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

Chris Peikert*

September 1, 2009

Abstract

We construct public-key cryptosystems that are secure assuming the *worst-case* hardness of approximating the minimum distance of 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 based on lattices.

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

*School of Computer Science, Georgia Institute of Technology, cpeikert@cc.gatech.edu. Much of this work was completed while at the Computer Science Laboratory of SRI International. This material is based upon work supported by the National Science Foundation under Grants CNS-0716786 and CNS-0749931, and the Department of Homeland Security under contract number HSHQDC-07-C-00006. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views or policies, either expressed or implied, of the NSF or DHS.

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 [Reg04, 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], (hierarchical) identity-based encryption [GPV08, Pei09, CHK09], and others [PVW08, AGV09, ACPS09], as well as for hardness of learning results relating to halfspaces [KS06]. We emphasize that 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 $\alpha \in (0, 1)$.) 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 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 (given the state of the art in lattice algorithms) to be exponentially hard in the dimension n. In our reduction, the modulus q grows with ζ, so relying on this (potentially) easier variant of GapSVP permits 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, it also happens that for *Gaussian* error distributions, search- and decision-LWE are also equivalent when q is the product of many distinct small primes [Reg08]. Using this fact and adapting prior cryptosystems to different values 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 [Reg04]	$O(n^4)$	$O(n^2)$	unique-SVP*	$\tilde{O}(n^{1.5})$
LWE, $q = \mathrm{poly}(n)$	$\tilde{O}(n^2)$	$O(\log n)$	$\frac{\mathrm{poly}(n)\text{-to-}\gamma\text{-GapSVP}}{\text{GapSVP/SIVP (quantum)}}$	$\tilde{O}(n^{1.5})$
old CCA: $q = \mathrm{poly}(n)$ [PW08]	$n^{2+\delta}$	$O(\log n)$	same two choices ↑	$n^{5+\delta}$
new CCA: $q = \mathrm{poly}(n)$	$n^{2+\delta}$	$O(\log n)$	same two choices ↑	$\tilde{O}(n^2)$
new CCA: $q = 2^{O(n)}$	$n^{4+\delta}$	$O(\log n)$	GapSVP	$\tilde{O}(n^3)$

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]. While our reduction solves the *decision* problem GapSVP when q is very large (and progressively easier variants of GapSVP for smaller values of q), 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. In all cases, 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 $O(\log n)$ (or even $O(1)$, at the expense of slightly stronger assumptions [KTX07, PVW08]). 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, Reg04, AD07].

Using a main component of our reduction and several additional ideas, Lyubashevsky and Micciancio [LM09] recently showed that the unique-SVP and GapSVP problems are actually *equivalent*, up to an $O(\sqrt{n/\log n})$ factor loss in the approximation factor. This implies that prior cryptosystems based on unique-SVP [AD97, Reg04, 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 and expansion factors.

A preliminary version of this work [Pei08b] also included a more technical reduction showing that particular bits of the LWE secret $\mathbf{s}$ 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 search-based 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 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 having worst-case connections [AD97, Reg04, Reg05, AD07]. These works consider 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, Reg04, 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 rely 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, alluded to earlier.) The second component of the reduction is *quantum*, and uses an oracle for the BDD problem to sample lattice points according to a more-concentrated distribution. These samples 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, Regev's 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 a 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 style of argument 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 lattice reduction; prior works (e.g., [Ajt04, MR07, Reg05]) solve GapSVP by obtaining relatively short vectors in 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 (simplified) discussion so far has ignored one very important issue: 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 (this is why we lose the iterative structure of the overall reduction). Instead, our reduction employs a Gaussian sampling algorithm for lattices, recently developed in [GPV08], using the 'best available' basis for the input lattice. With an LLL-reduced 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 Gaussian 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 equivalent to solving search-LWE (resp., decision-LWE) given m noisy inner products. Moreover, as shown in [GPV08], the function $g_{\mathbf{A}}$ has a *trapdoor* with which the input $\mathbf{s}$ may be recovered efficiently from $\mathbf{b}$ (when χ is sufficiently concentrated). Concretely, the trapdoor is a 'short' basis for a certain lattice defined by $\mathbf{A}$, which can be generated 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 in this application. 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) [RS09]. However, the precise meaning of "injective" turns out to be quite subtle in this context, and the LWE-based trapdoor functions must be carefully modified to satisfy the necessary conditions for the security proof. In addition, we exploit the pseudorandomness of LWE to handle any desired message length efficiently.

1.4 Organization

The remainder of the paper is organized as follows. In Section 2 we recall notation, concepts, and basic results related to lattices and the learning with errors problem. In Section 3 we give our main reduction from the shortest vector problem to the learning with errors problem, and relate different versions of LWE to each other. In Section 4 we give cryptographic applications of LWE, including a new chosen ciphertext-secure cryptosystem.

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 $[n] = \{1, \ldots, n\}$. For real x, $\lfloor x \rceil$ denotes the integer closest to x with ties broken upward (e.g., $\lfloor 1.5 \rceil = 2$), and we extend $\lfloor \cdot \rceil$ coordinate-wise to real vectors. 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 $\mathrm{poly}(n)$ denote an unspecified polynomial function $f(n) = O(n^c)$ for some constant c. A function $f(n)$ is *negligible*, written $\mathrm{negl}(n)$, if $f(n) = o(n^{-c})$ for every constant c. We say that a probability is *overwhelming* if it is $1 - \mathrm{negl}(n)$.

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, and define $\|\mathbf{X}\| = \max_j \|\mathbf{x}_j\|$. 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, $\widetilde{\mathbf{S}} = \{\widetilde{\mathbf{s}_1}, \ldots, \widetilde{\mathbf{s}_n}\}$ denotes its *Gram-Schmidt orthogonalization*, defined iteratively as follows: $\widetilde{\mathbf{s}_1} = \mathbf{s}_1$, and for each $i = 2, \ldots, n$, $\widetilde{\mathbf{s}_i}$ is the orthogonal projection of $\mathbf{s}_i$ onto $\mathrm{span}^{\perp}(\mathbf{s}_1, \ldots, \mathbf{s}_{i-1})$. In matrix notation, $\mathbf{S}$ decomposes uniquely as $\mathbf{S} = \mathbf{QU}$ for some orthogonal matrix $\mathbf{Q}$ and upper-triangular matrix $\mathbf{U}$ with positive diagonal entries, and $\widetilde{\mathbf{s}_i} = u_{i,i} \cdot \mathbf{q}_i$.

In the context of duality it is often convenient to orthogonalize in the *reverse* order, defining $\widetilde{\mathbf{s}_n} = \mathbf{s}_n$ and each $\widetilde{\mathbf{s}_i}$ as the orthogonal projection of $\mathbf{s}_i$ onto $\mathrm{span}^{\perp}(\mathbf{s}_{i+1}, \ldots, \mathbf{s}_n)$ for $i = n-1, \ldots, 1$. In matrix notation, $\mathbf{S} = \mathbf{QL}$ where $\mathbf{Q}$ is again an orthogonal matrix and $\mathbf{L}$ is lower triangular with positive diagonal entries, and $\widetilde{\mathbf{s}_i} = l_{i,i} \cdot \mathbf{q}_i$. In this work, by default we orthogonalize in forward order unless specified otherwise.

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 D_r 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 $\exp(-\pi t^2)$. Moreover, for $\mathbf{x} \in \mathbb{R}^n$ distributed according to $D_r^{(n)}$, $\|\mathbf{x}\| \leq r \cdot \sqrt{m}$ except with probability at most 2^{-n}.

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 sufficiently large (polynomial) number of bits 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 drawing a sample from $D_\alpha^{(1)}$ and reducing it 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, given access to any desired $\mathrm{poly}(n)$ number of samples from $A_{\mathbf{s},\phi}$ for some arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$, to find $\mathbf{s}$ (with overwhelming probability).

Remark 2.3. The above definition of LWE has a 'worst-case' flavor, in that the vector $\mathbf{s} \in \mathbb{Z}_q^n$ is arbitrary and must be found with overwhelming probability. Using standard amplification and random self-reduction techniques (see, e.g., [Reg05, Lemma 4.1 and Lemma 3.6]), this worst-case form is equivalent (up to a polynomial factor in the number of samples consumed) to an 'average-case' version in which the goal is to find a *uniformly random* $\mathbf{s} \in \mathbb{Z}_q^n$ with any *non-negligible* probability given $A_{\mathbf{s},\phi}$, where the probability is taken over all the randomness in the experiment.

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

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{Bz} = \sum\nolimits_{i \in [n]} z_i \cdot \mathbf{b}_i \ : \ \mathbf{z} \in \mathbb{Z}^n \right\}.$$

(Technically, this is the definition of a *full-rank* lattice, which is all we are concerned with in this work.) The origin-centered *fundamental parallelepiped* $\mathcal{P}_{1/2}(\mathbf{B})$ of a basis $\mathbf{B}$ is the half-open region defined as

$$\mathcal{P}_{1/2}(\mathbf{B}) = \left\{ \mathbf{Bc} = \sum\nolimits_{i \in [n]} c_i \cdot \mathbf{b}_i \ : \ \mathbf{c} \in [-\tfrac{1}{2}, \tfrac{1}{2})^n \right\} \subset \mathbb{R}^n.$$

For a point $\mathbf{w} \in \mathbb{R}^n$ and basis $\mathbf{B}$, we write $\mathbf{x} = \mathbf{w} \bmod \mathbf{B}$ to denote the unique $\mathbf{x} \in \mathcal{P}_{1/2}(\mathbf{B})$ such that $\mathbf{w} - \mathbf{x} \in \mathcal{L}(\mathbf{B})$. Given input $\mathbf{B}$ and $\mathbf{w}$, the value $\mathbf{x} = \mathbf{w} \bmod \mathbf{B}$ may be computed efficiently as $\mathbf{x} = \mathbf{B}(\mathbf{B}^{-1}\mathbf{w} - \lfloor \mathbf{B}^{-1}\mathbf{w} \rceil)$.

The *minimum distance* $\lambda_1(\Lambda)$ of a lattice Λ (by default, in the ℓ_2 norm) is the minimum distance between any two distinct lattice points, or equivalently, the length of the shortest nonzero lattice vector:

$$\lambda_1(\Lambda) = \min_{\mathbf{x} \neq \mathbf{y} \in \Lambda} \|\mathbf{x} - \mathbf{y}\| = \min_{\mathbf{0} \neq \mathbf{x} \in \Lambda} \|\mathbf{x}\|.$$

It is well-known and easy to show that for any basis $\mathbf{B}$ of Λ, we have

$$\lambda_1(\Lambda) \geq \min_i \|\widetilde{\mathbf{b}_i}\|.$$

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, then the dual basis $\mathbf{B}^* = (\mathbf{B}^{-1})^t$ is in fact a basis of Λ^*. The following simple fact relates the Gram-Schmidt orthogonalizations of a basis and its dual; we include a brief proof for completeness.

Lemma 2.4. *Let* $\mathbf{B}$ *be an (ordered) basis, and let* $\mathbf{D} = \mathbf{B}^* = \mathbf{B}^{-t}$ *be its dual basis. Then* $\widetilde{\mathbf{d}}_i = \widetilde{\mathbf{b}}_i / \|\widetilde{\mathbf{b}}_i\|^2$ *for all* $i \in [n]$*, where* $\mathbf{B}$ *(respectively,* $\mathbf{D}$*) is orthogonalized in forward (respectively, reverse) order. In particular,* $\|\widetilde{\mathbf{d}}_i\| = 1/\|\widetilde{\mathbf{b}}_i\|$.

Proof. We may write the Gram-Schmidt decomposition of $\mathbf{B}$ as $\mathbf{B} = \mathbf{Q}\mathbf{U}$ for some orthogonal matrix $\mathbf{Q}$ and upper-triangular matrix $\mathbf{U}$ with positive diagonal entries, so $\widetilde{\mathbf{b}}_i = u_{i,i} \cdot \mathbf{q}_i$ and $\widetilde{\mathbf{b}}_i / \|\widetilde{\mathbf{b}}_i\|^2 = \mathbf{q}_i / u_{i,i}$.

Observe that $\mathbf{D} = \mathbf{B}^{-t} = \mathbf{Q}^{-t} \cdot \mathbf{U}^{-t} = \mathbf{Q} \cdot \mathbf{U}^{-t}$, because $\mathbf{Q}^{-t} = \mathbf{Q}$. Note that $\mathbf{U}^{-t}$ is lower triangular and its diagonal entries (in order) are $u_{i,i}^{-1} > 0$. Therefore, $\mathbf{D} = \mathbf{Q} \cdot \mathbf{U}^{-t}$ is indeed the Gram-Schmidt decomposition (in reverse order) of the dual basis $\mathbf{D}$, and $\widetilde{\mathbf{d}}_i = \mathbf{q}_i / u_{i,i}$ as desired. □

For our chosen ciphertext-secure cryptosystem we recall two related 'decoding' algorithms on lattices, originally due to Babai [Bab86]. The first is a simple rounding algorithm that, given a lattice basis $\mathbf{B} \subset \mathbb{R}^{n \times n}$ and point $\mathbf{x} \in \mathbb{R}^n$, outputs the lattice point $\mathbf{v} = \mathbf{B} \cdot \lfloor \mathbf{B}^{-1}\mathbf{x} \rceil \in \mathcal{L}(\mathbf{B})$. It is clear that the set of points in $\mathbb{R}^n$ for which this algorithm outputs a particular $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ is exactly $\mathbf{v} + \mathcal{P}_{1/2}(\mathbf{B})$.

The second algorithm is usually known as the "nearest-plane" algorithm. For simplicity we assume that the given basis is orthogonalized in forward order; otherwise, the algorithm's main loop should be reordered correspondingly. The algorithm works as follows: given a lattice basis $\mathbf{B}$ and point $\mathbf{x}$, let $\mathbf{y} \leftarrow \mathbf{x}$, and then for $i = n, \ldots, 1$, let $\mathbf{y} \leftarrow \mathbf{y} - c_i \mathbf{b}_j$ where

$$c_i = \lfloor \langle \mathbf{y}, \widetilde{\mathbf{b}}_i \rangle / \langle \widetilde{\mathbf{b}}_i, \widetilde{\mathbf{b}}_i \rangle \rceil \in \mathbb{Z}.$$

Finally, output $\mathbf{x} - \mathbf{y}$. The set of points in $\mathbb{R}^n$ for which this algorithm outputs a particular $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ is exactly $\mathbf{v} + \mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$.

2.2.1 Computational Problems on Lattices

We are mainly interested in variants of the shortest vector problem on lattices.

Definition 2.5 (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 $\lambda_1(\Lambda) > \gamma(n) \cdot d$.

Remark 2.6. Given an oracle for GapSVP_γ, the minimum distance of a lattice can be efficiently approximated 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 upon which our main worst- to average-case reduction is based.

Definition 2.7 (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:

1. $\mathbf{B}$ is a basis of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$ for which $\lambda_1(\Lambda) \leq \zeta(n)$,
2. $\min_i \|\widetilde{\mathbf{b}}_i\| \geq 1$, and
3. $1 \leq d \leq \zeta(n)/\gamma(n)$.

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

Remark 2.8. A few notes about Definition 2.7 are in order.

1. The second condition $\min\|\widetilde{\mathbf{b}_i}\| \geq 1$ implies that $\lambda_1(\Lambda) \geq 1$, and is without loss of generality by scaling the basis $\mathbf{B}$.

2. Similarly, the last condition $1 \leq d \leq \zeta(n)/\gamma(n)$ is also without loss of generality, because the instance is trivially solvable when d lies outside that range.

3. The first condition is therefore the interesting one. First observe that for any $\zeta(n) \geq 2^{n/2}$, $\mathsf{GapSVP}_{\zeta,\gamma}$ is *equivalent* to the standard GapSVP_{γ} problem: 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

$$\lambda_1(\Lambda) \leq \|\mathbf{b}_1\| \leq 2^{n/2} \cdot \min_i \|\widetilde{\mathbf{b}_i}\|,$$

which can then be scaled to make $\min_i\|\widetilde{\mathbf{b}_i}\| = 1$.

For functions $\zeta(n) \ll 2^{n/2}$, particularly $\zeta(n) = \mathrm{poly}(n)$, the first condition is 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 actually contain a vector of length at most $\zeta(n)$, though this is not necessary for our reduction 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 efficiently verifiable, and even for such a tight choice of parameters as $\zeta(n) = 2 \cdot \gamma(n)$.

2.2.2 Gaussians on Lattices

Micciancio and Regev [MR07] introduced a lattice quantity called the *smoothing parameter*, and related it to various lattice quantities.

Definition 2.9. 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 $\rho_{1/r}(\Lambda^* \backslash \{\mathbf{0}\}) \leq \epsilon$.

Note that the smoothing parameter scales with the lattice: for any real $c \neq 0$, $\eta_\epsilon(c \cdot \Lambda) = c \cdot \eta_\epsilon(\Lambda)$.

Lemma 2.10 ([MR07, Lemma 3.2])**.** *For any n-dimensional lattice Λ, we have $\eta_{2^{-n}}(\Lambda) \leq \sqrt{n}/\lambda_1(\Lambda^*)$.*

The following bound on the smoothing parameter of the integer lattice $\mathbb{Z}$ is a special case of [MR07, Lemma 3.3].

Lemma 2.11. *For any $f(n) = \omega(\sqrt{\log n})$, there exists a negligible $\epsilon = \epsilon(n)$ such that $\eta_\epsilon(\mathbb{Z}) \leq f(n)$.*

When the parameter r exceeds the smoothing parameter of a lattice Λ, the Gaussian measure ρ_r associated with any *translate* (coset) of Λ is essentially unchanged:

Lemma 2.12 ([MR07, implicit in Lemma 4.4])**.** *For any n-dimensional lattice Λ, any $\epsilon \in (0,1)$, $r \geq \eta_\epsilon(\Lambda)$, and $\mathbf{c} \in \mathbb{R}^n$, we have*

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

For an n-dimensional lattice Λ and real $r > 0$, define the *discrete Gaussian probability distribution over* Λ with parameter r (and centered at zero) as

$$D_{\Lambda,r}(\mathbf{x}) = \frac{\rho_r(\mathbf{x})}{\rho_r(\Lambda)} \quad \forall\, \mathbf{x} \in \Lambda.$$

(Note that the denominator in the above expression is merely a finite normalization factor.)

Our reduction uses, as a subroutine, an efficient algorithm that samples from discrete Gaussian distributions. (The Gram-Schmidt orthogonalization used in the algorithm may be performed in any known order.)

Proposition 2.13 ([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* $r \geq \max_i \|\widetilde{\mathbf{b}_i}\| \cdot \omega(\sqrt{\log n})$*, outputs a sample from a distribution that is within* $\mathrm{negl}(n)$ *statistical distance of* $D_{\Lambda,r}$*.*

3 Classical Hardness of LWE

In this section we show that certain versions 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* $\alpha = \alpha(n) \in (0,1)$ *and* $\gamma = \gamma(n) \geq n/(\alpha\sqrt{\log n})$*. Let* $\zeta = \zeta(n) \geq \gamma(n)$ *and* $q = q(n) \geq \zeta(n) \cdot \omega(\sqrt{\log n/n})$*.*

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* $\mathrm{poly}(n)$ *samples.*

Remark 3.2. $\mathsf{GapSVP}_{\zeta,\gamma}$ is potentially hard in the worst case whenever $\zeta > 2\gamma$, so Theorem 3.1 allows for a choice of q as small as

$$q > 2\gamma \cdot \omega(\sqrt{\log n/n}) = \omega(\sqrt{n}/\alpha).$$

Remark 3.3. Using bounds from [Pei08a] on the smoothing parameter in terms of other ℓ_p norms, Theorem 3.1 can easily be generalized to work for $\mathsf{GapSVP}_{\zeta,\gamma}$ in any ℓ_p norm, $2 < p \leq \infty$, for essentially the same approximation $\gamma(n)$ (up to $O(\log n)$ factors).

3.1.1 Regev's Classical Reduction

We rely crucially on the classical component of Regev's reduction, restated here. Briefly stated, this component reduces a (worst-case) *bounded-distance decoding* problem to the LWE problem.

Proposition 3.4 ([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 any basis* $\mathbf{B}$ *of an* n*-dimensional lattice* $\Lambda = \mathcal{L}(\mathbf{B})$*, any real* $r \geq \sqrt{2}q \cdot \eta_\epsilon(\Lambda^*)$*, and any target point* $\mathbf{x}$ *within distance* $\alpha q/(\sqrt{2}r) < \lambda_1(\Lambda)/2$ *of* Λ*, and given access to*

1. *an oracle* W *that solves* $\mathsf{LWE}_{q,\phi}$ *using* $\mathrm{poly}(n)$ *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.*

Remark 3.5. Because they are particularly important for the proof of our main theorem, we want to stress the effect of the main parameters α, q, and r on the decoding radius $\alpha q/(\sqrt{2}r)$ that the reduction R can handle. The parameters q and r trade off against each other so that if, for example, the oracle D samples from a very wide discrete Gaussian distribution (i.e., r is large), a correspondingly larger value of q can compensate without hurting the decoding radius. The LWE noise parameter α also plays a role, but because it is typically the inverse of some small polynomial, its effect on the decoding radius (or on q) is generally mild.

For self-containment, we give a brief description of the reduction R described in Proposition 3.4 (however, this is not required to understand the proof of our main theorem and may be safely skipped). Let $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$ be the coefficient vector of the $\mathbf{v} \in \Lambda$ closest to $\mathbf{x}$, reduced modulo q. To generate a sample from the LWE distribution $A_{\mathbf{s},\phi}$, the reduction uses its oracle D to obtain a sample $\mathbf{y} \leftarrow D_{\Lambda^*,r}$, 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 an additional small error term chosen from a continuous Gaussian. Finally, the oracle W is invoked on these samples to solve for $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$, then the entire vector $\mathbf{v}$ can be obtained by iterating the procedure as described in [Reg05, Lemma 3.5].

Omitting many details, the above procedure faithfully simulates the LWE distribution $A_{\mathbf{s},\phi}$ for two reasons: first, $\mathbf{a} \in \mathbb{Z}_q^n$ is essentially uniform because $r \geq q \cdot \eta_\epsilon(\Lambda)$; second, because $\mathbf{x}$ and $\mathbf{v}$ are relatively close and $\mathbf{y}$ is relatively short (and Gaussian),

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

Of course, the error distribution in the $\langle \mathbf{y}, \mathbf{x} \rangle$ term above must be analyzed precisely, which requires some care; we refer the reader to [Reg05] for the full details.

3.1.2 Proof of Main Theorem

Conceptually, the reduction described in our main theorem (Theorem 3.1) has two components. The first part reduces GapSVP to a type of bounded-distance decoding (BDD) problem. The second part reduces BDD to LWE using the reduction R described in Proposition 3.4, instantiating the oracle D with the discrete Gaussian sampler described in Proposition 2.13. Due to the additional hypotheses of the $\mathsf{GapSVP}_{\zeta,\gamma}$ problem that are needed throughout our reduction, we elect to present these two parts as an integrated whole, but note that the GapSVP to BDD component has been studied on its own in [LM09].

In a bit more detail, our reduction works as follows: given a lattice Λ, it first perturbs a point $\mathbf{v} \in \Lambda$, then invokes the reduction R from Proposition 3.4 on the perturbed point, and tests whether R successfully returns $\mathbf{v}$. The reduction works because when $\lambda_1(\Lambda)$ is large, R is obliged to return $\mathbf{v}$ by hypothesis. On the other hand, when $\lambda_1(\Lambda)$ is small, the perturbation *information-theoretically hides* $\mathbf{v}$ from R (in some precise sense), so R must guess incorrectly with noticeable probability. The behavior of R therefore allows us to distinguish between these two cases. (In effect, the reduction R may be seen as 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 $\min\|\widetilde{\mathbf{b}_i}\| \geq 1$, 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 3.4 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 2.13 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 $\max_i \|\widetilde{\mathbf{d}}_i\| = 1/\min_i \|\widetilde{\mathbf{b}}_i\| \leq 1$, 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 2.13 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 2.10, we have

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

for $\epsilon(n) = 2^{-n} = \mathrm{negl}(n)$. Therefore $r \geq \sqrt{2} q \cdot \eta_\epsilon(\Lambda^*)$ as required by Proposition 3.4. 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 3.4 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 $\mathbf{w}$ 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

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

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 $\Pr[R(\mathbf{x}) = \mathbf{x} - \mathbf{w}] \leq 1 - 1/\mathrm{poly}(n)$. Then for a sufficiently large $N = \mathrm{poly}(n)$, we have $\mathbf{v} \neq \mathbf{x} - \mathbf{w}$ in at least one iteration and the reduction accepts, as desired. $\square$

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 3.6 is related to prior reductions [BFKL93, Reg05], which require the modulus q to be prime and bounded by $\mathrm{poly}(n)$. In contrast, the new reduction works for "smooth" moduli that are the product of distinct (and sufficiently large) $\mathrm{poly}(n)$-bounded primes. Interestingly, the new reduction applies to the *continuous* version of LWE and makes explicit use of a *Gaussian* error distribution, whereas the prior ones work for *arbitrary* error distributions over $\mathbb{Z}_q$. (We thank Oded Regev [Reg08] for his contributions to this result.)

Lemma 3.6 (Worst-Case Search to Decision)**.** *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 $q_j = \mathrm{poly}(n)$ 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 $U = U(\mathbb{Z}_q^n \times \mathbb{T})$ 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 $\mathrm{poly}(n)$. 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 every $i \in [n]$ and $j \in [t]$), by the following argument: we can 'shift' $A_{\mathbf{s},\phi}$ into $A_{\mathbf{s}+\mathbf{t},\phi}$ for any known $\mathbf{t} \in \mathbb{Z}_q^n$ by mapping each sample $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle / q)$. Then given $A_{\mathbf{s},\phi}$ for unknown $\mathbf{s}$, we can discover the value of $s_i \bmod q_j$ for every i, j by shifting s_i by each residue t_i modulo q_j, and checking whether $s_i + t_i$ is 0. (Note that this is efficient because every $q_j = \mathrm{poly}(n)$). We can then fully recover each entry $s_i \in \mathbb{Z}_q$ via the Chinese remainder theorem.

To check if $s_i = 0 \bmod q_j$, consider the following transformation: given a pair $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{T}$, randomize a_i 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 $b' = b$. Observe that if $s_i = 0 \bmod q_j$, the transformation maps $A_{\mathbf{s},\phi}$ to itself.

Now suppose that $s_i \neq 0 \bmod q_j$. Clearly $\mathbf{a}' \in \mathbb{Z}_q^n$ is uniformly random; moreover, conditioned on any fixed value of $\mathbf{a}'$, the value $r \in \mathbb{Z}_{q_j}$ is also uniformly random. Now by construction, b' is of the form

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

for $e \leftarrow \phi = \Psi_\alpha$. Because q_j is prime, $r' = s_i \cdot r \bmod q_j$ is uniformly random in $\mathbb{Z}_{q_j}$.

Now consider the probability density function of $r'/q_j + e \in \mathbb{T}$. The density for any particular $z \in [0,1)$ is proportional to $\rho_\alpha(\mathbb{Z} \cdot 1/q_j + z)$. Because $\alpha \geq \omega(\sqrt{\log n}) \cdot 1/q_j \geq \eta_\epsilon(\mathbb{Z} \cdot 1/q_j)$ for some $\epsilon = \mathrm{negl}(n)$ by Lemma 2.11, the distribution of $r'/q_j + e \bmod 1$ is within $\mathrm{negl}(n)$ statistical distance of uniform over $\mathbb{T}$ by Lemma 2.12. Therefore, the transformation maps $A_{\mathbf{s},\phi}$ to U (up to negligible statistical distance), so distinguishing between $A_{\mathbf{s},\phi}$ and U identifies whether $s_i = 0 \bmod q_j$, and we are done. □

Lemma 3.7 (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. For self-containment we briefly sketch the proof, which is a standard amplification argument: note that we can randomize the secret $\mathbf{s}$ in the LWE distribution $A_{\mathbf{s},\phi}$ by transforming samples $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{T}$ into $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle / q)$ for a uniformly random $\mathbf{t} \in \mathbb{Z}_q^n$. This transformation also maps the uniform distribution to itself. Therefore, the distinguishing advantage for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ may be amplified to be overwhelming for all $\mathbf{s}$. □

4 Public-Key Cryptosystems

In this section we develop public-key encryption schemes that are based on the LWE problem. The remainder is organized as follows: in Section 4.1 we introduce additional useful notation and recall some necessary cryptographic definitions and background. In Section 4.2 we briefly consider prior LWE-based cryptosystems that are secure against *passive* (eavesdropping) attacks, adapted to our new hardness results. In Section 4.3 we develop a certain family of LWE-based trapdoor functions, which are at the heart of the *chosen ciphertext*-secure cryptosystem developed in Section 4.4.

4.1 Notation and Cryptographic Background

In our applications it will be helpful to have some additional notation. For a modulus $q' \geq 2$, let $\mathbb{T}_{q'} \subset \mathbb{T}$ be the additive subgroup of integer multiples of $\frac{1}{q'}$, i.e., $\mathbb{T}_{q'} = \{0, \frac{1}{q'}, \ldots, \frac{q'-1}{q'}\}$ with modulo-1 addition. For any $x \in \mathbb{T}$, let $\lfloor x \rceil_{q'} \in \mathbb{T}_{q'}$ denote the element of $\mathbb{T}_{q'}$ closest to x modulo 1, i.e., $\lfloor x \rceil_{q'} = \lfloor x \cdot q' \rceil / q' \bmod 1$. Extend $\lfloor \cdot \rceil_{q'}$ to vectors in the natural way.

For any distribution D over $\mathbb{Z}_q^n \times \mathbb{T}$, we can define the discretized distribution $\bar{D}$ over $\mathbb{Z}_q^n \times \mathbb{T}_{q'}$, which is obtained by drawing a sample $(\mathbf{a}, b)$ from D and outputting $(\mathbf{a}, \bar{b} = \lfloor b \rceil_{q'})$. Clearly, $\bar{D}$ is uniform if the original D is uniform (over $\mathbb{Z}_q^n \times \mathbb{T}$). Therefore, the discretized distribution $\bar{A}_{\mathbf{s},\phi}$ is pseudorandom over $\mathbb{Z}_q^n \times \mathbb{T}_{q'}$ if the distribution $A_{\mathbf{s},\phi}$ is pseudorandom over $\mathbb{Z}_q^n \times \mathbb{T}$.

We now recall some standard cryptographic concepts and definitions. For all schemes the main cryptographic security parameter is n, and all algorithms (including the adversary) are implicitly given the security parameter n in unary. For a (possibly interactive) algorithm $\mathcal{A}$ having binary output, we define its *distinguishing advantage* between two distributions $\mathcal{X}$ and $\mathcal{Y}$ to be $|\Pr[\mathcal{A}(\mathcal{X}) = 1] - \Pr[\mathcal{A}(\mathcal{Y}) = 1]|$.

We present our encryption schemes in the framework of *key encapsulation*, which simplifies the definitions and allows for more modular constructions. (For example, a KEM is often useful in "hybrid" encryption schemes that incorporate both asymmetric- and symmetric-key components.)

Definition 4.1 (Key-Encapsulation Mechanism (KEM))**.** A KEM for keys of length $\ell = \ell(n)$ is a triple of probabilistic polynomial-time algorithms as follows:

- $\mathsf{Gen}()$ outputs a public key pk and a secret key sk.
- $\mathsf{Encaps}(pk)$ outputs a key $\kappa \in \{0,1\}^\ell$ and its encapsulation $\tau \in \{0,1\}^*$.
- $\mathsf{Decaps}(sk, \tau)$ outputs a key κ.

The correctness requirement is: for $(pk, sk) \leftarrow \mathsf{Gen}()$ and $(\kappa, \tau) \leftarrow \mathsf{Encaps}(pk)$, $\mathsf{Decaps}(sk, \tau)$ should output κ with overwhelming probability (over all the randomness in the experiment).

We consider two notions of security for a KEM. The first is indistinguishability under a *passive* (chosen-plaintext) attack, often called *ind-cpa* security. The attack is defined as follows: generate $(pk, sk) \leftarrow \mathsf{Gen}()$, $(\kappa^*, \tau^*) \leftarrow \mathsf{Encaps}(pk)$, and $\kappa' \leftarrow \{0,1\}^\ell$ (chosen uniformly and independently of the other values). The advantage of an adversary $\mathcal{A}$ in the attack is defined as its distinguishing advantage between (pk, τ^*, κ^*) and (pk, τ^*, κ'). We say that a KEM scheme is ind-cpa-secure if for every probabilistic polynomial-time adversary $\mathcal{A}$, its advantage in the attack is $\mathrm{negl}(n)$.

The second, much stronger notion of security is indistinguishability under an *active* (chosen-ciphertext) attack, usually called *ind-cca* security [NY90, RS91]. The attack is defined as follows: generate $(pk, sk) \leftarrow \mathsf{Gen}()$, $(\kappa^*, \tau^*) \leftarrow \mathsf{Encaps}(pk)$, and $\kappa' \leftarrow \{0,1\}^\ell$. The advantage of an adversary $\mathcal{A}$ in the attack is defined as its distinguishing advantage between (pk, τ^*, κ^*) and (pk, τ^*, κ'), where $\mathcal{A}$ is *also* given access to an

oracle that computes $\mathsf{Decaps}(sk, \cdot)$ on all inputs *except* the challenge string τ^* (for which the oracle simply returns $\perp$). We say that a KEM is ind-cca-secure if for every probabilistic polynomial-time adversary $\mathcal{A}$, its advantage in the attack is $\mathrm{negl}(n)$.

Definition 4.2 (Injective Trapdoor Functions)**.** A family $\{g_a : D \to R\}$ of (probabilistic) *injective trapdoor functions* from a domain D to range R is given by a triple $\mathrm{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ of polynomial-time algorithms as follows:

- $\mathsf{Gen}()$ is a randomized algorithm that outputs a function description a and trapdoor t.
- $\mathsf{Eval}(a, s; x)$ is a randomized algorithm that evaluates the (randomized) function $g_a(s; x)$ on input $s \in D$ with randomness x (drawn from some efficiently sampleable distribution), outputting $b \in R$. (We often omit the randomness argument x from $\mathsf{Eval}(a, s)$ and $g_a(s)$, taking it to be implicit.)
- $\mathsf{Invert}(t, b)$ is a *deterministic* inversion algorithm that outputs some $s \in D$.

(To be completely formal, we consider an ensemble of families $\{g_a : D_n \to R_n\}_{n \in \mathbb{N}}$ indexed by the security parameter n, which is implicitly provided in unary to all algorithms.)

The correctness requirement is that for any $s \in D$, and for $(a, t) \leftarrow \mathsf{Gen}()$, $b \leftarrow \mathsf{Eval}(a, s)$, $\mathsf{Invert}(t, b)$ should output s with overwhelming probability (over all the randomness of the experiment). One security requirement is that (informally) the function $g_a(\cdot)$ should be one-way (hard to invert) without knowledge of t; we omit a precise security definition because our application will exploit some additional properties of the particular construction.

Definition 4.3 (Signature Scheme)**.** A *signature scheme* SS is a triple of probabilistic polynomial-time algorithms as follows:

- $\mathsf{Gen}()$ outputs a verification key vk and a signing key sk.
- $\mathsf{Sign}(sk, \mu)$, given sk and a message $\mu \in \{0,1\}^*$, outputs a signature $\sigma \in \{0,1\}^*$.
- $\mathsf{Ver}(vk, \mu, \sigma)$ either accepts or rejects the signature σ for message μ.

The correctness requirement is: for any message $\mu \in \mathcal{M}$, and for $(vk, sk) \leftarrow \mathsf{Gen}()$, $\sigma \leftarrow \mathsf{Sign}(sk, \mu)$, $\mathsf{Ver}(vk, \mu, \sigma)$ should accept with overwhelming probability (over all the randomness of the experiment).

The notion of security we require for our ind-cca application is *strong existential unforgeability* under a *one-time* chosen-message attack, or seu-1cma security. The attack is defined as follows: generate $(vk, sk) \leftarrow \mathsf{Gen}()$ and give vk to the adversary $\mathcal{A}$, which then outputs a message μ. Generate $\sigma \leftarrow \mathsf{Sign}(sk, \mu)$ and give σ to $\mathcal{A}$. The advantage of $\mathcal{A}$ in the attack is the probability that it outputs some $(\mu^*, \sigma^*) \neq (\mu, \sigma)$ such that $\mathsf{Ver}(vk, \mu^*, \sigma^*)$ accepts. We say that SS is seu-1cma-secure if for every probabilistic polynomial-time adversary $\mathcal{A}$, its advantage in the attack is $\mathrm{negl}(n)$.

An seu-1cma-secure signature scheme can be constructed from any one-way function [Gol04], and more efficiently from collision-resistant hash functions [HWZ07]. Both primitives are efficiently realizable under the same worst-case lattice assumptions that we use for our cryptosystems.

4.2 Passively Secure Cryptosystem

There are several prior LWE-based cryptosystems that enjoy semantic security against passive eavesdropping attacks (ind-cpa security): the original scheme of Regev [Reg05], a more efficient amortized version [PVW08], and a "dual" (also amortized) scheme that is the foundation for identity-based encryption [GPV08]. In all cases, the only complexity assumption in their security proof is that the (discretized) LWE distribution 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 assumption as well.

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-to-ciphertext expansion factor (even in the amortized schemes) is at least $\lg q$. Fortunately, it is possible to improve their efficiency (without sacrificing correctness) by discretizing the LWE distribution more 'coarsely' using a relatively small modulus $q' = \mathrm{poly}(n)$.

As an illustrative example, here we briefly adapt the "dual" cryptosystem from [GPV08] to our setting. (The schemes from [Reg05, PVW08] may be adapted similarly.) The cryptosystem is presented as a KEM (see Definition 4.1), with the following parameters as functions of the main security parameter n: let $q \geq 2$, $m = (1+\delta)n \lg q$ for some constant $\delta > 0$, $q' \geq 4(m+1)$, and $\ell \geq 1$ be integers with $\log q$, m, q', and ℓ all bounded by $\mathrm{poly}(n)$. Let $\alpha \in (0,1)$ be such that $1/\alpha \geq \sqrt{m+1} \cdot \omega(\sqrt{\log n})$, and let $\phi = \Psi_\alpha$.

- $\mathsf{Gen}()$: Choose $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ uniformly at random, and secret key $sk = \mathbf{X} \in \{0,1\}^{m \times \ell}$ uniformly at random. The public key is $pk = (\mathbf{A}, \mathbf{U} = \mathbf{AX}) \in \mathbb{Z}_q^{n \times (m+\ell)}$.
- $\mathsf{Encaps}(pk = (\mathbf{A}, \mathbf{U}))$: choose key $\mathbf{k} \in \{0,1\}^\ell$ and $\mathbf{s} \in \mathbb{Z}_q^n$ uniformly at random. Let
$$\bar{\mathbf{b}_1} = \lfloor (\mathbf{A}^t\mathbf{s})/q + \mathbf{x}_1 \rceil_{q'} \quad \text{and} \quad \bar{\mathbf{b}_2} = \lfloor (\mathbf{U}^t\mathbf{s})/q + \mathbf{x}_2 + \mathbf{k}/2 \rceil_{q'},$$
where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \leftarrow \Psi_\alpha^{m+\ell}$. Output the encapsulation $\bar{\mathbf{b}} = (\bar{\mathbf{b}_1}, \bar{\mathbf{b}_2}) \in \mathbb{T}_{q'}^{m+\ell}$.
- $\mathsf{Decaps}(sk = \mathbf{X}, \bar{\mathbf{b}} = (\bar{\mathbf{b}_1}, \bar{\mathbf{b}_2}))$: compute $\mathbf{h} = \bar{\mathbf{b}_2} - \mathbf{X}^t\bar{\mathbf{b}_1} \in \mathbb{T}_{q'}^\ell$. Output $\mathbf{k} \in \{0,1\}^\ell$, computed as follows: for each $i \in [\ell]$, let $k_i = 0$ if h_i is closer to 0 (modulo 1) than to $1/2$, otherwise let $k_i = 1$.

We briefly summarize the efficiency, correctness, and security analysis of the scheme, the details of which are by now quite routine (see [Reg05, PVW08, GPV08] for full proofs).

Efficiency. The amortized time- and space-efficiency of the scheme 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 of key material into each entry of $\bar{\mathbf{b}_2}$, at the expense of a smaller (but still inverse polynomial) parameter α.

Correctness. The scheme's correctness (with overwhelming probability) follows by bounding the accumulated error terms in each entry of the decapsulation algorithm's vector
$$\mathbf{h} = \bar{\mathbf{b}_2} - \mathbf{X}^t\bar{\mathbf{b}_1} \approx \mathbf{k}/2 + (\mathbf{U}^t\mathbf{s} - \mathbf{X}^t\mathbf{A}^t\mathbf{s})/q = \mathbf{k}/2 \in \mathbb{T}^\ell.$$
The accumulated error $\mathbf{h} - \mathbf{k}/2$ comes from two sources, the Gaussian distribution $\phi = \Psi_\alpha$ and the discretization (rounding) step. Because each $\|\mathbf{x}_i\| \leq \sqrt{m}$, and rounding a single entry introduces at most $1/(2q')$ error, the accumulated rounding error in each h_i is at most
$$\sqrt{m+1} \cdot \sqrt{m+1}/(2q') \leq 1/8$$

by the Cauchy-Schwarz inequality and by hypothesis on q'. Similarly, the accumulated Gaussian error (before rounding) in each h_i is distributed as a Gaussian with parameter at most $\alpha \cdot \sqrt{m+1} \leq 1/\omega(\sqrt{\log n})$, hence has magnitude less than $1/8$ with overwhelming probability by the Gaussian tail bound. Therefore, the total accumulated error in every h_i term is less than $1/4$ (except with negligible probability), and decapsulation successfully recovers $\mathbf{k} \in \{0,1\}^\ell$.

Security. The argument for ind-cpa security relies on the assumption that $A_{\mathbf{s},\phi}$ is pseudorandom on the average. (Using the results of Section 3, this assumption follows from the worst-case hardness of $\mathsf{GapSVP}_{\zeta,\gamma}$ whenever

$$q \geq \zeta > \gamma = \tilde{O}(n/\alpha) = \tilde{O}(n^{1.5}\sqrt{\log q}).)$$

By the leftover hash lemma [HILL99], the public key $(\mathbf{A}, \mathbf{U})$ is within negligible statistical distance of uniform over $\mathbb{Z}_q^{n \times (m+\ell)}$. Therefore, the public key together with the vectors $(\mathbf{A}^t\mathbf{s})/q + \mathbf{x}_1$, $(\mathbf{U}^t\mathbf{s})/q + \mathbf{x}_2$ constitute $m + \ell = \mathrm{poly}(n)$ independent samples from $A_{\mathbf{s},\phi}$, which are indistinguishable from uniform samples over $\mathbb{Z}_q^n \times \mathbb{T}$ by hypothesis. Thus the entire view of the adversary is indistinguishable from uniform for any encapsulated key $\mathbf{k}$, which implies ind-cpa security.

4.3 Trapdoor Functions

Our chosen ciphertext-secure cryptosystem is based on a family of LWE-based injective trapdoor functions similar to those described in [GPV08], which are related to the proposal of [GGH97]. Due to some significant modifications that are especially important for the security of our cryptosystem, we present a full description here.

4.3.1 Background

We start with some notation and geometric perspectives that are helpful for understanding the construction. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be arbitrary. We recall two integer lattices associated with $\mathbf{A}$:

$$\begin{aligned} \Lambda^\perp(\mathbf{A}) &= \{\mathbf{x} \in \mathbb{Z}^m \ : \ \mathbf{A}\mathbf{x} = \mathbf{0} \in \mathbb{Z}_q^n\} \\ \Lambda(\mathbf{A}) &= \{\mathbf{x} \in \mathbb{Z}^m \ : \ \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ such that } \mathbf{x} = \mathbf{A}^t\mathbf{s} \bmod q\} \end{aligned}$$

It can be verified that $\Lambda^\perp(\mathbf{A})$ and $\Lambda(\mathbf{A})$ are q-periodic, i.e., they both contain $q \cdot \mathbb{Z}^m$ as sublattices, and are dual to each other up to a scaling factor q. Alternately — and this is the perspective that we find most useful below — $\Lambda(\mathbf{A})/q$ is 1-periodic and is the dual lattice of $\Lambda^\perp(\mathbf{A})$.

Lemma 4.4. *Let $q \geq 2$ and let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be uniformly random. Then the probability that there exists a nonzero $\mathbf{s} \in \mathbb{Z}^n$ such that $\mathbf{A}^t\mathbf{s} = \mathbf{0} \bmod q$ is at most at most $q^n/2^m$. In particular, for $m \geq (1+\delta)n \lg q$ (where $\delta > 0$ is some constant), the probability is at most $q^{-\delta n} = \mathrm{negl}(n)$.*

Proof. We show that for any fixed nonzero $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{A}^t\mathbf{s} \neq \mathbf{0} \bmod q$ except with probability at most 2^{-m}; the claim follows by applying the union bound over all nonzero $\mathbf{s} \in \mathbb{Z}_q^n$.

Let $\mathbf{0} \neq \mathbf{s} \in \mathbb{Z}_q^n$, and assume without loss of generality that $s_1 \neq 0$. Define the (additive) subgroup $G \subseteq \mathbb{Z}_q$ as the set of $x \in \mathbb{Z}_q$ for which $x \cdot s_1 = 0 \bmod q$, and observe that $|G| \leq \gcd(s_1, q) \leq q/2$. Then for any fixed $a_2, \ldots, a_n$, the set of $a_1 \in \mathbb{Z}_q$ such that $\langle \mathbf{a}, \mathbf{s} \rangle = 0 \bmod q$ is either the empty set or is a coset of G, hence has size at most $q/2$. If follows that $\langle \mathbf{a}, \mathbf{s} \rangle = 0$ with probability at most $1/2$ over the uniformly random choice of $\mathbf{a}$, and $\mathbf{A}^t\mathbf{s} = \mathbf{0} \in \mathbb{Z}_q^m$ with probability at most 2^{-m}, as desired. □

Lemma 4.4 implies that with overwhelming probability over the choice of $\mathbf{A}$, the vector $\mathbf{s} \in \mathbb{Z}_q^n$ is uniquely determined by the lattice point $\mathbf{y} = (\mathbf{A}^t\mathbf{s})/q \in \Lambda(\mathbf{A})/q$. Furthermore, $\mathbf{s}$ may be recovered efficiently from $\mathbf{A}$ and $\mathbf{y}$ using, e.g., Gaussian elimination. For the remainder of the paper we implicitly ignore the rare possibility that these properties fail to hold for uniformly random $\mathbf{A}$.

Our trapdoor functions use a special algorithm for generating a (nearly) uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ together with a certain 'good' trapdoor basis $\mathbf{T} \in \mathbb{Z}^{m\times m}$ of $\Lambda^\perp(\mathbf{A})$ whose vectors (or whose Gram-Schmidt orthogonalized vectors $\widetilde{\mathbf{T}}$) are relatively short. Ajtai [Ajt99] gave the first such generation algorithm, and Alwen and Peikert [AP09] recently improved Ajtai's method to yield optimal bounds on the lengths of the basis vectors.

Proposition 4.5 ([AP09, Theorems 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,*
- *$\mathbf{T}$ is a basis of $\Lambda^\perp(\mathbf{A})$, and*
- *$\|\mathbf{T}\| \leq L = O(\sqrt{n\lg q})$.*

Alternately, for odd q and $m \geq 6n\lg q$, there is another algorithm that outputs $(\mathbf{A}, \mathbf{T})$ satisfying the first two properties above, where $\|\widetilde{\mathbf{T}}\| \leq \widetilde{L} = O(\sqrt{n\log q})$ with overwhelming probability.

4.3.2 Construction

We now describe our family of trapdoor functions and its properties. The family $\{g_{\mathbf{A}} : \mathbb{Z}_q^n \to \mathbb{T}_{q'}^m\}$ is a set of randomized functions parameterized by moduli q, q', a dimension m and bound L or $\widetilde{L}$ as described in Proposition 4.5, and an error parameter $\alpha \in (0, 1)$. The randomness in the evaluation of $g_{\mathbf{A}}(\mathbf{s}; \mathbf{x})$ is an error term $\mathbf{x} \in \mathbb{T}^m$ drawn from Ψ_α^m; recall that $\mathbf{x}$ is *not* considered to be an explicit input to the function, and in particular it need not (and will not) be recovered by the inversion algorithm.

- $\mathsf{Gen}()$: run the appropriate algorithm from Proposition 4.5 to generate function index $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and trapdoor basis $\mathbf{T} \in \mathbb{Z}^{m\times m}$ of $\Lambda^\perp(\mathbf{A})$ such that $\|\mathbf{T}\| \leq L$ or $\|\widetilde{\mathbf{T}}\| \leq \widetilde{L}$, whichever is desired. (The trapdoor information also implicitly includes the function index $\mathbf{A}$.)
- $\mathsf{Eval}(\mathbf{A}, \mathbf{s})$: choose $\mathbf{x} \leftarrow \Psi_\alpha^m$ and output

$$\bar{\mathbf{b}} = g_{\mathbf{A}}(\mathbf{s}; \mathbf{x}) = \lfloor (\mathbf{A}^t\mathbf{s})/q + \mathbf{x} \rceil_{q'} \in \mathbb{T}_{q'}^m.$$

- $\mathsf{Invert}(\mathbf{T}, \bar{\mathbf{b}})$: do one of the following to compute some $\mathbf{y} \in \Lambda(\mathbf{A})/q$:
  - *First option:* run Babai's simple rounding algorithm with basis $\mathbf{T}^{-t}$ and point $\bar{\mathbf{b}}$. That is, compute

$$\mathbf{y} = \mathbf{T}^{-t} \cdot \lfloor \mathbf{T}^t \cdot \bar{\mathbf{b}} \rceil \in \Lambda(\mathbf{A})/q.$$

  - *Second option:* run the nearest-plane algorithm with basis $\mathbf{T}^{-t}$ and point $\bar{\mathbf{b}}$, yielding $\mathbf{y} \in \Lambda(\mathbf{A})/q$.

  Finally, compute and output $\mathbf{s} \in \mathbb{Z}_q^n$ such that $(\mathbf{A}^t\mathbf{s})/q = \mathbf{y} \bmod 1$.

Lemma 4.6. *Let $q' \geq 2L\sqrt{m}$ and $1/\alpha \geq L \cdot \omega(\sqrt{\log n})$. Then for any $\mathbf{s} \in \mathbb{Z}_q^n$, the first (simple rounding) algorithm $\mathsf{Invert}(\mathbf{T}, \bar{\mathbf{b}} = g_{\mathbf{A}}(\mathbf{s}; \mathbf{x}))$ correctly outputs $\mathbf{s}$ with overwhelming probability over the choice of $\mathbf{x} \leftarrow \Psi_\alpha^m$.*

The same is true for the second (nearest-plane) inversion algorithm, with $\widetilde{L}$ replacing L in the above bounds for q' and α.

Proof. Consider the first (simple rounding) inversion algorithm, and let $\mathbf{D} = \mathbf{T}^{-t}$ be the dual basis of the lattice $\Lambda(\mathbf{A})/q$. By the behavior of the simple rounding algorithm, it suffices to show that $\bar{\mathbf{b}} = g_{\mathbf{A}}(\mathbf{s}; \mathbf{x}) \in \mathbb{T}_{q'}^m$ lies in $\mathbf{y} + \mathcal{P}_{1/2}(\mathbf{D})$ for $\mathbf{y} = (\mathbf{A}^t\mathbf{s})/q \in \Lambda(\mathbf{A})/q$.

By definition of the evaluation procedure, we can write $\bar{\mathbf{b}}$ as

$$\bar{\mathbf{b}} = (\mathbf{A}^t\mathbf{s})/q + (\mathbf{x} + \mathbf{w}) \bmod 1$$

for some Gaussian error term $\mathbf{x} \leftarrow D_\alpha^m$ and some rounding term $\mathbf{w} \in \mathbb{R}^m$ such that $\|\mathbf{w}\| \leq \sqrt{m}/(2q') \leq 1/(4L)$. It therefore suffices to show that with overwhelming probability both $\mathbf{w}, \mathbf{x} \in \frac{1}{2}\mathcal{P}_{1/2}(\mathbf{D})$, which is equivalent to the condition $\mathbf{T}^t\mathbf{w}, \mathbf{T}^t\mathbf{x} \in [-\frac{1}{4}, \frac{1}{4})^n$.

First, observe that for all $i \in [n]$,

$$|\langle \mathbf{t}_i, \mathbf{w} \rangle| \leq \|\mathbf{t}_i\| \cdot \|\mathbf{w}\| \leq L/(4L) = 1/4$$

with certainty, by the Cauchy-Schwarz inequality. Next, $\langle \mathbf{t}_i, \mathbf{x} \rangle$ is distributed as the Gaussian $D_{\alpha \cdot \|\mathbf{t}_i\|}$, where $\alpha \cdot \|\mathbf{t}_i\| \leq 1/\omega(\sqrt{\log n})$ by hypothesis. Then by the tail bound on Gaussian distributions, $|\langle \mathbf{t}_i, \mathbf{x}' \rangle| < 1/4$ except with negligible probability.

For the second (nearest-plane) inversion algorithm, the exact same argument applies, with $\widetilde{\mathbf{T}}$ and $\widetilde{L}$ replacing $\mathbf{T}$ and L, respectively. □

4.3.3 Chosen-Output Security

As shown in Lemma 4.6, the trapdoor functions described above are injective (with high probability) *when the function is computed with the prescribed error distribution*. However, in the context of a chosen-ciphertext attack, the adversary may construct output values $\bar{\mathbf{b}} \in \mathbb{T}_{q'}^m$ *adversarially*. We need the behavior of the inversion algorithm on such adversarial outputs not to leak any information about the secret trapdoor basis; the notion of *chosen-output security* ensures that this is the case.

Definition 4.7 (Chosen Output-Secure TDF)**.** Augment Definition 4.2 of a family $\{g_a : D \to R\}$ of trapdoor functions given by $\mathrm{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert})$ with an additional *deterministic* polynomial-time algorithm $\mathsf{Ver}(a, s, b)$, called the *preimage verifier*, which takes as input a function description a, an input $s \in D$, and an output $b \in R$.

We say that the family is *chosen output-secure* if the following properties hold with overwhelming probability over the choice of $(a, t) \leftarrow \mathsf{Gen}()$:

1. *Completeness.* For any $x \in D$, $\mathsf{Ver}(a, s, b)$ accepts with overwhelming probability over the choice of $b \leftarrow g_a(s)$.

2. *Unique preimage.* For every $b \in R$, there is *at most one* value of s for which $V(a, s, b)$ accepts.

3. *Findable preimage.* For any $b \in R$, if b has a legal preimage s (i.e., for which $V(a, s, b)$ accepts), then $\mathsf{Invert}(t, b)$ outputs s.

The definition of chosen-output security immediately implies the following property, which holds with overwhelming probability over the choice of $(a, t) \leftarrow \mathsf{Gen}()$: for *any* value $b \in R$ (possibly generated adversarially), the following two deterministic processes produce the same output:

1. *Invert a given output and verify:* Given a, t, and b, let $s = \mathsf{Invert}(t, b)$; accept if $\mathsf{Ver}(a, s, b)$ accepts.
2. *Verify a given input/output pair:* Given a, s, and b, accept if $\mathsf{Ver}(a, s, b)$ accepts.

This identical behavior is the crucial property in the security proof for chosen-ciphertext attacks.

Realizing chosen-output security. Note that in the description of the family $\{g_{\mathbf{A}}\}$ from Section 4.3.2 above, *any* $\mathbf{s} \in \mathbb{Z}_q^n$ is a potential preimage of any $\bar{\mathbf{b}} \in \mathbb{T}_{q'}^m$, under the (perhaps very unlikely) error vector $\mathbf{x} = \bar{\mathbf{b}} - (\mathbf{A}^t\mathbf{s})/q \in \mathbb{T}^m$. Therefore, we need to restrict the notion of a legal preimage to satisfy Definition 4.7. We do so by requiring the output $\bar{\mathbf{b}}$ to be sufficiently close (modulo 1) to the lattice point $(\mathbf{A}^t\mathbf{s})/q$ corresponding to the purported preimage $\mathbf{s}$.

In more detail, for $x \in \mathbb{T}$ represented as a real $\hat{x} \in [0, 1)$, define $|x| = \min\{\hat{x}, 1 - \hat{x}\} \in \mathbb{R}$, and extend $|\cdot|$ coordinate-wise to $\mathbb{T}^m$. The preimage verifier $\mathsf{Ver}_\alpha(\mathbf{A}, \mathbf{s}, \bar{\mathbf{b}})$ is parameterized by the α associated with the collection, and operates as follows: let $\mathbf{x}' = |\bar{\mathbf{b}} - (\mathbf{A}^t\mathbf{s})/q| \in \mathbb{R}^m$, and accept if $\|\mathbf{x}'\| \leq 2\alpha\sqrt{m}$; otherwise, reject. (Implicitly, Ver_α rejects if any input is malformed, e.g., not from the appropriate domain.)

Lemma 4.8. *For $q' \geq 1/(2\alpha)$ and $1/\alpha > 4L\sqrt{m}$ (or $1/\alpha > 4\widetilde{L}\sqrt{m}$), the collection $\{g_{\mathbf{A}}\}$ using the first (respectively, second) inversion algorithm and preimage verifier Ver_α is chosen-output secure, i.e., satisfies Definition 4.7.*

Proof. First we prove Property 1 (completeness). For $\bar{\mathbf{b}} \leftarrow g_{\mathbf{A}}(\mathbf{s})$, we can write $\bar{\mathbf{b}}$ as

$$\bar{\mathbf{b}} = (\mathbf{A}^t\mathbf{s})/q + (\mathbf{x} + \mathbf{w}) \bmod 1$$

for some Gaussian error term $\mathbf{x} \leftarrow D_\alpha^m$ and some rounding term $\mathbf{w} \in \mathbb{R}^m$ such that $\|\mathbf{w}\| \leq \sqrt{m}/(2q') \leq \alpha\sqrt{m}$. By the tail bound on m-dimensional Gaussians, we have $\|\mathbf{x}\| \leq \alpha\sqrt{m}$ except with $2^{-m} = \mathrm{negl}(n)$ probability, hence $\|\mathbf{x} + \mathbf{w}\| \leq 2\alpha\sqrt{m} < 1/2$. Therefore, $\mathsf{Ver}_\alpha(\mathbf{A}, \mathbf{s}, \bar{\mathbf{b}})$ computes its $\mathbf{x}'$ as $\mathbf{x}' = \mathbf{x} + \mathbf{w}$, and accepts.

We now prove Property 2 (unique preimage). Let $\mathbf{D} = \mathbf{T}^{-t}$ be the dual basis of $\mathbf{T}$. In order for $\mathsf{Ver}_\alpha(\mathbf{A}, \mathbf{s}, \bar{\mathbf{b}})$ to accept, there must exist an $\mathbf{x}' \in \mathbb{R}^m$ such that $\|\mathbf{x}'\| \leq 2\alpha\sqrt{m}$ and $\bar{\mathbf{b}} = (\mathbf{A}^t\mathbf{s})/q + \mathbf{x}' \bmod 1$. Now because $\mathbf{A}^t\mathbf{s} = \mathbf{0} \bmod q$ has no nonzero solutions $\mathbf{s} \in \mathbb{Z}_q^n$, the regions $(\mathbf{A}^t\mathbf{s})/q + \mathcal{P}(\mathbf{D}) \bmod 1$ are disjoint for each distinct $\mathbf{s} \in \mathbb{Z}_q^n$. (In fact, these regions exactly partition $\mathbb{T}^m$.) Moreover, $\mathcal{P}(\mathbf{D})$ contains the Euclidean ball of radius $2\alpha\sqrt{m}$, because for any $\mathbf{x}'$ in the ball we have $|\langle \mathbf{t}_i, \mathbf{x}' \rangle| \leq L \cdot 2\alpha\sqrt{m} < 1/2$ by the Cauchy-Schwarz inequality. Therefore, Ver_α accepts at most one preimage $\mathbf{s}$ for any value $\bar{\mathbf{b}}$.

To prove Property 3 (findable preimage), suppose that $\bar{\mathbf{b}}$ and $\mathbf{s}$ are such that $\mathsf{Ver}_\alpha(\mathbf{A}, \mathbf{s}, \bar{\mathbf{b}})$ accepts. Then there exists an $\mathbf{x}' \in \mathbb{R}^m$ such that $\|\mathbf{x}'\| \leq 2\alpha\sqrt{m} < 1/2$ and

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

To see that the first (simple rounding) inversion algorithm returns $\mathbf{s}$ given $\mathbf{T}$ and $\bar{\mathbf{b}}$, simply note that $\mathbf{x}' \in \mathcal{P}(\mathbf{D})$ as shown in the prior paragraph, so the inverter returns $\mathbf{s}$. (Likewise, for the nearest-plane inversion algorithm, $\mathbf{x}' \in \mathcal{P}(\widetilde{\mathbf{D}})$.) $\square$

Remark 4.9. The bound on the parameter $1/\alpha$ in Lemma 4.8 (for chosen-output security) is a $\tilde{\Theta}(\sqrt{m})$ factor larger than the one in Lemma 4.6 (for correct inversion). This is due to the 'worst-case' nature of the findable-preimage property, versus the 'average-case' nature of the inversion task when the function $g_{\mathbf{A}}$ is evaluated with the prescribed error distribution. Using the bounds on m and $\widetilde{L}$ from Proposition 4.5, the worst-case approximation factor for GapSVP underlying our chosen output-secure trapdoor functions can be as small as

$$\gamma(n) = \tilde{O}(n/\alpha) = \tilde{O}(n \cdot \widetilde{L}\sqrt{m}) = \tilde{O}(n^2 \log q).$$

4.4 Chosen Ciphertext-Secure 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], some additional ideas of Rosen and Segev [RS09], and specific properties of the LWE problem.

A key observation for our scheme is the following: assuming that the LWE distribution $A_{\mathbf{s},\phi}$ is pseudorandom on the average, any $k = \mathrm{poly}(n)$ independently chosen functions $g_{\mathbf{A}_1}, \ldots, g_{\mathbf{A}_k}$ are also pseudorandom, even when evaluated on the *same* input $\mathbf{s} \in \mathbb{Z}_q^n$ (but with independent error terms $\mathbf{x}_1, \ldots, \mathbf{x}_k$, respectively). This is because the function indices $\mathbf{A}_i$ and outputs $\bar{\mathbf{b}}_i = g_{\mathbf{A}_i}(\mathbf{s}; \mathbf{x}_i)$ are simply made up of $k \cdot m$ independent (discretized) samples from the LWE distribution $A_{\mathbf{s},\phi}$. Essentially, this means that our trapdoor functions are one-way and pseudorandom under "correlated inputs," a notion defined in [RS09] as a relaxation of "lossy" trapdoor functions [PW08]. Moreover, even given all of the above function indices and outputs, additional samples from $A_{\mathbf{s},\phi}$ remain pseudorandom as well.

Rosen and Segev [RS09] construct an ind-cca-secure cryptosystem based on any family of injective trapdoor functions that are one-way under a general notion of correlated inputs (which captures the type enjoyed by our functions). Their construction closely resembles that of Dolev, Dwork and Naor [DDN00], but uses such trapdoor functions (instead of a generic ind-cpa-secure encryption scheme) to achieve witness-recovering decryption as in [PW08]. However, the construction and proof from [RS09] implicitly assume several additional properties of the trapdoor functions, e.g., that preimages are unique even for adversarially chosen outputs, and always findable given the trapdoor. In addition, their scheme uses generic hard-core bits to conceal the encrypted message, which, while completely general, limits the scheme to short messages.

A careful reading of the proof from [RS09] reveals that all of the necessary properties of the trapdoor functions are indeed guaranteed by our definition of chosen-output security. Moreover, the pseudorandomness of $A_{\mathbf{s},\phi}$ can be exploited to handle arbitrarily long messages with good efficiency (just as in the passively secure scheme from Section 4.2). Aside from these two main differences, our construction and proof are essentially identical to those in [RS09]. For completeness, we give a complete description of our scheme and a (brief) proof of security.

4.4.1 Construction

Our scheme uses the chosen output-secure trapdoor functions given by $\mathrm{TDF} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert}, \mathsf{Ver}_\alpha)$ constructed above in Section 4.3.3. It also relies upon a strongly unforgeable one-time signature scheme $\mathrm{SS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ whose verification keys are exactly k bits long. The description of the scheme is as follows.

- $\mathsf{Gen}()$: select $\mathbf{U} \in \mathbb{Z}_q^{n \times \ell}$ uniformly at random, and for each $i \in [k]$ and $b \in \{0,1\}$, generate $(\mathbf{A}_{i,b}, \mathbf{T}_{i,b}) \leftarrow \mathrm{TDF}.\mathsf{Gen}()$.[2] Output public key $pk = (\{\mathbf{A}_{i,b}\}, \mathbf{U})$ and secret key $sk = (\mathbf{T}_{1,0}, \mathbf{T}_{1,1}, pk)$.

[2] Strictly speaking, we need only generate trapdoor information $\mathbf{T}_{i,b}$ for $i = 1$, and all the other $\mathbf{A}_{i,b} \in \mathbb{Z}_q^{n \times m}$ may be generated

- $\mathsf{Encaps}(pk)$: generate $(vk, sk_{\mathrm{SS}}) \leftarrow \mathsf{SS.Gen}()$. Choose key $\mathbf{k} \in \{0,1\}^\ell$ and $\mathbf{s} \in \mathbb{Z}_q^n$ uniformly at random. For each $i \in [k]$, let $\bar{\mathbf{b}_i} \leftarrow \mathsf{TDF.Eval}(\mathbf{A}_{i,vk_i}, \mathbf{s})$, and let

$$\bar{\mathbf{b}_0} = \lfloor (\mathbf{U}^t\mathbf{s})/q + \mathbf{x}_0 + \mathbf{k}/2 \rceil_{q'},$$

where $\mathbf{x}_0 \leftarrow \Psi_\alpha^\ell$. Let $\bar{\mathbf{b}} = (\bar{\mathbf{b}_0}, \bar{\mathbf{b}_1}, \ldots, \bar{\mathbf{b}_k}) \in \mathbb{T}_{q'}^{km+\ell}$. Compute a signature $\sigma \leftarrow \mathsf{SS.Sign}(sk_{\mathrm{SS}}, \bar{\mathbf{b}})$. Finally, output the encapsulation

$$\tau = (vk, \bar{\mathbf{b}}, \sigma).$$

- $\mathsf{Decaps}(sk, \tau = (vk, \bar{\mathbf{b}}, \sigma))$: perform each of the following steps:

  1. Parse $\bar{\mathbf{b}}$ as $(\bar{\mathbf{b}_0}, \bar{\mathbf{b}_1}, \ldots, \bar{\mathbf{b}_k})$, where $\bar{\mathbf{b}_i} \in \mathbb{T}_{q'}^m$ for $i \in [k]$ and $\bar{\mathbf{b}_0} \in \mathbb{T}_{q'}^\ell$ (if $\bar{\mathbf{b}}$ cannot be parsed in this way, output $\perp$).
  2. Run $\mathsf{SS.Ver}(vk, \bar{\mathbf{b}}, \sigma)$, and output $\perp$ if Ver rejects.
  3. Let $\mathbf{s} = \mathsf{TDF.Invert}(\mathbf{T}_{1,vk_1}, \bar{\mathbf{b}_1}) \in \mathbb{Z}_q^n$.
  4. For each $i \in [k]$, run $\mathsf{TDF.Ver}_\alpha(\mathbf{A}_{i,vk_i}, \mathbf{s}, \bar{\mathbf{b}_i})$, and output $\perp$ if any execution rejects.
  5. Compute $\mathbf{h} = \bar{\mathbf{b}_0} - (\mathbf{U}^t\mathbf{s})/q \in \mathbb{T}^\ell$, and output $\mathbf{k} \in \{0,1\}^\ell$ computed as follows: for each $i \in [\ell]$, let $k_i = 0$ if h_i is closer to 0 (modulo 1) than to $1/2$, otherwise let $k_i = 1$.

4.4.2 Security Proof

Theorem 4.10. *Assuming the seu-1cma security of SS and the pseudorandomness of $A_{\mathbf{s},\Psi_\alpha}$ (on the average, for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$), the KEM described above is ind-cca-secure.*

Proof. First observe that decapsulation succeeds with overwhelming probability, by the completeness property of the chosen output-secure trapdoor functions and the usual Gaussian tail bounds applied to $\bar{\mathbf{b}_0}$.

For security, we give an extended proof sketch; full details may be extracted from [RS09]. We consider a sequence of experiments in which the adversary $\mathcal{A}$ operates.

Experiment 0: This is exactly the standard chosen-ciphertext attack as described in Definition 4.1.

Experiment 1: This is the same as the previous experiment, but $\mathcal{A}$'s oracle computing $\mathsf{Decaps}(sk, \cdot)$ is modified to output $\perp$ on *any* input of the form $(vk^*, \cdot, \cdot)$ (even if the signature verifies in Step 2), where vk^* is the first component of the challenge encapsulation $\tau^* = (vk^*, \bar{\mathbf{b}}^*, \sigma^*)$.

Experiment 2: This is the same as the previous experiment, but the $\bar{\mathbf{b}}^* \in \mathbb{T}_{q'}^{km+\ell}$ component of the challenge ciphertext $\tau^* = (vk^*, \bar{\mathbf{b}}^*, \sigma^*)$ is uniformly random and independent of all other variables.

By the seu-1cma security of SS, the distinguishing advantages of any polynomial-time $\mathcal{A}$ in experiments 0 and 1 can differ by only $\mathrm{negl}(n)$; the proof is routine.

Now we show that assuming the pseudorandomness of $A_{\mathbf{s},\Psi_\alpha}$, the distinguishing advantages of any polynomial-time $\mathcal{A}$ in experiments 1 and 2 can differ by only $\mathrm{negl}(n)$. Consider the following reduction $\mathcal{S}$, which is given access to a distribution D that is either $A_{\mathbf{s},\Psi_\alpha}$ or $U(\mathbb{Z}_q^n \times \mathbb{T})$, and interacts with $\mathcal{A}$ to simulate either experiment 1 or 2. $\mathcal{S}$ operates as follows:

1. Generate $(vk^*, sk_{\mathrm{SS}}^*) \leftarrow \mathsf{SS.Gen}()$ and choose $\mathbf{k} \in \{0,1\}^\ell$ uniformly at random.

uniformly without trapdoors. We adopt the above form merely for the sake of uniform notation.

2. Draw $km + \ell$ samples $(\mathbf{a}_i, b_i)$ from D and use them all to form the matrices $\mathbf{A}_{1,vk_1}, \ldots, \mathbf{A}_{k,vk_k}, \mathbf{U}$ and discretized vector $\bar{\mathbf{b}}^* \in \mathbb{T}_{q'}^{km+\ell}$ for the challenge encapsulation in the natural way.

3. Compute a signature $\sigma^* \leftarrow \mathsf{SS.Sign}(sk_{\mathrm{SS}}, \bar{\mathbf{b}})$ and generate the challenge $\tau^* = (vk^*, \bar{\mathbf{b}}^*, \sigma^*)$.

4. For each $i \in [k]$, generate $(\mathbf{A}_{i,1-vk_i}, \mathbf{T}_{i,1-vk_i}) \leftarrow \mathsf{TDF.Gen}()$.

5. Give the public key $pk = (\{\mathbf{A}_{i,b}\}, \mathbf{U})$, challenge τ^*, and $\mathbf{k}$ to the adversary $\mathcal{A}$.

6. Answer $\mathcal{A}$'s decryption queries as follows: given a query $\tau = (vk, \bar{\mathbf{b}}, \sigma)$, if $vk = vk^*$ then answer $\perp$. Now parse $\bar{\mathbf{b}}$ as in Step 1 of Decaps and answer $\perp$ if τ or $\bar{\mathbf{b}}$ cannot be parsed properly. Else, as in Step 2 of Decaps, run $\mathsf{SS.Ver}(vk, \bar{\mathbf{b}}, \sigma)$ and answer $\perp$ if Ver rejects. Else let i be such that $vk_i \neq vk_i^*$, and let $\mathbf{s} = \mathsf{TDF.Invert}(\mathbf{T}_{i,vk_i}, \bar{\mathbf{b}}_i)$. (Note the possibly different index $i \neq 1$ used here versus in Decaps.) Finally, execute Steps 4 and 5 just as in Decaps.

It can verified that when the distribution D provided to $\mathcal{S}$ is $A_{\mathbf{s},\Psi_\alpha}$, $\mathcal{S}$ simulates experiment 1 (up to $\mathrm{negl}(n)$ statistical distance). The main observation is that by the definition of chosen-output security, in order for all of the executions of $\mathsf{TDF.Ver}_\alpha$ in Step 4 of Decaps to accept for some $\mathbf{s} \in \mathbb{Z}_q^n$ (which must be unique if it exists), it must be the case that $\mathsf{TDF.Invert}(\mathbf{T}_{i,vk_i}, \bar{\mathbf{b}}_i)$ outputs the same $\mathbf{s}$ for all $i \in [k]$. So the result of Step 3 in Decaps and corresponding recovery of $\mathbf{s}$ by $\mathcal{S}$ are the same.

On the other hand, when the distribution D provided to $\mathcal{S}$ is uniform, $\mathcal{S}$ simulates experiment 2 (up to $\mathrm{negl}(n)$ statistical distance) because $\bar{\mathbf{b}}^*$ is uniform and independent of all other variables.

Finally, observe that in experiment 2 the challenge vector $\bar{\mathbf{b}}^*$ is uniform and independent of the key $\mathbf{k}$; therefore, the distinguishing advantage of any (even unbounded) $\mathcal{A}$ is zero. We conclude that the distinguishing advantage of $\mathcal{A}$ in the real chosen-ciphertext attack is $\mathrm{negl}(n)$, and we are done. □

Acknowledgments

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

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, 2009. To appear.

[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.

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO*, pages 278–291, 1993.

[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.

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

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[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.

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

[Gol04] Oded Goldreich. *Foundations of Cryptography*, volume II. Cambridge University Press, 2004.

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

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HWZ07] Qiong Huang, Duncan S. Wong, and Yiming Zhao. Generic transformation to strongly unforgeable signatures. In *ACNS*, pages 1–17, 2007.

[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. In *CRYPTO*, 2009. To appear.

[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.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.

[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.

[Pei09] Chris Peikert. Bonsai trees (or, arboriculture in lattice-based cryptography). Cryptology ePrint Archive, Report 2009/359, July 2009. `http://eprint.iacr.org/`.

[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.

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

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

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

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.

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