and even a close approximation to it may not have a succinct representation (e.g., when the parameter s is large). We first define a core subroutine that samples from the integer lattice $\mathbb{Z}$. We then use this subroutine to define a randomized variant of Babai's nearest plane algorithm [Bab86], which is essentially equivalent to one proposed by Klein [Kle00] in another context. The novelty here is a *nearly-exact analysis* of its output distribution for a suitable choice of parameters.

Theorem 4.1. *There is a probabilistic polynomial-time algorithm that, given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, a parameter $s \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$, and a center $\mathbf{c} \in \mathbb{R}^n$, outputs a sample from a distribution that is statistically close to $D_{\Lambda,s,\mathbf{c}}$.*

4.1 Sampling Integers

We first define the subroutine $\mathsf{Sample}\mathbb{Z}$, which samples from the discrete Gaussian $D_{\mathbb{Z},s,c}$ over the one-dimensional integer lattice $\mathbb{Z}$. Let $t(n) \geq \omega(\sqrt{\log n})$ be some fixed function, say, $t(n) = \log n$. $\mathsf{Sample}\mathbb{Z}$ uses rejection sampling, and works as follows: on input (s, c) and (implicitly) the security parameter n, choose an integer $x \leftarrow Z \doteq \mathbb{Z} \cap [c - s \cdot t(n), c + s \cdot t(n)]$ uniformly at random. Then with probability $\rho_s(x - c) \in (0, 1]$, output x, otherwise repeat.

The correctness of the $\mathsf{Sample}\mathbb{Z}$ relies on the following tail inequality on the distribution $D_{\mathbb{Z},s,c}$.

Lemma 4.2. *For any $\epsilon > 0$, any $s \geq \eta_\epsilon(\mathbb{Z})$, and any $t > 0$,*

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

In particular, 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.

Proof. Let $\mathcal{B} = (-1, 1) \subset R$ be the one-dimensional open unit ball. We use the following fact from [Ban95, Lemma 2.10]:

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

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 2.7 for the last inequality. This completes the proof. □

[6] By carefully employing an additional rejection sampling step, it is possible to compensate somewhat for the bias in the rounding scheme. However, the resulting algorithm is quite inefficient and "loose," i.e., it works for a parameter s that is a $\sqrt{n}$ factor larger than the *diameter* the basis (which may itself be up to an n factor larger than length of the basis).

Lemma 4.3. *For any $0 < \epsilon < \exp(-\pi)$, any $s \geq \eta_\epsilon(\mathbb{Z})$ and $c \in \mathbb{R}$, and any $\omega(\log n)$ function,* SampleZ *terminates within $t(n) \cdot \omega(\log n)$ iterations with overwhelming probability, and its output distribution is statistically close to $D_{\mathbb{Z},s,c}$. (Note that the number of iterations is independent of the Gaussian parameter s.)*

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. Then the output distribution of SampleZ is identical to D. Furthermore, D and $D_{\mathbb{Z},s,c}$ are statistically close, by Lemma 4.2.

We now analyze the running time. Each iteration of SampleZ picks an integer x uniformly at random from Z. The probability that x lies in $\mathbb{Z} \cap [c-s, c+s]$ is at least $(2s-1)/(2st+1) \geq 1/(3t)$, 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 algorithm therefore terminates within $t(n) \cdot \omega(\log n)$ iterations with overwhelming probability. (The probability can be made 1 without significantly altering the output distribution by terminating and outputting 0 after some $\omega(\log n)$ iterations.) □

4.2 Sampling from Arbitrary Lattices

We now describe a randomized nearest-plane algorithm, called SampleD, that samples from a discrete Gaussian $D_{\Lambda,s,\mathbf{c}}$ over *any* lattice Λ. In each iteration, the algorithm simply chooses a plane at random by sampling from an appropriate discrete Gaussian over the integers $\mathbb{Z}$.

The input to SampleD is an (ordered) basis $\mathbf{B}$ of an n-dimensional lattice Λ, a parameter $s > 0$, and a center $\mathbf{c} \in \mathbb{R}^n$. We describe the algorithm as if it has access to an oracle that samples exactly from $D_{\mathbb{Z},s',c'}$ for any desired $s' > 0$ and $c' \in \mathbb{R}$. (As long as s' is sufficiently large, the oracle can be implemented by the SampleZ algorithm described above.) SampleD proceeds as follows:

1. Let $\mathbf{v}_n \leftarrow \mathbf{0}$ and $\mathbf{c}_n \leftarrow \mathbf{c}$. For $i \leftarrow n, \ldots, 1$, do:
   (a) Let $c'_i = \langle \mathbf{c}_i, \tilde{\mathbf{b}_i} \rangle / \langle \tilde{\mathbf{b}_i}, \tilde{\mathbf{b}_i} \rangle \in \mathbb{R}$ and $s'_i = s / \|\tilde{\mathbf{b}_i}\| > 0$.
   (b) Choose $z_i \sim D_{\mathbb{Z},s'_i,c'_i}$ (this is the only step that differs from the nearest-plane algorithm).
   (c) Let $\mathbf{c}_{i-1} \leftarrow \mathbf{c}_i - z_i \mathbf{b}_i$ and let $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$.
2. Output $\mathbf{v}_0$.

Assuming scalar operations take unit time, the running time of the algorithm is $O(n^2)$ plus the running time of the n oracle calls. Note that every variable is assigned exactly once, and the value $\mathbf{c}_i$ (respectively, $\mathbf{v}_i$, c'_i, s'_i) is never used once $\mathbf{c}_{i-1}$ (resp., $\mathbf{v}_{i=1}$, c'_{i-1}, s'_{i-1}) is defined. Therefore, an implementation would typically use one mutable register to store the successive values of $\mathbf{c}_i$ (likewise, $\mathbf{v}_i, c'_i, s'_i$); the indices are only in place to aid the analysis.

By construction, the output of SampleD is always a lattice vector, and there is a bijective correspondence between the random choices of the z_is and the lattice. In the following, for any fixed lattice vector $\mathbf{v} = \sum_{i \in [n]} \hat{z}_i \mathbf{b}_i \in \Lambda$ (where the input $(\mathbf{B}, s, \mathbf{c})$ is implicit), let SampleD $\rightarrow \mathbf{v}$ denote the collection of values assinged to all the internal variables during a hypothetical execution of SampleD that outputs $\mathbf{v}$, i.e., where every choice of $z_i = \hat{z}_i$.

Lemma 4.4. *For any input $(\mathbf{B}, s, \mathbf{c})$ and any output $\mathbf{v} = \mathbf{v}_0 = \sum_{i \in [n]} \hat{z}_i \mathbf{b}_i \in \mathcal{L}(\mathbf{B})$ of* SampleD*,*

$$\mathbf{v} - \mathbf{c} = \sum_{i \in [n]} (\hat{z}_i - c'_i) \cdot \tilde{\mathbf{b}_i},$$

where the values c'_i are as in SampleD $\rightarrow \mathbf{v}$.

Proof. For $i \in [n]$, define $\pi_i : \mathbb{R}^n \to \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ to be the orthogonal projection onto its range. We prove by induction that for all $j = 0, \ldots, n$,

$$(\mathbf{v}_0 - \mathbf{v}_j) - \pi_j(\mathbf{c}_j) = \sum_{i \in [j]} (\hat{z}_i - c'_i) \cdot \tilde{\mathbf{b}}_i,$$

where the relevant variables above are as in $\mathsf{SampleD} \to \mathbf{v}$. The claim follows by taking $j = n$ and the fact that $\mathbf{v}_n = \mathbf{0}$, $\mathbf{c}_n = \mathbf{c}$, and $\mathbf{c} \in \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \mathbb{R}^n$.

First, observe that the base case $j = 0$ is trivially true. Now suppose the hypothesis is true for $j = k - 1$ for some $k \in [n]$. Using $\mathbf{v}_k = \mathbf{v}_{k-1} - \hat{z}_k \mathbf{b}_k$, $\mathbf{c}_k = \mathbf{c}_{k-1} + \hat{z}_k \mathbf{b}_k$, the definition of c'_k, and the inductive hypothesis, we have

$$\begin{aligned}
\mathbf{v}_0 - \mathbf{v}_k - \pi_k(\mathbf{c}_k) &= (\mathbf{v}_0 - \mathbf{v}_{k-1}) + \hat{z}_k \mathbf{b}_k - (\pi_{k-1}(\mathbf{c}_k) + c'_k \tilde{\mathbf{b}_k}) \\
&= (\mathbf{v}_0 - \mathbf{v}_{k-1}) + \hat{z}_k \mathbf{b}_k - \pi_{k-1}(\mathbf{c}_{k-1}) - \pi_{k-1}(\hat{z}_k \mathbf{b}_k) - c'_k \tilde{\mathbf{b}_k} \\
&= (\mathbf{v}_0 - \mathbf{v}_{k-1} - \pi_{k-1}(\mathbf{c}_{k-1})) + \hat{z}_k (\mathbf{b}_k - \pi_{k-1}(\mathbf{b}_k)) - c'_k \tilde{\mathbf{b}_k} \\
&= (\mathbf{v}_0 - \mathbf{v}_{k-1} - \pi_{k-1}(\mathbf{c}_{k-1})) + (\hat{z}_k - c'_k) \cdot \tilde{\mathbf{b}_k} \\
&= \sum_{i \in [k]} (\hat{z}_i - c'_i) \cdot \tilde{\mathbf{b}}_i,
\end{aligned}$$

which proves the claim for $j = k$, and we are done. □

Lemma 4.5. *For any input* $(\mathbf{B}, s, \mathbf{c})$ *and any* $\mathbf{v} = \sum_{i \in [n]} \hat{z}_i \mathbf{b}_i \in \mathcal{L}(\mathbf{B})$*, the probability that* $\mathsf{SampleD}$ *outputs* $\mathbf{v}$ *is exactly*

$$\rho_{s,\mathbf{c}}(\mathbf{v}) \cdot \prod_{i \in [n]} \frac{1}{\rho_{s'_i, c'_i}(\mathbb{Z})},$$

where the values s'_i, c'_i *are as in* $\mathsf{SampleD} \to \mathbf{v}$.

Proof. Consider the event E that $\mathsf{SampleD}$ outputs $\mathbf{v}$. First, observe that E occurs if and only if every random choice $z_i = \hat{z}_i$ for $i = n, \ldots, 1$. For each i, the probability that $z_i = \hat{z}_i$, conditioned on $z_j = \hat{z}_j$ for all $j = n, \ldots, i + 1$, is exactly $D_{\mathbb{Z}, s'_i, c'_i}(\hat{z}_i)$. Therefore, the probability of E is

$$\prod_{i \in [n]} D_{\mathbb{Z}, s'_i, c'_i}(\hat{z}_i) = \frac{\prod_{i \in [n]} \rho_{s'_i, c'_i}(\hat{z}_i)}{\prod_{i \in [n]} \rho_{s'_i, c'_i}(\mathbb{Z})}.$$

The numerator in the above expression is

$$\prod_{i \in [n]} \rho_{s'_i, c'_i}(\hat{z}_i) = \prod_{i \in [n]} \rho_s((\hat{z}_i - c'_i) \cdot \|\tilde{\mathbf{b}_i}\|) = \rho_s\left(\sum_{i \in [n]} (\hat{z}_i - c'_i) \cdot \tilde{\mathbf{b}_i}\right) = \rho_s(\mathbf{v} - \mathbf{c}) = \rho_{s,\mathbf{c}}(\mathbf{v}),$$

where the first equality is by definition of s'_i and $\rho_{s'_i, c'_i}$, the second equality is by mutual orthogonality of the Gram-Schmidt vectors $\tilde{\mathbf{b}_i}$ and the definition of ρ_s, and the third equality is by Lemma 4.4. This completes the proof. □

We now prove the main theorem.

Proof of Theorem 4.1. The algorithm is simply SampleD using SampleZ to implement the oracle for $D_{\mathbb{Z},s',c'}$. By Lemma 3.1, there is a negligible $\epsilon(n)$ for which each $s'_i = s/\|\tilde{\mathbf{b}}_i\| \geq \omega(\sqrt{\log n}) \geq \eta_\epsilon(\mathbb{Z})$. Then Lemma 4.3 implies that SampleZ faithfully implements the oracle (to within negligible statistical distance).

We now show that SampleD (given a perfect oracle for $D_{\mathbb{Z},s',c'}$) samples to within negligible statistical distance of $D_{\Lambda,s,\mathbf{c}}$. First note that under the desired distribution $D_{\Lambda,s,\mathbf{c}}$, the probability of $\mathbf{v}$ is $Q^{-1} \cdot \rho_{s,\mathbf{c}}(\mathbf{v})$, where $Q = \rho_{s,\mathbf{c}}(\Lambda)$ is a normalization factor. Now consider the output distribution of SampleD. Because each $s'_i \geq \eta_\epsilon(\mathbb{Z})$ where $\epsilon(n)$ is the negligible function from above, Lemma 2.7 implies that

$$\rho_{s'_i,c'_i}(\mathbb{Z}) \in [\tfrac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_{s'_i}(\mathbb{Z})$$

for any value of $c'_i \in \mathbb{R}$. By Lemma 4.5, for every $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ the probability that SampleD outputs $\mathbf{v}$ is in the range

$$R^{-1} \cdot \left[1, (\tfrac{1+\epsilon}{1-\epsilon})^n\right] \cdot \rho_{s,\mathbf{c}}(\mathbf{v}) \subseteq R^{-1} \cdot [1, 1+\epsilon'] \cdot \rho_{s,\mathbf{c}}(\mathbf{v}),$$

where $R = \prod_{i\in[n]} \rho_{s'_i}(\mathbb{Z})$ is a normalization factor independent of $\mathbf{v}$ and $\mathbf{c}$, and $\epsilon'(n)$ is some negligible function. It follows that $R \in [1, 1+\epsilon'] \cdot Q$, and a routine calculation shows that the statistical distance between SampleD's output distribution and $D_{\Lambda,s,\mathbf{c}}$ is at most $\epsilon'/2$. □

5 Trapdoors for Hard Lattices

In this section, we demonstrate trapdoors for certain families of random lattices that, roughly speaking, enjoy worst-case hardness. We then develop some foundational tools and primitives that our cryptographic applications will build upon.

5.1 Hard Random Lattices

We start by giving a unified description of two related families of random lattices that have appeared in recent works. Both families consist of integer lattices (i.e., subsets of $\mathbb{Z}^m$) that are invariant under shifts by q in each of the coordinates, for some specified integer modulus q. In other words, whether a vector $\mathbf{x} \in \mathbb{Z}^m$ belongs to the lattice is determined entirely by the entries of $\mathbf{x}$ modulo q.

In more detail, let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for some positive integers n, m, q. In this work (as in prior ones), 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., $O(n^3)$. We consider two kinds of full-rank m-dimensional integer lattices defined by $\mathbf{A}$. The first consists of those integer vectors that are "orthogonal" (modulo q) to the rows of $\mathbf{A}$, and is defined as

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

The second lattice is generated by the (transposed) rows of $\mathbf{A}$, and is defined as

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

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

Throughout the paper, we use two easy but important facts about the lattices defined above. First, it can be seen from their definitions that Λ and $\Lambda^\perp$ (appropriately scaled) are duals:

$$\Lambda^\perp = q \cdot \Lambda^* \quad \text{and} \quad \Lambda = q \cdot (\Lambda^\perp)^*.$$

The second 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 $(\mathbf{e} + \Lambda^\perp) \mapsto \mathbf{A}\mathbf{e} \bmod q$. In other words, computing the syndrome $\mathbf{A}\mathbf{e} \bmod q$ for some $\mathbf{e} \in \mathbb{Z}^m$ is equivalent to reducing $\mathbf{e}$ modulo the lattice $\Lambda^\perp(\mathbf{A})$.

Ajtai [Ajt96] first showed that for appropriate parameters, solving SVP on the lattice $\Lambda^\perp(\mathbf{A})$ for uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is as hard as approximating certain problems (e.g., SIVP and GapSVP) on *any* lattice of dimension n to within $\text{poly}(n)$ factors. Since then, the approximation factors for the underlying problems have been improved to as small as $\tilde{O}(n)$ [CN97, Mic04, MR07].

Regev [Reg05] defined the *learning with error* (LWE) problem, which can be phrased as a bounded-distance problem on $\Lambda(\mathbf{A})$ where $\mathbf{A}$ is chosen uniformly at random. He showed that LWE is hard on the average unless there is an efficient *quantum* algorithm for solving SIVP and GapSVP on *any* lattice of dimension n to within $\tilde{O}(n)$ factors.

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 5.1. *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 syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ there is a $\mathbf{e} \in \{0,1\}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$.*[7]

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 5.2. *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 $s \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$. Then for $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$, 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 $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$ 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 2.8, for $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$ 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 $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$ 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 $\mathbf{e} = \mathbf{t} + \mathbf{v}$, we see that $\mathbf{v} = \mathbf{e} - \mathbf{t}$ is distributed as $D_{\Lambda^\perp, s, -\mathbf{t}}$, and the claim follows. □

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

[7] 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 $\mathbf{A}$.

Lemma 5.3. *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 for any $\omega(\sqrt{\log m})$ function, there is a negligible function $\epsilon(m)$ such that $\eta_\epsilon(\Lambda^\perp(\mathbf{A})) \leq \omega(\sqrt{\log m})$

Proof. The second part of the claim follows by Lemma 2.6 and the fact that $\Lambda = q \cdot (\Lambda^\perp)^*$.

For the first part of 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 5.4. *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 $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$.*

Proof. By Lemmas 5.1 and 5.3, 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 $s \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ for some negligible function $\epsilon(m)$. Now by Lemma 5.2, the distribution of $\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. □

5.2 Hard Average-Case 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 $\Lambda^\perp(\mathbf{A})$. The problem was formalized as follows in [MR07].

Definition 5.5. 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).

Definition 5.6. 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 $\Lambda^\perp = \Lambda^\perp(\mathbf{A})$. 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

$\mathbf{Av} = \mathbf{At} - \mathbf{Ae} = \mathbf{0} \bmod q$; 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 $\mathbf{Ae} = \mathbf{u} \bmod q$. In this work it will be more convenient and efficient 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 $\Lambda^\perp$ in $\mathbb{Z}_q^n$, rather than an unrestricted target point 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 prime q), Lemma 5.1 implies that with overwhelming probability over the choice of $\mathbf{A}$, there is an $\mathbf{e} \in \{0,1\}^m$ such that $\mathbf{Ae} = \mathbf{u} \bmod q$ for any $\mathbf{u} \in \mathbb{Z}_q^n$. Then because $\|\mathbf{e}\| \leq \sqrt{m} \leq \beta$, we see that a uniformly 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 5.7. *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 SIVP problem (among others) in the worst case to within certain $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$ factors.*[8]

Note that Proposition 5.7 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 basis; the shorter the basis, the smaller we may take β to be, and the weaker the underlying assumptions can be.

The proof of Proposition 5.7 appears in Section 9.

5.3 Preimage Sampleable Functions

5.3.1 Definitions

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

A collection of one-way *preimage sampleable functions* (PSFs) 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 $(a, t) \leftarrow \mathsf{TrapGen}(1^n)$.

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

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.[9]

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

A collection of *collision-resistant* preimage sampleable functions 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.

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' form 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 PSFs 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.

5.3.2 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 relatively short full-rank "trapdoor" set of lattice vectors $\mathbf{S} \subset \Lambda^{\perp}(\mathbf{A})$.

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

Proposition 5.8 ([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 $\mathbf{S} \subset \Lambda^\perp(\mathbf{A})$, where the distribution of $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ and the length $\|\mathbf{S}\| \leq L = m^{2.5}$.*

In particular, by Lemma 2.1, the set $\mathbf{S}$ can be converted efficiently to a "good" basis $\mathbf{T}$ of $\Lambda^\perp(\mathbf{A})$ such that $\|\tilde{\mathbf{T}}\| \leq \|\tilde{\mathbf{S}}\| \leq L$.

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

We can now construct a collection of PSFs based on the average-case hardness of SIS and/or ISIS. Let q, m, and L be as in Proposition 5.8. The collection is parameterized by some Gaussian parameter $s \geq L \cdot \omega(\sqrt{\log m})$.

- The function generator uses the algorithm from Proposition 5.8 to choose $(\mathbf{A}, \mathbf{T})$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is statistically close to uniform and $\mathbf{T} \subset \Lambda^\perp(\mathbf{A})$ is a good basis with $\|\tilde{\mathbf{T}}\| \leq L$. The matrix $\mathbf{A}$ (and q) defines the function $f_{\mathbf{A}}(\cdot)$, and the good basis $\mathbf{T}$ 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 SampleD with the standard basis for $\mathbb{Z}^m$.

- The trapdoor inversion algorithm $\mathsf{SampleISIS}(\mathbf{A}, \mathbf{T}, s, \mathbf{u})$ 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 $\mathbf{A}\mathbf{t} = \mathbf{u} \bmod q$ (such a $\mathbf{t}$ exists for all but an at most q^{-n} fraction of $\mathbf{A}$, by Lemma 5.1). Then sample $\mathbf{v} \sim D_{\Lambda^\perp, s, -\mathbf{t}}$ using $\mathsf{SampleD}(\mathbf{T}, s, -\mathbf{t})$, and output $\mathbf{e} = \mathbf{t} + \mathbf{v}$.

We stress that it is important to sample the input from the *discrete* Gaussian $D_{\mathbb{Z}^m, s}$, rather than (say) sampling from a *continuous* Gaussian over $\mathbb{R}^m$ (with parameter s) and rounding off each coordinate to the nearest integer. The reason is that the inversion algorithm samples a preimage from the *former* distribution (conditioned on a particular output), and the latter distribution differs from the former by non-negligible statistical distance (see the discussion at the beginning of Section 4).

Theorem 5.9. *The algorithms described above give a collection of* one-way *PSFs if $\mathsf{ISIS}_{q,m,s\sqrt{m}}$ is hard. Moreover, they give a collection of* collision-resistant *PSFs if $\mathsf{SIS}_{q,m,2s\sqrt{m}}$ is hard.*

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 3.1, because $L \geq \|\tilde{\mathbf{T}}\|$.

We start with domain sampling. A sample $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$ lands in the domain D_n (except with exponentially small probability), by Lemma 2.9. 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 5.4.

We now show preimage sampling. Because $s \geq \|\tilde{\mathbf{T}}\| \cdot \omega(\sqrt{\log m})$, Theorem 4.1 implies that SampleD samples from a distribution that is statistically close to $D_{\Lambda^\perp, s, -\mathbf{t}}$. Then by the second claim of Lemma 5.2, 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 5.2), and by the min-entropy of the discrete Gaussian (Lemma 2.10).

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 $(\mathbf{A}, \mathbf{T})$ as above, as well as a uniform $\mathbf{w} \in \mathbb{Z}_q^n$. It outputs a pair of functions $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \bmod q$ and $f_{\mathbf{A},\mathbf{w}}(\mathbf{e}) = \mathbf{A}\mathbf{e} + \mathbf{w} \bmod q$. 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 $\mathbf{A}\mathbf{e} = (\mathbf{u} - \mathbf{w}) \bmod q$.

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 4.1) 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.)

6 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 FDH (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 originally intended to be instantiated with trapdoor *permutations*, such as RSA. In this section, we show that they can be instantiated securely using our notion of preimage sampleable functions. 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 trapdoor permutations treated as a black-box, 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.

6.1 Full-Domain Hash Scheme

We start with a version of the FDH signature scheme using trapdoor *collision-resistant* PSFs; 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 (e.g., the random oracle itself) 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 collision-resistant PSFs 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 efficiently-recognizable 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 6.1. *The scheme described above is strongly existentially unforgeable under a chosen-message attack.*

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_m 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 5.3.1 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 sampleability 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. □

6.2 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 6.2. *The scheme described above is strongly existentially unforgeable under a chosen-message attack.*

Proof. The proof is almost identical the prior one, so we simply describe 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 $H(m\|r)$. □

7 Identity-Based Encryption

In this section we construct an identity-based encryption (IBE) system based on the LWE problem, in the random oracle model. (We also show that a similar system is secure in the plain model, under an "interactive" assumption relating to LWE.)

As we show in Section 8, there is a variant of Regev's cryptosystem [Reg05] in which a single lattice $\Lambda(\mathbf{A})$ can support the public keys of many users, and a trapdoor for $\mathbf{A}$ enables extraction of 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 $\Lambda(\mathbf{A})$. 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 Regev's cryptosystem, 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 8.1, 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 5.3. (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 $\mathbb{Z}_q$.

- 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 $\mathbf{u} = f_{\mathbf{A}}(\mathbf{e})$.

- 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 $x \leftarrow \chi$.

- 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.

The above cryptosystem can be extended to encrypt messages of length $k = \text{poly}(n)$ bits, with ciphertexts of $\tilde{O}(m + k)$ bits and public keys of size $\tilde{O}(kn)$ bits. The idea is to include k independent syndromes $\mathbf{u}_1, \ldots, \mathbf{u}_k$ in the public key, and to encrypt to each of them using the same $\mathbf{s}$ and $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$. (This is similar to an amortized construction from [PVW07] for Regev's original system, and to the IBE from [BGH07]). For $k = \Omega(m)$, this yields amortized encryption/decryption time of $\tilde{O}(n)$ bit operations per message bit, and ciphertext expansion factor of $O(\log n)$. It is also possible to securely encrypt $\Omega(\log n)$ bits per syndrome under essentially the same assumption on LWE, which yields a ciphertext expansion factor of $O(1)$. (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 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 DualKeyGen *is statistically close to uniform over $\mathbb{Z}_q^n$.*

Proof. The claim on the distribution of public keys follows directly from Corollary 5.4.

We show that DualDec is correct with overwhelming probability (over the randomness of DualKeyGen and 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 8.2, 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 5.9. 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 $\mathbf{A}$). □

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 preimage sampleability as proved in Theorem 5.9. As with the full-domain hash signature scheme from Section 6, 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 a trapdoor function $f_{\mathbf{A}}$ with trapdoor $\mathbf{T}$, as described in Section 5.3.2. The master public key is $\mathbf{A}$, which is taken as the shared matrix for the dual cryptosystem, and the master secret key is $\mathbf{T}$.
- $\mathsf{IBEExtract}(\mathbf{A}, \mathbf{T}, id)$: 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 $\mathbf{T}$. 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 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 anonymous and 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 6. 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 discrete (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 Trapdoors for Learning with Errors

Here we demonstrate some trapdoor techniques for the learning with errors (LWE) problem [Reg05] and certain cryptosystems based upon it. Some of these techniques have been applied in a concurrent work of Peikert, Vaikuntanathan, and Waters [PVW07] to construct efficient and universally composable oblivious transfer protocols based on LWE.

8.1 Variant Cryptosystem

We start with a slight variant of Regev's cryptosystem [Reg05], which differs only in the encryption algorithm. The original algorithm chooses a uniformly random binary vector $\mathbf{e} \in \{0,1\}^m$ (corresponding to a random subset of $\{1, \ldots, m\}$) and computes the corresponding subset sum $\mathbf{A}\mathbf{e}$ 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 4), and computes the syndrome $\mathbf{A}\mathbf{e}$. The particular value of the Gaussian parameter r will be a parameter of the scheme, and can be chosen based on the needs of the particular application.

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

- 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 $\mathbf{A}^T\mathbf{s} \in \Lambda(\mathbf{A}) \bmod q$, perturbed by some (small) random error vector $\mathbf{x}$.

- 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 $(\mathbf{u}, c) = (\mathbf{A}\mathbf{e}, \mathbf{p}^T\mathbf{e} + b \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^{n+1}$.

- $\mathsf{LWEDec}(\mathbf{s}, (\mathbf{u}, c))$: 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 8.1. *For parameters r, q, m, and α satisfying the hypotheses of Lemmas 8.2 and 8.4, the cryptosystem above is 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 2.11 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, Reg04b, Reg05]. We first show in Lemma 8.2 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 "messy" public keys, whose defining property is that the encryption algorithm *statistically* hides the message bit when encrypting under such keys. Lastly, in Lemma 8.3 we describe an explicit geometric condition that makes $\mathbf{A}'$ messy, and in Lemma 8.4 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 messy. Therefore encrypting under keys generated by $\mathsf{LWEKeyGen}$ hides the encrypted bit *computationally*. □

Lemma 8.2 (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 2.9, 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. □

Message-lossy ("messy") public keys. Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a fixed common matrix and $\mathbf{p}$ be a fixed public key, and let $\mathbf{A}' \in \mathbb{Z}_q^{(n+1)\times m}$ be constructed from $\mathbf{A}$ and $\mathbf{p}$ as above. Define $\delta = \delta(\mathbf{p})$ 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 $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, r}$. Then

$$\Delta(\mathsf{LWEEnc}(\mathbf{p}, 0), \mathsf{LWEEnc}(\mathbf{p}, 1)) \leq 2\delta,$$

because both $\mathsf{LWEEnc}(\mathbf{p}, 0)$ and $\mathsf{LWEEnc}(\mathbf{p}, 1)$ are within $\delta(pk)$ of uniform. When δ is negligibly small, we say that $\mathbf{p}$ is a "messy" (message-lossy) public key. Of course, the correctness of LWEDec implies that the public keys $\mathbf{p}$ generated by $\mathsf{LWEKeyGen}$ typically have large $\delta(\mathbf{p})$.

Messy keys (though not named as such) have played a crucial role in the security proofs for prior lattice-based cryptosystems [AD97, Reg04b, Reg05]. In those works it was sufficient to show that *most* keys are messy, without necessarily identifying their particular characteristics. This was always done via non-constructive probabilistic arguments.

In the proof of security for the oblivious transfer protocol in [PVW07], the simulator needs to *identify messy keys efficiently* (with the help of a trapdoor). This calls for an *explicit* condition that identifies such keys. The following lemma shows that two conditions together imply messiness: first, the columns of $\mathbf{A}'$ should generate all of $\mathbb{Z}_q^{n+1}$; second, the modular lattice $\Lambda^\perp(\mathbf{A}')$ should have small smoothing parameter, which is the case if $\Lambda(\mathbf{A}')$ has large minimum distance λ_1^∞ in the ℓ_∞ norm.

Lemma 8.3 (Messy Sufficient Condition)**.** *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 $r \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}'))$ used by LWEEnc, we have $\delta(\mathbf{p}) \leq 2\epsilon$.*

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

Proof. The first claim is a consequence of Lemma 5.2 (for dimension $n+1$ instead of n), which directly implies that $\delta(\mathbf{p}) \leq 2\epsilon$. The second claim follows directly from Lemma 2.6 and the duality between $\Lambda^\perp$ and Λ. □

Lemma 8.4 (Density of Messy 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 messy 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 Lemma 5.1, the columns of $\mathbf{A}'$ generate $\mathbb{Z}_q^{n+1}$ for all but an at most $q^{-(n+1)} < q^{-n}$ fraction of all $\mathbf{A}'$. Likewise, by Lemma 5.3, we have $\lambda_1^\infty(\Lambda(\mathbf{A}')) \geq q/4$ 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 8.3 implies that $\mathbf{p}$ is a messy key. □

8.2 Identifying Messy Keys Efficiently

Here present an algorithm that identifies (most) messy 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 uniformly random shared matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and an arbitrary public key $\mathbf{p} \in \mathbb{Z}_q^m$ that together form $\mathbf{A}'$ as above. Recall that $\mathbf{p}$ is messy if: (1) the columns of $\mathbf{A}'$ generate $\mathbb{Z}_q^{n+1}$, and (2) the minimum distance $\lambda_1^\infty(\Lambda(\mathbf{A}'))$ in ℓ_∞ norm is large enough. The first condition is true if and only if the rows of $\mathbf{A}'$ are linearly independent over $\mathbb{Z}_q$, which is easy to check. The second condition can be checked using a trapdoor for $\mathbf{A}$ in the following way: first, we know that with high probability the lattice $\Lambda = \Lambda(\mathbf{A})$ has large minimum distance λ_1^∞. Adjoining the vector $\mathbf{p}$ to Λ yields the lattice $\Lambda' = \Lambda(\mathbf{A}')$, 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 $k \in \{1, \ldots, q-1\}$. Using techniques of Aharonov and Regev [AR05] (extended to arbitrary ℓ_p norms by Peikert [Pei07]), it can be efficiently checked that each of the multiples $k \cdot \mathbf{p}$ is far from Λ (in ℓ_∞ norm), using samples from a discrete Gaussian over $\Lambda^* = \Lambda^\perp(\mathbf{A})/q$. The Gaussian sampling algorithm with a short full-rank set of vectors in $\Lambda^\perp(\mathbf{A})$ can be used to produce such samples efficiently.

The actual algorithm and its analysis depend crucially on the relationship between the Gaussian parameter r used in encryption and the Gaussian parameter s of the distribution over Λ^*. The smaller s is, the smaller

we can make r and still correctly identify messy keys. In turn, a smaller value of r means that we can use a larger LWE error parameter α and still maintain correctness of the cryptosystem. Finally, a larger value of α corresponds to tighter approximation factors for worst-case lattice problems under the quantum reduction of [Reg05].

We remark that we only know how to identify messy keys for the *single-bit* cryptosystem that uses one public key vector $\mathbf{p}$,as opposed to the more efficient multi-bit system that uses multiple vectors $\mathbf{p}_i$, constructed in [PVW07]. The reason is in the multi-bit system, the key is messy if the minimum distance remains large after adjoining *all* the $\mathbf{p}_i$s to the common lattice $\Lambda(\mathbf{A})$. Testing for this condition (at least naively) seems to require checking all of the exponentially-many integer 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 Λ and those that are close to it, given access to Gaussian samples over the dual lattice Λ^*. The meanings of "far" and "close" are determined by the Gaussian parameter s.

Proposition 8.5 ([AR05, Pei07])**.** *There is a deterministic polynomial-time oracle machine $\mathcal{V}^{\mathcal{O}}$ that, given a full-rank lattice $\Lambda \subset \mathbb{R}^m$ (specified by an arbitrary basis) and a point $\mathbf{x} \in \mathbb{R}^m$, and given access to an oracle $\mathcal{O}$ 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," except with probability $2^{-\Omega(m)}$.*
- *If $\mathrm{dist}^\infty(\mathbf{x}, \Lambda) \geq \sqrt{\log m}/s$, then $\mathcal{V}$ outputs "far," except with probability $2^{-\Omega(m)}$.*

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

Figure 1 defines an algorithm called IsMessy that identifies messy public keys. It has two essential properties (putting aside some very rare exceptional cases):

- IsMessy outputs "messy" on an overwhelming fraction of all public keys $\mathbf{p}$.
- If IsMessy outputs "messy" on a particular public key $\mathbf{p}$, then $\mathbf{p}$ is indeed messy.

Note that IsMessy has one-sided error; it might output "not sure" on some public keys $\mathbf{p}$ that are actually messy. Nevertheless, *most* messy keys are identified as such, and this is good enough for the oblivious transfer application in [PVW07].

Before analyzing IsMessy, let us define GOOD to be the set of common matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ whose columns generate $\mathbb{Z}_q^n$ (i.e., the rows are linearly independent over $\mathbb{Z}_q$) and $\lambda_1^\infty(\Lambda(\mathbf{A})) \geq q/4$. By Lemmas 5.1 and 5.3, all but an at most $2q^{-n}$ fraction of matrices $\mathbf{A}$ are in GOOD when $m \geq 2(n+1)\lg q$.

The next two lemmas establish the two main properties of the IsMessy algorithm. We first show that for all $\mathbf{A} \in$ GOOD, IsMessy outputs "messy" (with overwhelming probability) on almost all public keys $\mathbf{p}$.

Lemma 8.6. *Let $m \geq 2(n+1)\lg q$, and $\mathbf{A} \in$ GOOD. Then for all but an at most*

$$\left(\frac{5\sqrt{\log m}}{q \cdot s}\right)^m$$

fraction of all $\mathbf{p} \in \mathbb{Z}_q^m$, $\mathsf{IsMessy}^{\mathcal{O}}(\mathbf{A}, \mathbf{p})$ outputs "messy" with probability $1 - 2^{-\Omega(m)}$.

The input to the oracle algorithm $\mathsf{IsMessy}^{\mathcal{O}}$ is a common matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a public key $\mathbf{p} \in \mathbb{Z}_q^m$, as well as access to an oracle $\mathcal{O}$ that samples from $D_{\Lambda^*,s}$ (where $\Lambda = \Lambda(\mathbf{A})$, so $\Lambda^* = \Lambda^\perp(\mathbf{A})/q$) for some parameter s where $1/q \leq s \leq 1$.

1. Construct $\mathbf{A}' \in \mathbb{Z}_q^{(n+1)\times m}$ from $\mathbf{A}$ and $\mathbf{p}$ as before. Check that the columns of $\mathbf{A}'$ generate $\mathbb{Z}_q^{n+1}$, i.e., that that rows of $\mathbf{A}'$ are linearly independent over $\mathbb{Z}_q$. If not, output "not sure."
2. For each $k \in \{1, \ldots, q-1\}$, run $\mathcal{V}^{\mathcal{O}}(\Lambda, k \cdot \mathbf{p} \bmod q)$.

   If every run of $\mathcal{V}$ outputs "far," then output "messy." Otherwise, output "not sure."

Note that the oracle $\mathcal{O}$ is used only when $\mathcal{V}$ is called as a subroutine. Typically, $\mathcal{O}$ would be implemented by the $\mathsf{SampleD}$ algorithm using a good basis $\mathbf{T}$ of $\Lambda^\perp(\mathbf{A})$, for a parameter $s \geq \|\tilde{\mathbf{T}}\|/q \cdot \omega(\sqrt{\log m})$ (note that $1/q \leq s \leq 1$ for non-trivial $\mathbf{T}$). This affects the output probabilities of $\mathsf{IsMessy}$ by only a negligible amount.

Figure 1: The $\mathsf{IsMessy}$ algorithm.

Proof. We proceed by a counting argument. Let $\mathbf{A}'$ be as in the $\mathsf{IsMessy}$ algorithm. Because $\mathbf{A} \in \mathsf{GOOD}$, the number of $\mathbf{p}$ such that the rows of $\mathbf{A}'$ are linearly dependent (over $\mathbb{Z}_q$) is at most $q^n \leq 2^m$. From now on, suppose the rows of $\mathbf{A}'$ are linearly independent.

Let $\Lambda = \Lambda(\mathbf{A})$, and 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 within ℓ_∞ norm less than $\sqrt{\log m}/s$ of Λ. By Proposition 8.5, $\mathsf{IsMessy}$ outputs "messy" with probability $1 - 2^{-\Omega(m)}$ on any $\mathbf{p} \in \mathbb{Z}_q^m$ not contained in

$$B = \bigcup_{k \in [q-1]} (Z \cdot k^{-1} \bmod q).$$

We bound the size of B.

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 B at most

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

We conclude that $\mathsf{IsMessy}$ outputs "messy" with probability $1 - 2^{-\Omega(m)}$ on all but at most

$$2^m + (4\sqrt{\log m}/s)^m \leq (5\sqrt{\log m}/s)^m$$

values of $\mathbf{p} \in \mathbb{Z}_q^m$ for all sufficiently large m. This completes the proof. □

We now show, assuming the Gaussian parameter r used by LWEEnc is large enough, that $\mathsf{IsMessy}$ is indeed correct when it declares a key to be messy.

Lemma 8.7. *Let $r \geq q \cdot s\sqrt{m} \cdot \omega(\sqrt{\log m})$ for some arbitrary $\omega(\sqrt{\log m})$ function. There there is a negligible function $\epsilon(m)$ such that the following holds.*

For any $\mathbf{A} \in \mathsf{GOOD}$ and public key $\mathbf{p}$ such that $\delta(\mathbf{p}) > 2\epsilon$ (under LWEEnc with parameter r), $\mathsf{IsMessy}^{\mathcal{O}}(\mathbf{A}, \mathbf{p})$ outputs "messy" with probability at most $2^{-\Omega(m)}$.

Proof. Let $\mathbf{A}' \in \mathbb{Z}_q^{(n+1)\times m}$ be constructed from $\mathbf{A}$ and $\mathbf{p}$ as above. First, by the fact that $\Lambda^\perp(\mathbf{A}')^* = \Lambda(\mathbf{A}')/q$ and Lemma 2.6, there exists a negligible $\epsilon(m)$ such that

$$r \geq 10s\sqrt{m} \cdot \eta_\epsilon(\Lambda^\perp(\mathbf{A}')) \cdot \lambda_1^\infty(\Lambda(\mathbf{A}')) \tag{8.1}$$

for all sufficiently large m.

Now suppose $\delta(\mathbf{p}) > 2\epsilon$. By the contrapositive of Lemma 8.3, either the rows of $\mathbf{A}'$ are linearly dependent over $\mathbb{Z}_q$ or $r < \eta_\epsilon(\Lambda^\perp(\mathbf{A}'))$. In the former case, IsMessy always outputs "not sure," so suppose the latter case is true. Then by (8.1), we obtain

$$\lambda_1^\infty(\Lambda') < \frac{1}{10s\sqrt{m}}.$$

Now because $\mathbf{A} \in \mathsf{GOOD}$, we have $\lambda_1^\infty(\Lambda(\mathbf{A})) \geq q/4 > \lambda_1^\infty(\Lambda')$. 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 8.5, $\mathcal{V}$ outputs "close" for that choice of k, except with probability $2^{-\Omega(m)}$, and the proof is complete. □

8.3 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 $\Lambda(\mathbf{A})$. Whereas distinguishing messy 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 8.8 ([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 $\Lambda = \Lambda(\mathbf{A})$ with $\Lambda^* = \Lambda^\perp(\mathbf{A})/q$, we can implement the oracle in the above lemma for any $s = L/q \cdot \omega(\sqrt{\log m})$ using the SampleD algorithm with any good basis $\mathbf{T}$ of $\Lambda^\perp(\mathbf{A})$ such that $\|\tilde{\mathbf{T}}\| \leq L$.

Lemma 8.9. *Let $s = L/q \cdot \omega(\sqrt{\log m})$ as above, and let $\chi = \bar{\Psi}_\alpha$ for $\alpha \leq 1/(q \cdot s\sqrt{m} \cdot \omega(\sqrt{\log m}))$. Let $\Lambda = \Lambda(\mathbf{A})$, 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 8.8 one can efficiently extract the secret key $\mathbf{s}$ from the public key $\mathbf{p}$ using a good basis $\mathbf{T}$ of $\Lambda^\perp(\mathbf{A})$ such that $\|\tilde{\mathbf{T}}\| \leq L$.

Proof. By Lemma 5.3, 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 8.9 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 $\mathbf{A}$, $\mathbf{s}$, $\mathbf{x}$.

9 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 9.1 (Incremental Independent Vectors Decoding)**.** An input to $\mathsf{IncIVD}^\phi_{\gamma,g}$ 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 9.2. *For any $g(n) > 1$ and negligible $\epsilon(n)$, there is a probabilistic poly-time reduction from solving $\mathsf{IncIVD}_{\gamma,g}^{\eta_\epsilon}$ 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 $\mathsf{IncIVD}_{\gamma,g}^{\eta_\epsilon}$ 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 9.2. 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 $\mathsf{IncIVD}_{\gamma,g}^{\eta_\epsilon}$ 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 $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{Z}_q^{n \times m}$.

   Finally, using the procedure from Lemma 2.1, convert $(\mathbf{B}, \mathbf{S})$ into a basis $\mathbf{T}$ of $\mathcal{L}(\mathbf{B})$ such that $\|\tilde{\mathbf{T}}\| \leq \|\tilde{\mathbf{S}}\| \leq \|\mathbf{S}\|$.

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 $\mathsf{SampleD}(\mathbf{T}, s, \mathbf{c}_i)$. 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 9.3. *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 $s = \|\mathbf{S}\| \cdot q/\gamma \geq \|\tilde{\mathbf{T}}\| \cdot \omega(\sqrt{\log n})$, so by Theorem 4.1, the distribution sampled by 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 2.8, $\mathbf{y}_i \bmod q\mathcal{L}(\mathbf{B})$ 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 $m = \text{poly}(n)$, 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 $1/(2\beta m) = 1/\text{poly}(n)$. The claim follows. □

Claim 9.4. *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 9.5. *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 9.2. □

10 Acknowledgments

We thank Vadim Lyubashevsky for reviewing an earlier draft of this work and directing us to Klein's paper [Kle00], Daniele Micciancio, Phong Nguyen, and Oded Regev for helpful comments and advice, and Brent Waters for useful observations about our IBE system. We also thank the STOC committee and reviewers for many constructive comments, and one anonymous reviewer in particular for suggesting a closer look at the role of the Gram-Schmidt vectors in the sampling procedure.

References

[ABC+05] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *CRYPTO*, pages 205–222, 2005.

[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.

[Ber08] Daniel J. Bernstein. Proving tight security for Rabin/Williams signatures. In *EUROCRYPT*, 2008.

[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*, pages 647–657, 2007. Full version at `http://eprint.iacr.org/2007/177`.

[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 CCS*, 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.

[Cai98] Jin-Yi Cai. A relation of primal-dual lattices and the complexity of shortest lattice vector problem. *Theor. Comput. Sci.*, 207(1):105–116, 1998.

[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.

[CS00] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.

[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.

[GHR99] Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT*, pages 123–139, 1999.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[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.

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

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In *ACM Conference on Computer and Communications Security*, pages 155–164, 2003.

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László 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 *APPROX-RANDOM*, pages 450–461, 2006.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP (2)*, pages 144–155, 2006. Full version in ECCC Report TR05-142.

[LM08] Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In *TCC*, pages 37–54, 2008.

[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.

[Mic07] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, December 2007. Preliminary version in FOCS 2002.

[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.

[NV08] Phong Q. Nguyen and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2008. To appear.

[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 Report TR05-158.

[PR07] Chris Peikert and Alon Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *STOC*, pages 478–487, 2007. Full version in ECCC Report TR06-147.

[PVW07] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. Cryptology ePrint Archive, Report 2007/348, 2007. Available at `http://eprint.iacr.org/2007/348`.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, 2008.

[Reg04a] Oded Regev. Lecture notes on lattices in computer science, 2004. Available at `http://www.cs.tau.ac.il/~odedr/teaching/lattices_fall_2004/index.html`, last accessed 28 Feb 2008.

[Reg04b] 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.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[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.