Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem

Chris Peikert[*]

March 19, 2009

Abstract

We construct public-key cryptosystems that are secure assuming the *worst-case* hardness of approximating the minimum distance on n-dimensional lattices to within small $\mathrm{poly}(n)$ factors. Prior cryptosystems with worst-case connections were based either on the shortest vector problem for a *special class* of lattices (Ajtai and Dwork, STOC 1997; Regev, J. ACM 2004), or on the conjectured hardness of lattice problems for *quantum* algorithms (Regev, STOC 2005).

Our main technical innovation is a reduction from variants of the shortest vector problem to corresponding versions of the "learning with errors" (LWE) problem; previously, only a *quantum* reduction of this kind was known. As an additional contribution, we construct a natural *chosen ciphertext-secure* cryptosystem having a much simpler description and tighter underlying worst-case approximation factor than prior schemes.

Keywords: Lattice-based cryptography, learning with errors, quantum computation

[*] Computer Science Lab, SRI International, Menlo Park, CA, `cpeikert@alum.mit.edu`. 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.

1 Introduction

The seminal work of Ajtai in 1996 revealed the intriguing possibility of basing cryptography on *worst-case* complexity assumptions related to *lattices* [Ajt04]. (An n-dimensional lattice is a discrete additive subgroup of $\mathbb{R}^n$.) Since then, basic cryptographic primitives such as one-way functions and collision-resistant hash functions, along with other notions from "Minicrypt" [Imp95], have been based on the conjectured hardness of important and well-studied lattice problems. Perhaps the most well-known of these, the *shortest vector problem* GapSVP, is to approximate the minimum distance of a lattice, i.e., the length of its shortest nonzero vector. Another, called the *short independent vectors problem* SIVP, is (informally) to find a full-rank set of lattice vectors that are relatively short.

For *public-key encryption* (and related strong notions from "Cryptomania"), however, the underlying worst-case lattice assumptions are somewhat more subtle. The ground-breaking cryptosystem of Ajtai and Dwork [AD97] and subsequent improvements [Reg04b, AD07] are based on a special case of the shortest vector problem, called "unique-SVP," in which the shortest nonzero vector of the input lattice must be significantly shorter than all other lattice vectors that are not parallel to it. Compared to other standard problems, the complexity of unique-SVP is not as well-understood, and there is theoretical and experimental evidence [Cai98, GN08] that it may not be as hard as problems on *general* lattices (for matching approximation factors), due to the extra geometric structure.

A different class of cryptosystems (and the only other known to enjoy worst-case hardness) stem from a work of Regev [Reg05], who defined a very natural intermediate problem called *learning with errors* (LWE). The LWE problem is a generalization of the well-known "learning parity with noise" problem to larger moduli. It is parameterized by a dimension n, a modulus q, and an error distribution χ over $\mathbb{Z}_q$; typically, one considers a Gaussian-like distribution χ that is relatively concentrated around 0. In the *search* version of LWE, the goal is to solve for an unknown vector $\mathbf{s} \in \mathbb{Z}_q^n$ (chosen uniformly at random, say), given any desired $m = \mathrm{poly}(n)$ independent "noisy random inner products"

$$(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q, \qquad i = 1, \ldots, m,$$

where each $\mathbf{a}_i \in \mathbb{Z}_q^n$ is uniformly random and each x_i is drawn from the error distribution χ. In the *decision* version, the goal is merely to distinguish between noisy inner products as above and *uniformly random* samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. It turns out that when the modulus q is *prime* and bounded by $\mathrm{poly}(n)$, the search and decision variants are *equivalent* via an elementary reduction [Reg05]. (As we shall see later on, the hypotheses on q can be relaxed somewhat).

The LWE problem is amazingly versatile. In addition to its first application in a public-key cryptosystem [Reg05], it has provided the foundation for many cryptographic schemes, including chosen ciphertext-secure cryptosystems [PW08], identity-based encryption [GPV08], and others [PVW08, AGV09, CPS09], as well as for hardness of learning results relating to halfspaces [KS06]. We emphasize that all of the above cryptographic applications are based on the presumed hardness of *decision*-LWE.

The main result of [Reg05] is a remarkable connection between lattices and the learning with errors problem, namely: the search version of LWE is at least as hard as *quantumly* approximating GapSVP and SIVP on n-dimensional lattices, in the worst case. (The exact approximation factor is $\tilde{O}(n/\alpha)$, where the error distribution has standard deviation $\approx \alpha \cdot q$ for parameter $\alpha = \alpha(n) \in (0, 1)$.) In other words, there is a polynomial-time quantum algorithm (a reduction) that solves standard lattice problems, given access to an oracle that solves search-LWE. This is an intriguing and nontrivial connection, because despite significant research efforts, efficient quantum algorithms for the lattice problems in question have yet to be discovered. Under the plausible conjecture that no such algorithms exist, it follows that LWE is hard and all of the above cryptographic constructions are secure (even against quantum adversaries).

Due to the relative novelty of quantum computing, however, it may still be premature to place a great deal of confidence in such conjectures, and in any case, it is worthwhile to base hardness results and cryptographic schemes on the weakest possible assumptions. The central question left open in [Reg05] is whether there is a *classical* reduction from lattice problems to LWE. More generally, basing a public-key cryptosystem on any "conventional" worst-case lattice assumption has remained an elusive open question.

1.1 Results

Our main result is the first public-key cryptosystem whose security is based on the conjectured worst-case hardness of approximating the shortest vector problem GapSVP on arbitrary lattices. The core technical innovation is a *classical* reduction from certain lattice problems to corresponding versions of the learning with errors problem. In more detail:

- We show that for *large* moduli $q \geq 2^{n/2}$, the search version of LWE is at least as hard as approximating GapSVP in the worst case, via a classical (probabilistic polynomial-time) reduction. As in [Reg05], the approximation factor for GapSVP is $\tilde{O}(n/\alpha)$, where (roughly speaking) $\alpha \cdot q$ is the standard deviation of the Gaussian error distribution over $\mathbb{Z}_q$.

  More generally, our reduction actually implies that for moduli *as small as* (say) $q \geq n/\alpha$, search-LWE is at least as hard as classically approximating a *novel variant* of the shortest vector problem on general lattices (in the worst case, to within $\tilde{O}(n/\alpha)$ factors). The new problem, which we call ζ*-to-*γ-GapSVP, is (informally) to approximate the minimum distance to within a γ factor, given a promise that it lies within a range having gap $\zeta > \gamma$. This problem is *equivalent* to standard GapSVP for $\zeta \geq 2^{n/2}$; for smaller ζ it is *no harder* than GapSVP, yet even for $\zeta = \mathrm{poly}(n)$ it still appears to be exponentially hard in the dimension n, given the state of the art in lattice algorithms. In our reduction, the modulus q depends linearly on ζ, so relying on an easier variant of GapSVP allows a smaller choice of q.

- We then consider prior LWE-based schemes, such as public-key cryptosystems [Reg05, PVW08] and identity-based encryption [GPV08], in the context of the above classical hardness results. Generally speaking, the security of these schemes is based on the hardness of *decision*-LWE, which (as mentioned above) is equivalent to the *search* version for prime modulus $q = \mathrm{poly}(n)$. While this suffices for basing security on the ζ-to-γ-GapSVP problem for $\zeta = \mathrm{poly}(n)$, it is not enough to give a connection to standard GapSVP, due to the large modulus $q \geq 2^{n/2}$ needed by our reduction.

  Fortunately, an argument communicated by Regev [Reg08] implies that search and decision are also equivalent when q is the product of many small primes. By adapting prior cryptosystems to different sizes of q, we obtain semantically secure cryptosystems based on the worst-case hardness of GapSVP and its ζ-to-γ variant.[1] See Section 1.2.1 for a detailed summary and comparison to prior works.

- As an additional contribution, we construct a very natural LWE-based cryptosystem that is secure under the strong notion of an *adaptive chosen-ciphertext attack*. This provides an alternative to a recent construction of Peikert and Waters [PW08], with the advantages of a much simpler description and analysis, and tighter underlying approximation factors (which are only slightly worse than those of the semantically secure schemes; see Figure 1).

[1] A preliminary version of this work [Pei08b] constructed a different style of cryptosystem based directly on the *search* version of LWE, which gave a connection to standard GapSVP without needing a search/decision equivalence for large q. However, systems based on decision-LWE seem more natural and flexible; see Section 1.2.1 for further discussion.

Cryptosystem	Public key	Expansion	Worst-case problem	Approximation
LWE, $q = 2^{O(n)}$	$O(n^4)$	$O(\log n)$	GapSVP	$\tilde{O}(n^2)$
Ajtai-Dwork [AD97, AD07]	$O(n^4)$	$O(n)$	unique-SVP*	$\tilde{O}(n^2)$
Regev [Reg04b]	$O(n^4)$	$O(n^2)$	unique-SVP*	$\tilde{O}(n^{1.5})$
LWE, $q = \mathrm{poly}(n)$	$\tilde{O}(n^2)$	$O(\log n)$	$\mathrm{poly}(n)$-to-γ-GapSVP GapSVP/SIVP (quantum)	$\tilde{O}(n^{1.5})$
CCA: $q = \mathrm{poly}(n)$ [PW08]	$n^{2+\delta}$	$O(\log n)$	same as above ↑	$n^{5+\delta}$
new CCA: $q = \mathrm{poly}(n)$	$n^{2+\delta}$	$O(\log n)$	same as above ↑	$\tilde{O}(n^2)$
new CCA: $q = 2^{O(n)}$	$n^{4+\delta}$	$O(\log n)$	GapSVP	$\tilde{O}(n^4)$

Figure 1: Efficiency measures and underlying problems for lattice-based cryptosystems with worst-case connections. "Expansion" is the best known *amortized* ratio of ciphertext length to plaintext length, for many-bit messages (we omit $\log n$ factor improvements that are sometimes possible at the expense of slightly looser approximation factors). The final three rows describe known *chosen ciphertext-secure* cryptosystems, all of which are based on LWE; δ denotes an arbitrary positive constant that varies from entry to entry. *See Section 1.2.1 for discussion of a recently discovered connection between GapSVP and unique-SVP.

Our classical hardness results for LWE are *incomparable* to the quantum connections demonstrated by Regev [Reg05]. The reason is that our reduction solves the *decision* problem GapSVP when q is very large, as well as progressively easier variants of GapSVP for smaller values of q. In contrast, Regev's reduction approximates both the *search* problem SIVP as well as GapSVP for small q, but using the extra power of quantum computation.

1.2 Discussion

1.2.1 Efficiency, Approximation Factors, and Prior Cryptosystems

In adapting prior LWE-based (semantically secure) cryptosystems [Reg05, PVW08, GPV08] to our hardness results, the modulus q is the main parameter governing efficiency, as well as the underlying worst-case problem and approximation factor. The public key size is $O(n^2 \log^2 q)$, and the amortized plaintext-to-ciphertext expansion factor can be made as small as $O(\log n)$. The underlying worst-case approximation factor for GapSVP (or its ζ-to-γ variant) is $\gamma = \tilde{O}(n^{1.5}\sqrt{\log q})$. Figure 1 summarizes the efficiency and underlying problems for LWE-based cryptosystems (for selected interesting values of q) and those based on the unique-SVP problem [AD97, Reg04b, AD07].

Using a core component of our reduction and several other ideas, Lyubashevsky and Micciancio [LM09] recently showed that the γ-unique-SVP and $(\gamma \cdot \sqrt{\frac{n}{\log n}})$-GapSVP problems are actually *equivalent*. This implies that prior cryptosystems based on unique-SVP [AD97, Reg04b, AD07] are also secure under the worst-case hardness of GapSVP, with a $\tilde{\Theta}(\sqrt{n})$ relaxation in the underlying approximation factor. Considering the top three lines of Figure 1, we see that all these systems are therefore nearly identical with respect to key size and security, though LWE-based schemes enjoy the "best of all worlds" with respect to approximation factors and plaintext expansion.

A preliminary version of this work [Pei08b] also included a more technical reduction showing that particular bits of the LWE secret s are "hard-core" (pseudorandom). The purpose was to construct cryptosystems (enjoying both semantic and chosen-ciphertext security) based on the *search* version of LWE, due to the lack of a search/decision equivalence for large q at the time. With such an equivalence now in hand (for q of a certain form), it is more convenient and natural to base cryptosystems on decision-LWE, and we consider the other cryptosystems obsolete.

1.2.2 Open Problems

Our core reduction from lattice problems to LWE is *non-adaptive* (all queries to the LWE oracle can be prepared in advance), and seems to be limited to solving variants of the *decision* version GapSVP of the shortest vector problem. In contrast, the quantum reduction of [Reg05] and prior classical reductions for Minicrypt primitives (e.g., [Ajt04, MR07]) are *iterative*. That is, they work by adaptively using their oracles to find shorter and shorter lattice vectors, which also lets them approximate *search* problems such as SIVP. A key question that remains open is whether a classical, iterative reduction exists for LWE.

It would be very interesting to study the complexity of the new ζ-to-γ variant of GapSVP (and other decision problems), in which a gap of intermediate quality is already promised, and a tighter approximation is desired. For example, are such problems NP-hard for any nontrivial values of ζ? Are there reductions from larger to smaller values of ζ, possibly by trading off against γ? In the other direction, are there algorithms that perform better as ζ decreases toward γ?

1.3 Technical Overview

1.3.1 Background

We start by giving a brief, high-level description of the common approach underlying prior cryptosystems [AD97, Reg04b, Reg05, AD07]. These works deal with two types of probability distributions over some domain: the uniform distribution, and distributions that are highly concentrated, or "lumpy," over certain parts of the domain. The two types of distributions are used in the construction and analysis of public-key cryptosystems (the details of which are not relevant at this point).

The heart of each work is a *reduction* from solving some worst-case lattice problem to *distinguishing* between the two types of distributions (uniform and lumpy). In order to guarantee that the reduction produces the prescribed kind of lumpy distributions, it has so far been necessary for the input to obey some kind of *geometric constraint* during some phase of the reduction. For instance, in the work of Ajtai and Dwork and its improvements [AD97, Reg04b, AD07], the input is a lattice that must have a "unique" shortest vector.

Regev's quantum reduction for LWE [Reg05] is more subtle, and because we will be relying on one of its components, we describe it in more detail. The reduction has two parts that alternately feed back to each other. The first is entirely *classical* (non-quantum), and has the following form: given access to an LWE oracle *and* many lattice points drawn from a certain distribution, it solves a *bounded-distance decoding* (BDD) problem to within a certain radius. The goal of the BDD problem is to find the unique lattice vector that is closest to a given target point, under the promise that the target is within some small fraction of the lattice's minimum distance. (This promise is the geometric constraint imposed by the reduction.) The second component of the reduction is *quantum*, and uses an oracle for the BDD problem to generate lattice points according to a more concentrated distribution. These are then fed back to the first component of the reduction to solve BDD for a larger decoding radius, and so on.

The main obstacle to obtaining a purely classical reduction seems to be in making use of an oracle for the BDD problem. Quoting [Reg05], "…it seems to us that the only way to generate inputs to the oracle is the following: somehow choose a lattice point $\mathbf{y}$ and let $\mathbf{x} = \mathbf{y} + \mathbf{z}$ for some perturbation vector $\mathbf{z}$ of length at most d [a small fraction of the minimum distance]. Clearly, on input $\mathbf{x}$ the oracle outputs $\mathbf{y}$. But this is useless since we already know $\mathbf{y}$!" In contrast, the quantum reduction uses the BDD oracle to *uncompute* $\mathbf{y}$ from $\mathbf{x}$, which turns out to be very powerful.

1.3.2 Our Approach

Briefly stated, we find a way to use the BDD oracle to solve a lattice problem, classically. The main idea is to imagine, as a complementary case, a lattice whose minimum distance is *significantly less than* the decoding radius d that the oracle handles. If the reduction generates a point $\mathbf{x} = \mathbf{y} + \mathbf{z}$ as described above, then the original lattice point $\mathbf{y}$ is *statistically hidden* from the oracle. Of course, this process does *not* result in a valid instance of the BDD problem (and the subsequent reduction will not produce valid LWE samples), but this is of no consequence — no matter what the BDD oracle does, it must fail to guess $\mathbf{y}$ with some noticeable probability. On the other hand, when the minimum distance is large enough relative to d, the oracle is obliged to return $\mathbf{y}$ with overwhelming probability. The reduction can therefore distinguish between lattices having small and large minimum distance, thereby solving GapSVP.

We note that this is exactly the main idea behind the Arthur-Merlin protocol for coGapSVP of Goldreich and Goldwasser [GG00]. In effect, our reduction and the BDD oracle play the roles of the verifier and unbounded prover, respectively, in their protocol. To our knowledge, this is the first use of the technique in a worst-case to average-case reduction; prior works solve GapSVP by dealing with the dual lattice. The approach is also closely related to the concept of "lossy (trapdoor) functions" [PW08], which influence our new chosen ciphertext-secure cryptosystems (described below).

The discussion so far has ignored one very important subtlety: the classical reduction from BDD to LWE requires not only an LWE oracle, but also several lattice points drawn from a certain Gaussian-like distribution. In [Reg05], these points are iteratively produced by the quantum component of the reduction, which is unavailable in the classical setting (so we unfortunately lose the iterative nature of the overall reduction). Instead, we use a Gaussian sampling algorithm recently developed in [GPV08]. When run on an LLL-reduced lattice basis [LLL82] (which may always be computed in polynomial time), the quality of the resulting distribution induces a large modulus $q = 2^{O(n)}$ for the LWE problem. For the ζ-to-γ variant of GapSVP, the standard deviation of the Gaussian distribution decreases with ζ, and we can use a correspondingly smaller value of q. (We note that the sampling algorithm from [GPV08] is especially important in this case to get a tight connection between ζ and q.)

1.3.3 Chosen Ciphertext-Secure Cryptosystems

Here we summarize the ideas behind a new, very natural cryptosystem that enjoys CCA-security, i.e., security under active *chosen-ciphertext* attacks. At its heart is a collection of injective trapdoor functions based on LWE. This collection was defined in the recent work of Gentry, Peikert, and Vaikuntanathan [GPV08], and is closely related to an earlier proposal by Goldreich, Goldwasser, and Halevi [GGH97].

The description of a function $g_{\mathbf{A}}$ from the collection is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ made up of m uniformly random and independent vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$, for some large enough m. The function $g_{\mathbf{A}}$ is typically evaluated on a random input, which comes in two parts: a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$, and an error vector $\mathbf{x} \in \mathbb{Z}_q^m$ whose entries x_i are chosen independently from the error distribution χ of the LWE problem. The function is defined simply as

$$\mathbf{b} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \mathbf{A}^t \mathbf{s} + \mathbf{x} \in \mathbb{Z}_q^m.$$

Note that each entry of the output vector is $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i$, so inverting $g_{\mathbf{A}}$ (respectively, distinguishing its output from uniform) is syntactically identical to solving search-LWE (resp., decision-LWE) given m noisy inner products. As shown in [GPV08], the function $g_{\mathbf{A}}$ has a *trapdoor* with which the input $\mathbf{s}$ may be efficiently recovered from $\mathbf{b}$ (when χ is sufficiently concentrated). Concretely, the trapdoor is a "short" basis for a certain lattice defined by $\mathbf{A}$, and the two can be generated together so that $\mathbf{A}$ has the required (nearly) uniform distribution [Ajt99, AP09].

Our CCA-secure scheme relies on the recent "witness-recovering decryption" approach of [PW08], some additional perspectives due to Rosen and Segev [RS09], and a few more techniques that are particular to the use of LWE. The key observation is that k independent functions $g_{\mathbf{A}_1}, g_{\mathbf{A}_2}, \ldots, g_{\mathbf{A}_k}$ remain pseudorandom even when evaluated on the *same* input $\mathbf{s}$ and independent error vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$, because the output simply consists of $k \cdot m$ samples from the LWE distribution. (This fact was also independently observed by Goldwasser and Vaikuntanathan.) For *injective* trapdoor functions, one-wayness under such "correlated inputs" immediately yields chosen-ciphertext security (for short messages), as shown in [RS09]. However, the precise meaning of "injective" turns out to be quite subtle in this context, and our LWE-based trapdoor functions must be carefully modified to satisfy the required conditions. In addition, we show how to use the pseudorandomness of LWE to handle any desired message length.

2 Preliminaries

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

The main security parameter throughout the paper is n, and all other quantities are implicitly functions of n. We use standard $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, and $\omega(\cdot)$ notation to describe the growth of functions, and write $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n) \cdot \log^c n)$ for some fixed constant c. We let $\operatorname{poly}(n)$ denote an unspecified polynomial function $f(n) = O(n^c)$ for some constant c. A function $f(n)$ is ***negligible***, written $\operatorname{negl}(n)$, if $f(n) = o(n^{-c})$ for every constant c. We say that a probability is *overwhelming* if it is $1 - \operatorname{negl}(n)$.

Vector spaces. By convention, all vectors are in column form and are named using bold lower-case letters (e.g., $\mathbf{x}$), and x_i denotes the ith component of $\mathbf{x}$. Matrices are named using bold capital letters (e.g., $\mathbf{X}$), and $\mathbf{x}_i$ denotes the ith column vector of $\mathbf{X}$. We identify a matrix $\mathbf{X}$ with the (ordered) set of its column vectors. For a set $S \subseteq \mathbb{R}^n$, point $\mathbf{x} \in \mathbb{R}^n$, and scalar $c \in \mathbb{R}$, we define $S + \mathbf{x} = \{\mathbf{y} + \mathbf{x} \ : \ \mathbf{y} \in S\}$ and $cS = \{c\mathbf{y} \ : \ \mathbf{y} \in S\}$.

The Euclidean (or ℓ_2) norm on $\mathbb{R}^n$ is $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$. The open unit ball $\mathcal{B}_n \subset \mathbb{R}^n$ (in the ℓ_2 norm) is defined as $\mathcal{B}_n = \{\mathbf{x} \in \mathbb{R}^n \ : \ \|\mathbf{x}\| < 1\}$.

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 as follows: let $\tilde{\mathbf{s}_1} = \mathbf{s}_1$, and for each $i = 2, \ldots, n$, let $\tilde{\mathbf{s}_i}$ be the projection of $\mathbf{s}_i$ onto $\operatorname{span}^\perp(\mathbf{s}_1, \ldots, \mathbf{s}_{i-1})$, i.e., $\tilde{\mathbf{s}_i} = \mathbf{s}_i - \sum_{j=1}^{i-1} \mu_{i,j} \tilde{\mathbf{s}_j}$, where $\mu_{i,j} = \langle \mathbf{s}_i, \tilde{\mathbf{s}_j} \rangle / \langle \tilde{\mathbf{s}_j}, \tilde{\mathbf{s}_j} \rangle$.

Probability. The *statistical distance* between two distributions X and Y over D (or two random variables having those distributions) is defined as $\Delta(X, Y) = \max_{A \subseteq D} |f_X(A) - f_Y(A)|$. Statistical distance is a metric on probability distributions; in particular, it obeys the triangle inequality. Applying a (possibly randomized) function g cannot increase the statistical distance: $\Delta(g(X), g(Y)) \leq \Delta(X, Y)$. The uniform distribution over D is denoted $U(D)$.

For any positive integer n and real $r > 0$, define the n-dimensional Gaussian function $\rho_r^{(n)} : \mathbb{R}^n \to \mathbb{R}$ with parameter r as $\rho_r^{(n)}(\mathbf{x}) = \exp(-\pi(\|\mathbf{x}\|/r)^2)$. (We often omit n when it is clear from context.) The total measure associated to ρ_r is $\int_{\mathbb{R}^n} \rho_r(\mathbf{x})\, d\mathbf{x} = r^n$, so we can define a continuous Gaussian probability distribution over $\mathbb{R}^n$ by its density function $D_r(\mathbf{x}) = \rho_r(\mathbf{x})/r^n$. The Gaussian distribution D_r is spherically symmetric, and its projection onto any unit vector is $D_r^{(1)}$. For $x \in \mathbb{R}$ distributed according to $D_r^{(1)}$ and any $t \geq 1$, a standard tail inequality is that $|x| < r \cdot t$ except with probability at most $\exp(-\pi t^2)$.

It is possible to sample efficiently from D_r to within any desired level of precision. It is also possible to sample efficiently from $U(\mathcal{B}_n)$; see, e.g., [GG00]. For simplicity, we use real numbers in this work and assume that we can sample from D_r^n and $U(\mathcal{B}_n)$ exactly; all the arguments can be made rigorous by using a suitable amount of precision.

We need the following lemma, which says that for two n-dimensional balls whose centers are relatively close, the uniform distributions over the balls have statistical distance bounded away from 1.

Lemma 2.1 ([GG00]). *For any constants $c, d > 0$ and any $\mathbf{z} \in \mathbb{R}^n$ with $\|\mathbf{z}\| \leq d$ and $d' = d \cdot \sqrt{n/(c \log n)}$, we have $\Delta(U(d' \cdot \mathcal{B}_n),\ U(\mathbf{z} + d' \cdot \mathcal{B}_n)) \leq 1 - 1/\operatorname{poly}(n)$.*

2.1 Learning with Errors

Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ be the additive group on the real interval $[0, 1)$ with modulo 1 addition. For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution ϕ on $\mathbb{T}$, define $A_{\mathbf{s},\phi}$ to be the distribution on $\mathbb{Z}_q^n \times \mathbb{T}$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing an error term $e \in \mathbb{T}$ according to ϕ, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle/q + e)$, where the addition is performed in $\mathbb{T}$.

We are primarily concerned with error distributions ϕ over $\mathbb{T}$ that are derived from Gaussians. For $\alpha > 0$, define Ψ_α to be the distribution on $\mathbb{T}$ obtained by taking a sample from the one-dimensional Gaussian D_α and reducing modulo 1.

Definition 2.2. For an integer function $q = q(n)$ and an error distribution ϕ on $\mathbb{T}$, the goal of the *learning with errors* problem $\mathsf{LWE}_{q,\phi}$ in n dimensions is to find $\mathbf{s} \in \mathbb{Z}_q^n$ (with overwhelming probability) given access to any desired $\operatorname{poly}(n)$ number of samples from $A_{\mathbf{s},\phi}$ for some arbitrary $\mathbf{s}$.

2.2 Lattices

An n-dimensional *lattice* is a discrete additive subgroup of $\mathbb{R}^n$. Equivalently, let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$ consist of n linearly independent vectors; the lattice Λ generated by the *basis* $\mathbf{B}$ is

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

(Technically, this is the definition of a *full-rank* lattice, which is all we are concerned with in this work.)

The *minimum distance* $\lambda_1(\Lambda)$ of Λ (in the ℓ_2 norm) is the length of its shortest nonzero vector: $\lambda_1(\Lambda) = \min_{\mathbf{0} \neq \mathbf{x} \in \Lambda} \|\mathbf{x}\|$. It is well-known, and easy to show, that the minimum distance $\lambda_1(\Lambda) \geq \min_i \|\tilde{\mathbf{b}}_i\|$ for any basis $\mathbf{B}$ of Λ.

The *dual lattice* of Λ, denoted Λ^*, is defined as $\Lambda^* = \{\mathbf{x} \in \mathbb{R}^n \ : \ \forall\, \mathbf{v} \in \Lambda, \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}\}$. 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 2.3. *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, $\|\tilde{\mathbf{d}_i}\| = 1/\|\tilde{\mathbf{b}_i}\|$.*

Computational problems.

Definition 2.4 (Shortest Vector Problem)**.** For a function $\gamma(n) \geq 1$, an input to the *shortest vector problem* GapSVP_γ is a pair $(\mathbf{B}, d)$, where $\mathbf{B}$ is a basis of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$ and $d > 0$ is a real number. It is a YES instance if $\lambda_1(\Lambda) \leq d$, and is a NO instance if $\lambda_1(\Lambda) > \gamma(n) \cdot d$.

Note that given an oracle for GapSVP_γ, the minimum distance λ_1 of any lattice can be computed to within a factor of (say) 2γ by binary search on the value d.

We now define a generalization of the shortest vector problem, which is actually the problem that our main worst-case to average-case reduction will be based upon.

Definition 2.5 (ζ-to-γ-GapSVP)**.** For functions $\zeta(n) \geq \gamma(n) \geq 1$, an input to *$\zeta$-to-$\gamma$ shortest vector problem* $\mathsf{GapSVP}_{\zeta,\gamma}$ is a pair $(\mathbf{B}, d)$, where:

- $\mathbf{B}$ is a basis of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$ for which $\lambda_1(\Lambda) \leq \zeta(n)$,
- $\min_i \|\tilde{\mathbf{b}_i}\| \geq 1$, and
- $1 \leq d \leq \zeta(n)/\gamma(n)$.

It is a YES instance if $\lambda_1(\Lambda) \leq d$, and is a NO instance if $\lambda_1(\Lambda) > \gamma(n) \cdot d$.

A few remarks about this definition are in order. First, note that the second condition $\min \|\tilde{\mathbf{b}_i}\| \geq 1$ implies that $\lambda_1(\Lambda) \geq 1$, and is without loss of generality by scaling the basis $\mathbf{B}$. Similarly, the last condition $1 \leq d \leq \zeta(n)/\gamma(n)$ is without loss of generality, because the instance is trivially solvable when d lies outside that range.

The first condition is the interesting one. For any $\zeta(n) \geq 2^{n/2}$, $\mathsf{GapSVP}_{\zeta,\gamma}$ is *equivalent* to the standard GapSVP_γ problem, because an arbitrary basis $\mathbf{B}'$ of Λ can be reduced in polynomial time using the LLL algorithm [LLL82] to another basis $\mathbf{B}$ of Λ so that $\lambda_1(\Lambda) \leq \|\mathbf{b}_1\| \leq 2^{n/2} \cdot \min_i \|\tilde{\mathbf{b}_i}\|$.

For smaller functions $\zeta(n)$, particularly $\zeta(n) = \mathrm{poly}(n)$, the condition is nontrivial and more interesting. The nature of the problem is to approximate the minimum distance to within a gap $\gamma(n)$, given a promise that it lies within a looser range having gap $\zeta(n)$. The promise could be made efficiently verifiable by restricting to "high quality" bases that contain (or guarantee the existence of) a vector of length at most $\zeta(n)$, though this is not necessary and could potentially make the problem easier. To our knowledge, none of the lattice algorithms in the literature (e.g., [AKS01]) are able to solve $\mathsf{GapSVP}_{\zeta,\gamma}$ for $\gamma(n) < \zeta(n) = \mathrm{poly}(n)$ in time better than $2^{\Omega(n)}$, even when the promise is verifiable efficiently, and even when, say, $\zeta(n) = 2\gamma(n)$.

Gaussians on lattices. Micciancio and Regev [MR07] introduced a lattice quantity called the *smoothing parameter*, and related it to the minimum distance of the dual lattice.

Definition 2.6. For an n-dimensional lattice Λ and positive real $\epsilon > 0$, the *smoothing parameter* $\eta_\epsilon(\Lambda)$ is defined to be the smallest r such that $\rho_{1/r}(\Lambda^* \backslash \{\mathbf{0}\}) \leq \epsilon$.

Lemma 2.7 ([MR07, Lemma 3.2])**.** *For any n-dimensional lattice Λ, we have $\eta_{2^{-n}}(\Lambda) \leq \sqrt{n}/\lambda_1(\Lambda^*)$.*

For an n-dimensional lattice Λ and real $r > 0$, define the *discrete Gaussian probability distribution over* Λ with parameter r as $D_{\Lambda,r}(\mathbf{x}) = \rho_r(\mathbf{x})/\rho_r(\Lambda)$ for all $\mathbf{x} \in \Lambda$. Note that the denominator in the above expression is merely a normalization factor. Our reduction uses, as a subroutine, an efficient algorithm that generates samples from discrete Gaussian distributions.

Proposition 2.8 ([GPV08, Theorem 4.1])**.** *There exists a probabilistic polynomial-time algorithm that, given any basis* $\mathbf{B}$ *of an* n*-dimensional lattice* Λ *and any* $r \geq \max_i \|\tilde{\mathbf{b}_i}\| \cdot \omega(\sqrt{\log n})$*, outputs a sample from a distribution that is within* $\operatorname{negl}(n)$ *statistical distance of* $D_{\Lambda,r}$*.*

3 Classical Hardness of LWE

In this section we show that certain forms of the learning with errors problem are at least as hard as classically solving corresponding versions of the shortest vector problem.

3.1 Main Theorem

Theorem 3.1. *Let* $\alpha = \alpha(n) \in (0, 1)$ *be a real number and* $\gamma = \gamma(n) \geq n/(\alpha\sqrt{\log n})$*. Let* $\zeta = \zeta(n) \geq \gamma$ *and* $q = q(n) \geq (\zeta/\sqrt{n}) \cdot \omega(\sqrt{\log n})$*.*

There is a probabilistic polynomial-time reduction from solving $\mathsf{GapSVP}_{\zeta,\gamma}$ *in the worst case (with overwhelming probability) to solving* $\mathsf{LWE}_{q,\Psi_\alpha}$ *using* $\operatorname{poly}(n)$ *samples.*

Note that $\mathsf{GapSVP}_{\zeta,\gamma}$ is potentially hard in the worst case whenever $\zeta > \gamma$, so Theorem 3.1 allows for a choice of q as small as

$$q \geq (\gamma/\sqrt{n}) \cdot \omega(\sqrt{\log n}) = \omega(\sqrt{n}/\alpha).$$

We also mention that using results from [Pei08a], Theorem 3.1 can easily be generalized to work for $\mathsf{GapSVP}_{\zeta,\gamma}$ in any ℓ_p norm, $2 \leq p \leq \infty$, for essentially the same approximation factor γ. We defer the details to the full version.

3.1.1 Regev's Classical Reduction

We rely crucially on the classical component of Regev's quantum reduction, restated here.

Proposition 3.2 ([Reg05, Lemma 3.4])**.** *Let* $\epsilon = \epsilon(n)$ *be a negligible function,* $q = q(n) \geq 2$ *be an integer,* $\alpha = \alpha(n) \in (0, 1)$ *and* $\phi = \Psi_\alpha$*. There is a classical probabilistic polynomial-time reduction* $R^{W,D}(\mathbf{B}, r, \mathbf{x})$ *that, given as input a basis* $\mathbf{B}$ *of an* n*-dimensional lattice* $\Lambda = \mathcal{L}(\mathbf{B})$*, a number* $r \geq \sqrt{2}q \cdot \eta_\epsilon(\Lambda^*)$*, and a target point* $\mathbf{x}$ *within distance* $\alpha q/(\sqrt{2}r)$ *of* Λ*, and given access to*

1. *an oracle* W *that solves* $\mathsf{LWE}_{q,\phi}$ *using* $\operatorname{poly}(n)$ *samples, and*
2. *an oracle* D *that generates samples from* $D_{\Lambda^*,r}$*,*

finds (the unique) $\mathbf{v} \in \Lambda$ *closest to* $\mathbf{x}$ *with overwhelming probability.*

For completeness, we give a brief description of the reduction R described in Proposition 3.2 (however, this is not required to understand the proof of Theorem 3.1 and may be safely skipped). Suppose $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$ is the coefficient vector of $\mathbf{v}$ reduced modulo q. To generate a sample from $A_{\mathbf{s},\phi}$, the reduction uses its oracle D to obtain a sample $\mathbf{y}$ from $D_{\Lambda^*,r}$, lets $\mathbf{a} = (\mathbf{B}^*)^{-1}\mathbf{y} = \mathbf{B}^t\mathbf{y} \bmod q$, and outputs

$$(\mathbf{a}\ ,\ b = \langle \mathbf{y}, \mathbf{x} \rangle/q + e) \in \mathbb{Z}_q^n \times \mathbb{T},$$

where $e \in \mathbb{R}$ is a small extra error term chosen from a continuous Gaussian. Omitting many details, this faithfully simulates the LWE distribution for two reasons: first, $\mathbf{a}$ is essentially uniform over $\mathbb{Z}_q^n$ since $r \geq q \cdot \eta_\epsilon(\Lambda)$, and second,

$$\langle \mathbf{y}, \mathbf{x} \rangle \approx \langle \mathbf{y}, \mathbf{v} \rangle = \langle \mathbf{B}^t \mathbf{y}, \mathbf{B}^{-1} \mathbf{v} \rangle = \langle \mathbf{a}, \mathbf{s} \rangle \bmod q.$$

(The error distribution in the $\langle \mathbf{y}, \mathbf{x} \rangle$ term above requires some care to analyze precisely; we refer the reader to [Reg05] for the full details.) The oracle W solves for $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$ by hypothesis, and the entire vector $\mathbf{v}$ can be obtained by iterating the procedure as described in [Reg05, Lemma 3.5].

3.1.2 Proof of Main Theorem

Conceptually, the reduction claimed in Theorem 3.1 has two components. The first piece reduces GapSVP to a version of the bounded-distance decoding (BDD) problem. The second part is the reduction R from BDD to LWE described in Proposition 3.2, with a concrete implementation of the oracle D. Due to the additional hypotheses of the $\mathsf{GapSVP}_{\zeta,\gamma}$ problem that are needed throughout, we elect to present one integrated reduction, but remark that the GapSVP to BDD component has been abstracted out in [LM09].

In more detail, our reduction works as follows: given a lattice Λ, it perturbs a point $\mathbf{v} \in \Lambda$, invokes the reduction R from Proposition 3.2 on the perturbed point, and checks whether R successfully recovers $\mathbf{v}$. When $\lambda_1(\Lambda)$ is large, R must indeed recover $\mathbf{v}$ by hypothesis. When $\lambda_1(\Lambda)$ is small, $\mathbf{v}$ is *statistically hidden* and R must guess incorrectly with some non-negligible probability. (In effect, the reduction R is playing the role of the unbounded prover in the interactive Arthur-Merlin proof of Goldreich and Goldwasser [GG00].)

Proof of Theorem 3.1. The input to our reduction is an instance of $\mathsf{GapSVP}_{\zeta,\gamma}$, i.e., a pair $(\mathbf{B}, d)$ where $\min\|\tilde{\mathbf{b}}_i\| \geq 1$, the minimum distance $\lambda_1(\mathcal{L}(\mathbf{B})) \leq \zeta$, and $1 \leq d \leq \zeta/\gamma$. Let $\Lambda = \mathcal{L}(\mathbf{B})$.

The reduction runs the following procedure some large number $N = \mathrm{poly}(n)$ times.

1. Choose a point $\mathbf{w}$ uniformly at random from the ball $d' \cdot \mathcal{B}_n$ where $d' = d \cdot \sqrt{n/(4 \log n)}$, and let $\mathbf{x} = \mathbf{w} \bmod \mathbf{B}$.

2. Invoke the reduction R from Proposition 3.2 on $\mathbf{B}$ and $\mathbf{x}$ with parameter

$$r = \frac{q \cdot \sqrt{2n}}{\gamma \cdot d},$$

where the oracle D for sampling from $D_{\Lambda^*,r}$ is implemented by the algorithm from Proposition 2.8 on the reversed dual basis $\mathbf{D}$ of $\mathbf{B}$. Let $\mathbf{v}$ be R's output.

If $\mathbf{v} \neq \mathbf{x} - \mathbf{w}$ in any of the N iterations, then *accept*. Otherwise, *reject*.

We now analyze the reduction. First recall that $\max_i \|\tilde{\mathbf{d}}_i\| = 1/\min_i \|\tilde{\mathbf{b}}_i\| \leq 1$, and the parameter

$$r = \frac{q \cdot \sqrt{2n}}{\gamma \cdot d} \geq \frac{q \cdot \sqrt{2n}}{\zeta} \geq \omega(\sqrt{\log n})$$

by hypothesis on d and q, so the algorithm from Proposition 2.8 correctly samples from a distribution that is within negligible statistical distance of $D_{\Lambda^*,r}$.

Now consider the case when $(\mathbf{B}, d)$ is a NO instance, i.e., $\lambda_1(\Lambda) > \gamma \cdot d$. Then by Lemma 2.7, we have

$$\eta_\epsilon(\Lambda^*) \leq \frac{\sqrt{n}}{\gamma \cdot d}$$

for $\epsilon(n) = 2^{-n} = \text{negl}(n)$. Therefore $r \geq \sqrt{2}q \cdot \eta_\epsilon(\Lambda^*)$ as required by Proposition 3.2. Now because $\mathbf{x} - \mathbf{w} \in \Lambda$, the distance from $\mathbf{x}$ to Λ is at most

$$d' = d \cdot \sqrt{\frac{n}{4 \log n}} \leq \frac{\alpha \cdot \gamma \cdot d}{\sqrt{4n}} = \frac{\alpha q}{\sqrt{2} r},$$

by hypothesis on γ and the definition of r. Moreover, $\lambda_1(\Lambda) > \gamma \cdot d > 2d'$, so the reduction from Proposition 3.2 must return $\mathbf{v} = \mathbf{x} - \mathbf{w}$ in each of the iterations (with overwhelming probability), and the reduction rejects as desired.

Finally, consider the case when $(\mathbf{B}, d)$ is a YES instance, i.e., $\lambda_1(\Lambda) \leq d$. Let $\mathbf{z} \in \Lambda$ have norm $\|\mathbf{z}\| = \lambda_1(\Lambda)$. Consider an alternate experiment in which of $\mathbf{w}$ is replaced by $\mathbf{w}' = \mathbf{z} + \mathbf{w}$ for $\mathbf{w}$ chosen uniformly from $d' \cdot \mathcal{B}_n$, so $\mathbf{x}' = \mathbf{w}' \bmod \mathbf{B}$ and R is invoked on $\mathbf{x}'$. Then by Lemma 2.1 and the fact that statistical distance cannot increase under any randomized function, we have

$$\begin{aligned} \Pr[R(\mathbf{x}) = \mathbf{x} - \mathbf{w}] &\leq 1 - \tfrac{1}{\text{poly}(n)} + \Pr[R(\mathbf{x}') = \mathbf{x}' - \mathbf{w}'] \\ &\leq 2 - \tfrac{1}{\text{poly}(n)} - \Pr[R(\mathbf{x}') = \mathbf{x}' - \mathbf{w}]. \end{aligned}$$

But now notice that $\mathbf{x}' = \mathbf{z} + \mathbf{w} = \mathbf{w} \bmod \mathbf{B}$, so $\mathbf{x}'$ is distributed identically to $\mathbf{x}$ in the real experiment, and can replace $\mathbf{x}$ in the above expression. Rearranging, it follows that $\Pr[R(\mathbf{x}) = \mathbf{x} - \mathbf{w}] \leq 1 - 1/\text{poly}(n)$. Then for a sufficiently large $N = \text{poly}(n)$, we have $\mathbf{v} \neq \mathbf{x} - \mathbf{w}$ in at least one iteration and the reduction accepts, as desired. □

3.2 Variants of LWE

The next two lemmas reduce the worst-case search version of LWE to an average-case decision problem, which is more useful in cryptographic applications. The search to decision reduction in Lemma 3.3 is related to one given in [Reg05], which required the modulus q to be prime and bounded by $\text{poly}(n)$. The new lemma works for moduli that are the *product* of distinct $\text{poly}(n)$-bounded primes. Interestingly, it applies to the *continuous* version of LWE and requires the error distribution to be *Gaussian*, whereas the prior lemma worked for an *arbitrary discrete* distribution over $\mathbb{Z}_q$.

Lemma 3.3 (Worst-Case Search to Decision [Reg08])**.** *Let $n \geq 1$ be an integer, let $\alpha = \alpha(n) \in (0,1)$ and $\phi = \Psi_\alpha$, and let $q = q_1 \cdots q_t$ for distinct primes $q_j = \text{poly}(n)$ such that $q_j \geq \omega(\sqrt{\log n})/\alpha$. There is a probabilistic polynomial-time reduction from solving $\mathsf{LWE}_{q,\phi}$ with overwhelming probability to distinguishing between $A_{\mathbf{s},\phi}$ and $U(\mathbb{Z}_q^n \times \mathbb{T})$ for arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$ with overwhelming advantage.*

Proof. The factorization $q_1 \cdots q_t$ of q may computed efficiently because all the factors q_j are bounded by $\text{poly}(n)$. It is enough to give a method for checking whether the ith coordinate $s_i \in \mathbb{Z}_q$ of $\mathbf{s}$ is congruent to 0 modulo q_j, for arbitrary $i \in [n]$ and $j \in [t]$. This is because by transforming $A_{\mathbf{s},\phi}$ in the natural way, we can efficiently "shift" s_i by each value modulo q_j to discover $s_i \bmod q_j$ (because every $q_j = \text{poly}(n)$), and then recover each entry $s_i \in \mathbb{Z}_q$ via the Chinese remaindering.

To check if $s_i = 0 \bmod q_j$, consider the following transformation: given a pair $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{T}$, replace a_i with a random value modulo q_j, leaving its value modulo q/q_j the same. That is, let $\mathbf{a}' = \mathbf{a} + \mathbf{e}_i \cdot r \cdot (q/q_j)$ for some uniformly random $r \in \mathbb{Z}_{q_j}$ (where $\mathbf{e}_i \in \mathbb{Z}_q^n$ is the ith standard basis vector), and let $b' = b$. If $s_i = 0 \bmod q_j$, then the transformation maps $A_{\mathbf{s},\phi}$ to itself. Now suppose $s_i \neq 0 \bmod q_j$. Clearly $\mathbf{a}'$ is uniformly random; fix its value from here on. Because q_j is prime, b' is of the form

$$b' = \langle \mathbf{a}', \mathbf{s} \rangle / q + (r'/q_j + e) \in \mathbb{T}$$

for uniformly random $r' \in \mathbb{Z}_{q_j}$ and $e \leftarrow \phi = \Psi_\alpha$. Because $\alpha \geq \omega(\sqrt{\log n}) \cdot 1/q_j \geq \eta_\epsilon(\mathbb{Z} \cdot 1/q_j)$ for some $\epsilon = \text{negl}(n)$, the distribution of $r'/q_j + e \bmod 1$ is within $\text{negl}(n)$ statistical distance of uniform over $\mathbb{T}$ by [MR07, Lemma 4.1]. Distinguishing between the two cases therefore identifies whether $s_i = 0 \bmod q_j$, and we are done. □

Lemma 3.4 (Worst-Case to Average-Case Decision [Reg05, Lemma 4.1]). *Let $n, q \geq 1$ be integers and let ϕ be an arbitrary distribution on $\mathbb{T}$. There is a probabilistic polynomial-time reduction from distinguishing between $A_{\mathbf{s},\phi}$ and $U(\mathbb{Z}_q^n \times \mathbb{T})$ with overwhelming advantage for arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$, to distinguishing between $A_{\mathbf{s},\phi}$ and $U(\mathbb{Z}_q^n \times \mathbb{T})$ with non-negligible advantage for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$.*

Proof. The proof is via a standard amplification argument: repeatedly randomize the secret $\mathbf{s}$ by transforming samples $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{T}$ from $A_{\mathbf{s},\phi}$ into $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle / q)$ from $A_{\mathbf{s}+\mathbf{t},\phi}$ for a uniformly random $\mathbf{t} \in \mathbb{Z}_q^n$. Full details can be found in [Reg05]. □

4 Public-Key Cryptosystems

There are several prior LWE-based cryptosystems that enjoy semantic security against passive eavesdropping attacks [GM84]: the original scheme of Regev [Reg05], a more efficient amortized version [PVW08], and a "dual" (amortized) scheme that is the foundation for identity-based encryption [GPV08]. The security proofs for these schemes rely solely on the hypothesis that the (discretized) LWE distribution used in the scheme is pseudorandom, i.e., indistinguishable from uniform on the average. In Section 3 we established this pseudorandomness property (for moduli q of a certain form) assuming the worst-case hardness of $\mathsf{GapSVP}_{\zeta,\gamma}$, so all the prior proofs go through under that same assumption.

When using a large value of q (e.g., $q = 2^{O(n)}$), however, the efficiency of the prior schemes is suboptimal, because the plaintext expansion factor (even in the amortized schemes) is at least $\lg q$. Fortunately, improved efficiency is possible by discretizing the LWE distribution more "coarsely" using a relatively small modulus $q' = \text{poly}(n)$. Specifically, for any distribution D over $\mathbb{Z}_q^n \times \mathbb{T}$, define $\bar{D}$ to be the discretized distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_{q'}$ obtained by drawing a sample $(\mathbf{a}, b)$ from D and outputting $(\mathbf{a}, \bar{b} = \lfloor q' \cdot b \rceil \bmod q')$. Clearly, if D is uniform then $\bar{D}$ is also uniform. Therefore, the discretized distribution $\bar{A}_{\mathbf{s},\phi}$ is pseudorandom over $\mathbb{Z}_q^n \times \mathbb{Z}_{q'}$ if the continuous distribution $A_{\mathbf{s},\phi}$ is pseudorandom over $\mathbb{Z}_q^n \times \mathbb{T}$.

4.1 Example Scheme (Passive Security)

Due to the simplicity of its security proof and the similarity to our chosen ciphertext-secure schemes below, here we adapt a version of the "dual" cryptosystem from [GPV08] to our setting, and analyze it briefly. (The other schemes from [Reg05, PVW08] work out similarly.)

The parameters of the scheme are as follows. Fix integers n and $q \in [2, 2^{O(n)}]$. Let $m = (1 + \delta) n \lg q$ for some constant $\delta > 0$, let $q' \geq 2(m + 1)$ be bounded by $\text{poly}(n)$, and let ℓ be the message length (in bits). Fix $\alpha \in (0, 1)$ such that $1/\alpha \geq \sqrt{m + 1} \cdot \omega(\sqrt{\log n})$, and let $\phi = \Psi_\alpha$.

- **Key Generation.** Choose $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ uniformly at random, and secret key $\mathbf{X} \in \{0, 1\}^{m \times \ell}$ uniformly at random. The public key is $(\mathbf{A}, \mathbf{U} = \mathbf{AX}) \in \mathbb{Z}_q^{n \times (m + \ell)}$.
- **Encryption.** Given message $\mathbf{g} \in \{0, 1\}^\ell$, choose $\mathbf{s} \in \mathbb{Z}_q^n$ uniformly at random and $\mathbf{e}_1 \leftarrow \phi^m$, $\mathbf{e}_2 \leftarrow \phi^\ell$. Let $\mathbf{b}_1 = (\mathbf{A}^t \mathbf{s})/q + \mathbf{e}_1 \in \mathbb{T}^m$ and $\mathbf{b}_2 = (\mathbf{U}^t \mathbf{s})/q + \mathbf{e}_2 + \mathbf{g}/2 \in \mathbb{T}^\ell$. Output the ciphertext $(\bar{\mathbf{b}_1}, \bar{\mathbf{b}_2}) \in \mathbb{Z}_{q'}^{m+\ell}$, where $\bar{\mathbf{b}_i} = \lfloor q' \cdot \mathbf{b}_i \rceil \bmod q'$ for $i = 1, 2$.

- **Decryption.** Given ciphertext $(\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2)$ and secret key $\mathbf{X}$, compute $\mathbf{h} = \bar{\mathbf{b}}_2 - \mathbf{X}^t \bar{\mathbf{b}}_1 \in \mathbb{Z}_{q'}^{\ell}$. For each $i \in [\ell]$, let $g_i = 0$ if h_i is closer to 0 modulo q' than to $\lfloor q'/2 \rfloor$, otherwise let $g_i = 1$. Output $\mathbf{g} \in \{0,1\}^{\ell}$.

We first analyze the efficiency of the scheme. The amortized efficiency is optimized when $\ell = O(m)$, so suppose $\ell = m$ for simplicity. Then the public key size is $O(n^2 \log^2 q)$, and the plaintext expansion factor is $O(\log q') = O(\log n)$. (As observed in [KTX07, PVW08], it is even possible to reduce the expansion to $O(1)$ by fitting $\Omega(\log q')$ bits into each component of $\mathbf{g}_2$, at the expense of a somewhat smaller α.)

Correctness of the scheme (with overwhelming probability) follows by a routine argument bounding the accumulated error in the decryption algorithm, using the hypotheses on α and q', the bound $\|\mathbf{x}_i\| \leq \sqrt{m}$ on each column of $\mathbf{X}$, and the standard tail bound on Gaussians.

We briefly sketch the security proof. It relies solely on the hypothesis that $A_{\mathbf{s},\phi}$ is pseudorandom (for uniform $\mathbf{s} \in \mathbb{Z}_q^n$, which by the results of Section 3 follows from the worst-case hardness of (say) $\mathsf{GapSVP}_{\zeta,\gamma}$ for $\zeta = q > \gamma = \tilde{O}(n^{1.5}\sqrt{\log q})$. By a standard argument using the leftover hash lemma, the public key $(\mathbf{A}, \mathbf{U})$ is negligibly close to uniform. Then by construction, the public key together with the vectors $\mathbf{b}_1, \mathbf{b}_2$ (ignoring the $\mathbf{g}/2$ component of $\mathbf{b}_2$) constitute $m + \ell$ samples from $A_{\mathbf{s},\phi}$, which are indistinguishable from uniform by hypothesis. Therefore the view of the adversary is indistinguishable from uniform (for any message $\mathbf{g}$), which implies semantic security.

4.2 Chosen Ciphertext-Secure Scheme

4.2.1 Trapdoor Functions

Our CCA-secure cryptosystem is based on a collection of LWE-based injective trapdoor functions described in [GPV08], which are related to a proposal of [GGH97]. For completeness, and due to some modifications needed for the CCA application, we present a full description of the collection here.

The first component is a special algorithm for generating a (nearly) uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ that serves as the index of the public function $g_{\mathbf{A}}$, together with a trapdoor $\mathbf{T} \in \mathbb{Z}^{m \times m}$ made up of integer vectors whose lengths are bounded by some relatively small L.[2] Ajtai [Ajt99] gave the first such generation algorithm with a somewhat loose bound L, and Alwen and Peikert [AP09] recently gave improved algorithms that yield an optimal bound L (up to constant factors) for large enough m, or a somewhat looser bound for smaller m.

Proposition 4.1 ([AP09, Theorem 3.1 and 3.2]). *There is a probabilistic polynomial-time algorithm that, on input a positive integer n (in unary), positive integer $q \geq 2$ (in binary), and a $\mathrm{poly}(n)$-bounded positive integer $m \geq 2n \lg^2 q$, outputs a pair $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{T} \in \mathbb{Z}_q^{m \times m})$ such that:*

- *$\mathbf{A}$ is within $\mathrm{negl}(n)$ statistical distance of uniform,*
- *$\mathbf{AT} = \mathbf{0} \bmod q$, and*
- *$\|\mathbf{t}_i\| \leq 5\sqrt{n \lg q}$ for every $i \in [m]$.*

Alternately, for $m \geq 3(1+\delta) n \lg q$ for any $\delta > 0$, there is another algorithm that outputs $(\mathbf{A}, \mathbf{S})$ as above where $\|\mathbf{T}\| \leq m \cdot \omega(\sqrt{\log n})$ with overwhelming probability.

Let $m \geq (1+\delta) n \lg q$. A routine counting argument reveals that except with probability $q^n / 2^m = \mathrm{negl}(n)$ over the choice of a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, there does not exist any nonzero $\mathbf{s} \in \mathbb{Z}_q^n$ such

[2] As described in more detail in [Ajt99, GPV08, AP09], $\mathbf{T}$ can be seen as a full-rank set of short vectors in a certain lattice defined by $\mathbf{A}$, but that interpretation is not essential for this work.

that $\mathbf{A}^t\mathbf{s} = \mathbf{0} \in \mathbb{Z}_q^m$. That is, $\mathbf{s} \in \mathbb{Z}_q^n$ is uniquely determined by $\mathbf{y} = \mathbf{A}^t\mathbf{s}/q \in \mathbb{T}^m$. Furthermore, $\mathbf{s}$ may be recovered efficiently from $\mathbf{A}$ and $\mathbf{y}$ using, e.g., Gaussian elimination.

We now specify the family of trapdoor functions and their properties. The family $\{g_{\mathbf{A}} : \mathbb{Z}_q^n \times \mathbb{T}^m \to \mathbb{Z}_{q'}^m\}$ is parameterized by moduli q, q', a dimension m (satisfying the hypothesis in Proposition 4.1), and an error parameter $\alpha \in (0, 1)$.

- **Generation.** On security parameter n (in unary), run the algorithm from Proposition 4.1 to generate index $\mathbf{A}$ and trapdoor $\mathbf{T}$.
- **Evaluation.** On index $\mathbf{A}$ and inputs $\mathbf{s} \in \mathbb{Z}_q^n$ chosen uniformly at random and $\mathbf{x} \in \mathbb{T}^m$ chosen from Ψ_α^m, compute $\mathbf{b} = \mathbf{A}^t\mathbf{s}/q + \mathbf{x} \in \mathbb{T}^m$. Output $g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \bar{\mathbf{b}} = \lfloor \mathbf{b} \cdot q' \rceil \bmod q'$.
- **Inversion.** To invert $\bar{\mathbf{b}} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) \in \mathbb{Z}_{q'}^m$ given the trapdoor $\mathbf{T}$, let $\mathbf{b}' = \bar{\mathbf{b}}/q' \in \mathbb{T}^m$, compute
$$\mathbf{y} = \mathbf{T}^{-t} \cdot \lfloor \mathbf{T}^t \cdot \mathbf{b}' \rceil \bmod 1,$$
and compute $\mathbf{s}'$ from $\mathbf{y}$ as described above. Also output $\mathbf{x}' = \mathbf{b}' - (\mathbf{A}^t\mathbf{s})/q \in \mathbb{T}^m$ (the original value of $\mathbf{x}$ cannot always be recovered from $\bar{\mathbf{b}}$ due to rounding, but any consistent $\mathbf{x}'$ suffices for our applications).

Lemma 4.2. *Let $q' = q'(n) \geq 2L\sqrt{m}$ and $1/\alpha \geq L \cdot \omega(\sqrt{\log n})$. Then for any $\mathbf{s} \in \mathbb{Z}_q^n$ and for $\mathbf{x}$ drawn from Ψ_α^m, the inversion algorithm on $\bar{\mathbf{b}} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$ correctly outputs $\mathbf{s}$ with overwhelming probability over the choice of $\mathbf{x}$.*

Proof. We start with a few facts that we use later to analyze the rounding step. First, suppose $\mathbf{w} \in \mathbb{R}^m$ is such that $|w_i| \leq 1/(2q')$ for all $i \in [m]$. Then for all $i \in [m]$, we have
$$|\langle \mathbf{t}_i, \mathbf{w} \rangle| \leq \|\mathbf{t}_i\| \cdot \|\mathbf{w}\| \leq L \cdot \sqrt{m}/(2q') \leq 1/4$$
by the Cauchy-Schwarz inequality and by hypothesis on $\|\mathbf{t}_i\|$ and q'. Second, suppose $\mathbf{x}' \in \mathbb{R}^m$ is distributed according to D_α^m. Then for all $i \in [m]$, the inner product $\langle \mathbf{t}_i, \mathbf{x}' \rangle$ is distributed according to D_r for $r = \|\mathbf{t}_i\| \cdot \alpha \leq 1/\omega(\sqrt{\log n})$ by hypothesis on $\|\mathbf{t}_i\|$ and α. By the tail bound on Gaussian distributions, $|\langle \mathbf{t}_i, \mathbf{x}' \rangle| < 1/4$ except with probability $\exp(-\Omega(1/r^2)) = \mathrm{negl}(n)$.

Now consider the inversion algorithm on $\bar{\mathbf{b}} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$ where $\mathbf{x}$ is chosen from Ψ_α^m. By the definition of $g_{\mathbf{A}}$, there exist $\mathbf{w} \in \mathbb{R}^m$ with $|w_i| \leq 1/(2q')$ for all $i \in [m]$ and an $\mathbf{x}'$ distributed according to D_α^m such that
$$\bar{\mathbf{b}} = (\mathbf{A}^t\mathbf{s})/q + \mathbf{x}' + \mathbf{w} \bmod 1.$$

Thus,
$$\mathbf{T}^t \cdot \bar{\mathbf{b}} = (\mathbf{AT}/q)^t \cdot \mathbf{s} + \mathbf{T}^t \cdot (\mathbf{x}' + \mathbf{w}) \bmod \mathbf{T}^t.$$

Observe that $(\mathbf{AT}/q) = \mathbf{0} \bmod 1$ and $\mathbf{T}^t = \mathbf{0} \bmod 1$ by hypothesis on $\mathbf{T}$. Therefore,
$$\lfloor \mathbf{T}^t \cdot \mathbf{b}' \rceil = (\mathbf{AT}/q)^t\mathbf{s} + \lfloor \mathbf{T}^t(\mathbf{x}' + \mathbf{w}) \rceil = \mathbf{T}^t(\mathbf{A}^t\mathbf{s}/q) \bmod \mathbf{T}^t,$$
where the second inequality is with overwhelming probability over the choice of $\mathbf{x}'$ by the bounds established above. Finally, we see that $\mathbf{y} = \mathbf{T}^{-t} \cdot \lfloor \mathbf{T}^t \cdot \mathbf{b}' \rceil = (\mathbf{A}^t\mathbf{s}/q) \bmod 1$, and the inversion algorithm recovers $\mathbf{s}$ from $\mathbf{y}$. □

We remark that the inversion algorithm presented above works in *parallel* by rounding each entry of $\mathbf{T}^t \cdot \mathbf{b}'$ independently. An *iterative* rounding scheme akin to the "nearest-plane" algorithm of Babai [Bab86] can also be used, and succeeds (with overwhelming probability) whenever $\alpha(n) \leq 1/(\tilde{L} \cdot \omega(\sqrt{\log n}))$, where $\tilde{L} = \max_i \|\tilde{\mathbf{t}_i}\|$ is the norm of the longest vector in the *Gram-Schmidt orthogonalization* of $\mathbf{T}$. (The proof is virtually identical to the one given above.)

4.2.2 Chosen-Output Security

As shown in Lemma 4.2, the trapdoor functions described above are injective (with high probability) *if the input is chosen from the prescribed distribution*. However, when used in the context of an active chosen-ciphertext attack, the adversary may construct output values $\bar{\mathbf{b}} \in \mathbb{Z}_{q'}^m$ *adversarially*. We therefore need the functions to satisfy some additional properties.

Definition 4.3 (Chosen-Output Security)**.** Let G be a collection of trapdoor functions and let V be a deterministic polynomial-time algorithm, called the *preimage verifier* for G. We say that is (G, V) is *chosen-output secure* if the following properties hold with overwhelming probability over the choice of function g and trapdoor t from the collection G:

1. *Completeness.* For x chosen from the input distribution of g, and x' output by the inversion algorithm given $y = g(x)$ and t, $V(g, x', y)$ accepts with overwhelming probability over the choice of x.

2. *Unique preimage.* For every y, there is *at most one* legal preimage x of y under g, i.e., $V(g, x, y)$ accepts for at most one value of x.

3. *Findable preimage.* For any y, the inversion algorithm on input y and trapdoor t *always* outputs the unique legal preimage x of y, i.e., the x that makes $V(g, x, y)$ accept, if such x exists.

Chosen-output security ensures that for any value y in the range (possibly generated adversarially), the following two processes behave *identically*: (1) on input x, y (and the description of g), accept if $V(g, x, y)$ accepts; (2) on input y and the trapdoor, run the inverter to obtain some x, and then accept if $V(g, x, y)$ accepts. This identical behavior is a crucial property in the security proof for chosen-ciphertext attacks.

Making the functions chosen-output secure. Note that in the above description of the collection $\{g_{\mathbf{A}}\}$, *any* value $\mathbf{s} \in \mathbb{Z}_q^n$ is (part of) a potential preimage of $\mathbf{b}' \in \mathbb{T}^m$, under the (possibly very unlikely) error vector $\mathbf{x} = \mathbf{b}' - (\mathbf{A}^t\mathbf{s})/q \in \mathbb{T}^m$. Therefore, we need to restrict the notion of a legal preimage to satisfy Definition 4.3.

Define $|\cdot|$ on $\mathbb{T} = [0, 1)$ as $|x| = \min\{x, 1 - x\}$, and extend it coordinate-wise to $\mathbb{T}^m$. The preimage verifier V depends on the parameter α associated with the collection, and some arbitrary $t = t(n) = \omega(\sqrt{\log n})$. Define $V(\mathbf{A}, (\mathbf{s}, \mathbf{x}), \bar{\mathbf{b}})$ as follows: compute $\mathbf{b}' = \bar{\mathbf{b}}/q' \in \mathbb{T}^m$, and accept if every entry of $|\mathbf{x}|$ is strictly less than $\alpha \cdot t$, and if $\mathbf{b}' = (\mathbf{A}^t\mathbf{s})/q + \mathbf{x} \in \mathbb{T}^m$.

Lemma 4.4. *For $q' \geq 1/(\alpha \cdot t) \geq 2L \cdot \sqrt{m} \geq 8$, the preimage verifier V for the collection $\{g_{\mathbf{A}}\}$ satisfies Definition 4.3.*

Proof. For Property 1 (completeness), say $\mathbf{s} \in \mathbb{Z}_q^n$ is arbitrary and $\mathbf{x}$ is drawn from Ψ_α. Let $\bar{\mathbf{b}} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$. By Lemma 4.2 and construction, the inversion algorithm on $\bar{\mathbf{b}}$ outputs $(\mathbf{s}, \mathbf{x}')$ satisfying V's second test, with overwhelming probability over the choice of $\mathbf{x}$. Moreover, observe that for every $i \in [m]$, we have $|x_i| < \alpha \cdot t/2$ with overwhelming probability by the Gaussian tail bound. Additionally, every entry of $\left|\mathbf{b}' - ((\mathbf{A}^t\mathbf{s})/q + \mathbf{x})\right|$ is at most $1/(2q') \leq \alpha \cdot t/2$, so the property holds by the triangle inequality.

Property 2 (unique preimage) follows by a simple fact that holds with all but $q^n/2^m = \text{negl}(n)$ probability over the choice of $\mathbf{A}$: for every nonzero $\mathbf{s} \in \mathbb{Z}_q^n$, $(\mathbf{A}^t\mathbf{s})/q \bmod 1$ has at least one entry with absolute value greater than $1/4$. (This can be seen by analyzing the probability for any fixed nonzero $\mathbf{s}$, then invoking the union bound.) Then for any $\mathbf{b}' \in \mathbb{T}^m$ computed by V, there can be at most one $\mathbf{s} \in \mathbb{Z}_q^n$ such that every entry of $\left|\mathbf{b}' - (\mathbf{A}^t\mathbf{s})/q\right|$ is strictly less than $\alpha \cdot t \leq 1/8$.

For Property 3 (findable preimage), we observe that for any $\bar{\mathbf{b}}$ that has a legal preimage $(\mathbf{s}, \mathbf{x})$, we have $\|\mathbf{x}\| \leq \sqrt{m} \cdot \alpha \cdot t$ and

$$\mathbf{b}' = (\mathbf{A}^t \mathbf{s})/q + \mathbf{x} \bmod 1.$$

Then as in the proof of Lemma 4.2, we see that the inversion algorithm *always* correctly outputs $(\mathbf{s}, \mathbf{x})$ because $L \leq 1/(2\|\mathbf{x}\|)$ by hypothesis. □

Note that the parameter α in Lemma 4.4 is smaller than the one in Lemma 4.2 by a factor of $O(\sqrt{m})$, due to the "worst-case" inversion requirement imposed by the findable preimage property for chosen-output security. This yields an underlying worst-case approximation factor of $\gamma(n) = \tilde{O}(n/\alpha) = \tilde{O}(n \cdot L\sqrt{m})$, where L and m may be set according to Proposition 4.1.

4.2.3 Cryptosystem

To construct a cryptosystem that enjoys security under chosen-ciphertext attacks, we use the "witness recovering decryption" paradigm recently proposed by Peikert and Waters [PW08], and additional perspectives of Rosen and Segev [RS09]. The most important technical subtleties relating to LWE have already been addressed in the previous subsection, and we defer a complete description and proof to the full version.

The main observation is that any $k = \text{poly}(n)$ independently chosen functions $g_{\mathbf{A}_1}, \ldots, g_{\mathbf{A}_k}$ are pseudorandom (assuming LWE is hard) even when evaluated on the *same* input $\mathbf{s}$ and independent $\mathbf{x}_1, \ldots, \mathbf{x}_k$ (respectively) from the error distribution Ψ_α^m. This is because the indices $\mathbf{A}_1, \ldots, \mathbf{A}_k$ and outputs $\bar{\mathbf{b}}_1 = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}_1), \ldots, \bar{\mathbf{b}}_k = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}_k)$ can be simulated simply by drawing $k \cdot m$ samples from $A_{\mathbf{s},\Psi_\alpha}$ and discretizing. Essentially, this shows that our trapdoor functions are secure under "correlated inputs," as defined in [RS09]. For the CCA-secure cryptosystem constructed in [RS09], chosen-output security is sufficient for the proof to remain sound with our trapdoor functions. Moreover, the efficiency may be dramatically improved by observing that additional noisy inner products $\mathbf{u}_i, \langle \mathbf{u}_i, \mathbf{s} \rangle / q + e_i \in \mathbb{T}$ (for $e_i \leftarrow \Psi_\alpha$) are still pseudorandom, even given the above view. As in the cryptosystem from Section 4.1, we may use ℓ such inner products to encode an ℓ-bit ciphertext.

Acknowledgments

I thank Vadim Lyubashevsky, Daniele Micciancio, Oded Regev, and the anonymous STOC reviewers for very helpful discussions and comments.

References

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

[AD07] Miklós Ajtai and Cynthia Dwork. The first and fourth public-key cryptosystems with worst-case/average-case equivalence. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(97), 2007.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, pages 474–495, 2009.

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

[Ajt04] Miklós Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

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

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

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

[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.

[CPS09] David Cash, Chris Peikert, and Amit Sahai. Efficient circular-secure encryption from hard learning problems. Manuscript, 2009.

[GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.

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

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.

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

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, pages 31–51, 2008.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147, 1995.

[KS06] Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *FOCS*, pages 553–562, 2006.

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

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

[LM09] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. Manuscript, 2009.

[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.

[Pei08a] Chris Peikert. Limits on the hardness of lattice problems in ℓ_p norms. *Computational Complexity*, 17(2):300–351, May 2008. Preliminary version in CCC 2007.

[Pei08b] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(100), 2008.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 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.

[Reg08] Oded Regev, December 2008. Personal communication.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.