Electronic Colloquium on Computational Complexity, Report No. 100 (2008)

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

Chris Peikert[*]

November 13, 2008

Abstract

We construct public-key cryptosystems that are secure assuming the *worst-case* hardness of approximating the length of a shortest nonzero vector in an n-dimensional lattice to within a small $\mathrm{poly}(n)$ factor. 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 certain 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. In addition, we construct new cryptosystems based on the *search* version of LWE, including a very natural *chosen ciphertext-secure* system that has a much simpler description and tighter underlying worst-case approximation factor than prior constructions.

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

[*] Computer Science Lab, SRI International, Menlo Park, CA, cpeikert@alum.mit.edu. This material is based upon work supported by the National Science Foundation under Grants CNS-0716786 and CNS-0749931. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

ISSN 1433-8092

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 length (typically, in the Euclidean norm) of the shortest nonzero vector in a given lattice; another, called the *short independent vectors problem* SIVP, is (essentially) to find a full-rank set of lattice vectors that are relatively short.

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

A different class of cryptosystems (and the only others known to enjoy worst-case hardness) stem from a work of Regev [Reg05], who defined a 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, where $\mathbb{Z}_q$ is represented by the integer residues $\lceil -\frac{q}{2} \rceil, \ldots, \lfloor \frac{q-1}{2} \rfloor$. 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 *uniform* samples over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. It turns out that when the modulus q is prime and polynomial in n, the search and decision variants are *equivalent* via an elementary reduction (but no such equivalence is known for larger q).

The LWE problem has turned out to be amazingly versatile. In addition to its first application in a public-key cryptosystem [Reg05], it has provided the foundation for chosen ciphertext-secure cryptosystems [PW08], identity-based encryption [GPV08], and universally composable oblivious transfer [PVW08], as well as for strong hardness of learning results relating to halfspaces [KS06]. We emphasize that all of the above cryptographic applications are based on the *decision* version of LWE.

The main technical 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 the problems GapSVP and SIVP on n-dimensional lattices, in the worst case. 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 result, 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 then 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 yet 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. A 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 on general 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 the *search* version of LWE, for any sufficiently large modulus $q \geq 2^n$, is at least as hard as approximating GapSVP in the worst case, via a classical (probabilistic polynomial-time) reduction. The concrete approximation factor for GapSVP has essentially the same dependence on the error distribution as in the quantum reduction of [Reg05].

- Our main reduction additionally shows that for moduli as small as $q \geq \omega(\sqrt{n})$, the search version of LWE is at least as hard as (classically) approximating a *novel variant* of the shortest vector problem on general lattices in the worst case. The new problem is essentially the GapSVP problem on "higher quality" representations of the input lattice; hence, it is no harder than standard GapSVP, yet it still appears to be exponentially hard given the state of the art in lattice algorithms [AKS01].

  By the above-mentioned equivalence between search- and decision-LWE for prime $q = \mathrm{poly}(n)$, our result provides a classical (but incomparable) foundation for the hardness of decision-LWE and the many cryptographic applications that are based upon it.

- We construct new cryptosystems based on the *search* version of LWE (for any modulus q), including a simple and natural cryptosystem that is secure under *chosen-ciphertext attack*.

  In our basic (semantically secure) system, public keys are of size $O(n^2 \log^2 q)$, and the expansion factor of an n-bit plaintext can be as small as $O(\log q)$. (The chosen ciphertext-secure cryptosystem just incurs additional n^δ factors.) The underlying worst-case approximation factor for GapSVP (or its new variant) is $\tilde{O}(n^2 \log q)$, and has the potential to be reduced to $\tilde{O}(n^{1.5}\sqrt{\log q})$ with an improved key-generation algorithm.[1]

  Assuming hardness of the standard GapSVP problem (and letting $q = 2^{O(n)}$), the public key size and ciphertext expansion factor are therefore $O(n^4)$ and $O(n)$, respectively; these quantities match the (amortized) Ajtai-Dwork cryptosystem based on unique-SVP [AD07].

  Assuming hardness of the new GapSVP variant (and letting $q = \mathrm{poly}(n)$), the public key size and ciphertext expansion can be as small as $O(n^2)$ and $O(\log n)$, respectively; these match the most efficient known cryptosystems based on *decision*-LWE [PVW08, GPV08].

Our chosen ciphertext-secure cryptosystem provides an alternative to a recent construction of Peikert and Waters [PW08] based on the decision-LWE problem. In addition to the new system's classical worst-case foundation, other key advantages include its tighter underlying approximation factor and its relatively simple description and analysis (the construction in [PW08] is somewhat cumbersome in both regards).

[1] The $\tilde{O}(\cdot)$ notation hides factors that are polynomial in $\log n$.

1.2 Overview

1.2.1 Conceptual Summary

We start by giving a high-level description of the common design and analysis paradigm of prior cryptosystems with worst-case connections [AD97, Reg04b, Reg05, AD07]. These works consider two types of probability distributions over some additive domain: one is the uniform distribution, while the other type consists of "lumpy" distributions that are *periodic* and *concentrated* around multiples of the period. As a simple example, in [Reg04b] the domain is the real interval $[0, 1)$ with addition modulo 1, and lumpy distributions are concentrated around integer multiples of $1/h$ for some large integer h.

The cryptosystems are constructed roughly as follows: the secret key is a period chosen at random, and the public key consists of several samples from the corresponding lumpy distribution. A 0 bit is encrypted by letting the ciphertext be a *random subset-sum* of the samples in the public key; a 1 is encrypted by choosing a *uniformly random* value in the domain (other slight variations are also possible). Decryption simply tests whether the ciphertext is "relatively close" to a multiple of the secret period (to decrypt as 0) or not (to decrypt as 1).

Semantic security is proved by a thought experiment in which the public key is instead made up of samples drawn from the *uniform* distribution. It so happens that encrypting under such a key hides the message bit *statistically* (i.e., information-theoretically), because random subset sums are distributed almost uniformly. It follows that an adversary capable of breaking the semantic security of the cryptosystem can likewise distinguish between the uniform and lumpy distributions.

Finally, the core technical component is a reduction demonstrating that the two kinds of distributions are *computationally indistinguishable*, assuming the worst-case hardness of some lattice problem. Essentially, the reduction takes a lattice as input and produces samples from one of the two kinds of distributions, depending on the geometric properties of the lattice. Crucially, in order to guarantee that the reduction produces samples from the specific kinds of *structured* lumpy distributions that are used in the cryptosystem, it has so far been necessary to impose additional geometric constraints on the reduction's input. This is why prior works have relied on specialized assumptions, e.g., relating to unique-SVP.

Our Approach. We retain the use of uniform and (a certain kind of) lumpy distributions, and give a reduction that samples from one of the two types. Our cryptosystems, on the other hand, depart substantially from the previous design and analysis paradigm: public keys in our systems are instead drawn from the *uniform* distribution, whereas lumpy distributions are used only in the *security proof* to show statistical hiding. The principal advantage of this approach is that it *significantly relaxes the structural properties* required of the lumpy distributions: first, because they no longer need to support decryption, and more importantly, because they never need to be sampled in the "real world" at all! This makes additional geometric constraints on the reduction's input unnecessary, and allows for a security proof under worst-case assumptions on general lattices.

Several natural questions immediately arise about this approach, such as: What is the supporting secret key for a uniformly-distributed public key? How does one encrypt and decrypt? And how do the lumpy distributions induce statistically secure encryption? We address these issues in the following more technical overview.

1.2.2 New Cryptosystems

Here we describe new cryptosystems based on the search-LWE problem. At their heart is a certain collection of injective (i.e., one-to-one) trapdoor functions. This collection appeared in a recent work of Gentry, Peikert, and Vaikuntanathan [GPV08], and is closely related to an earlier proposal by Goldreich, Goldwasser, and Halevi [GGH97]. In this work, we prove that the collection is one-way under classical worst-case assumptions, and we establish additional properties that are useful in constructing cryptosystems.

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 columns $\mathbf{a}_i \in \mathbb{Z}_q^n$, for some large enough m. A random input to $g_{\mathbf{A}}$ 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 in the output vector $\mathbf{b}$, each entry $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i$, so inverting the function is syntactically identical to solving search-LWE given m noisy inner products (note that $\mathbf{x}$ is easily computed once $\mathbf{s}$ is known, and vice versa). Moreover, if $g_{\mathbf{A}}$ is one-way, then there is a generic hard-core predicate $h(\mathbf{s})$ for $g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$ [GL89].

As shown in [GPV08], the function $g_{\mathbf{A}}$ has a *trapdoor* that enables efficient recovery of the input $\mathbf{s}$ from $\mathbf{b}$, so long as the error distribution χ is sufficiently concentrated. Concretely, the trapdoor $\mathbf{T}$ is a "good" basis for a certain lattice defined by $\mathbf{A}$, which can be generated together with an $\mathbf{A}$ having the desired (almost-)uniform distribution [Ajt99, AP08]. The inversion algorithm uses the trapdoor basis $\mathbf{T}$ in a simple rounding algorithm to recover $\mathbf{s}$.

Using this collection of trapdoor functions, it is straightforward to construct a basic semantically secure cryptosystem. The secret and public keys are $\mathbf{T}$ and $\mathbf{A}$ (respectively), as above. An encryption of a message bit μ consists of $\mathbf{b} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$ for random $\mathbf{s}$ and $\mathbf{x}$ as above, as well as $\mu \oplus h(\mathbf{s})$. The decryption algorithm uses the trapdoor $\mathbf{T}$ to recover $\mathbf{s}$ from $\mathbf{b}$, recomputes the predicate $h(\mathbf{s})$, and recovers the message μ.

Improved efficiency and chosen-ciphertext security. One of our technical results is that the function $g_{\mathbf{A}}$ actually admits a very simple hard-core predicate, namely, the *parity* of any coordinate $s_i \in \mathbb{Z}_q$ of $\mathbf{s}$ (when q is even). Moreover, we show how to extend this hard bit into ℓ *simultaneously* hard bits, by lifting the LWE problem from dimension n to $n + \ell - 1$ via an elementary reduction. This results in an "amortized" cryptosystem that can encrypt messages of length, say, $\ell = n$ bits using public keys and ciphertexts that are only a constant factor larger than in the basic system. (Similar amortization techniques for other lattice-based cryptosystems were also recently proposed in [PVW08, AD07].) As a further optimization, we also show that the output of $g_{\mathbf{A}}$ can be represented in a "coarser" group $\mathbb{Z}_{q'}^m$ for some modulus $q' = \mathrm{poly}(n)$, which reduces the ciphertext size by an almost-linear factor in n when q is large (e.g., $q = 2^n$).

To construct cryptosystems that are secure under chosen-ciphertext attacks, we rely on a recent approach of [PW08] and additional perspectives of Rosen and Segev [RS08]. The key observation is that k independently chosen functions $g_{\mathbf{A}_1}, g_{\mathbf{A}_2}, \ldots, g_{\mathbf{A}_k}$ remain one-way even when evaluated on the *same* input $\mathbf{s}$ (but independent error vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$), assuming the hardness of search-LWE given $k \cdot m$ samples. (This fact was also observed independently by Goldwasser and Vaikuntanathan [GV08], for the same purpose.) For injective trapdoor functions, one-wayness under such "correlated inputs" immediately yields chosen-ciphertext security, as shown in [RS08]. At the same time, our proof of one-wayness under correlation follows by showing that the functions have "lossy" counterparts *a la* [PW08], as we now explain.

1.2.3 Classical Hardness of LWE

Here we give a simplified description of our worst-case GapSVP to LWE reduction, which conveys all the essential ideas (we refer the reader to Section 3 for full details). The input to the reduction is some arbitrary n-dimensional lattice Λ (represented by a basis), and the goal is to approximate GapSVP given access to an oracle that solves the search-LWE problem on m samples. That is, the reduction should determine if the minimum distance of Λ (i.e., the length of its shortest nonzero vector) is "small" or "large," where these quantities are separated by some $\text{poly}(n)$ multiplicative gap (and in between, any answer is acceptable).

Abstractly, the reduction first invokes a certain sampling procedure over the *dual lattice* Λ^* to generate independent $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{Z}_q^n$ according to some (unknown) distribution. Concretely, the procedure samples vectors $\mathbf{y}_i \in \Lambda^*$ from a *Gaussian-like* distribution (as first used in [Reg04b], and refined in subsequent works [MR07, Reg05, GPV08]), and lets $\mathbf{a}_i$ identify the residue class of $(\Lambda^*/q\Lambda^*) \equiv \mathbb{Z}_q^n$ containing $\mathbf{y}_i$. The reduction then chooses a random secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error terms x_i from χ, and gives the noisy inner products $(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + x_i)$ to the LWE oracle. If the oracle correctly produces $\mathbf{s}$ as its solution, the reduction outputs "large," otherwise it outputs "small."

When the minimum distance of Λ is large, the $\mathbf{a}_i$ are distributed essentially *uniformly* over $\mathbb{Z}_q^n$; this follows by a bound on the *smoothing parameter* of Λ^* due to Micciancio and Regev [MR07]. Therefore, the input provided to the oracle is faithful to the LWE distribution, the oracle solves for $\mathbf{s}$ by hypothesis, and the reduction outputs "large" as desired.

The case of small minimum distance is more interesting, and constitutes the chief novelty of our approach and analysis. In this case, the distribution of the $\mathbf{a}_i$ is *lumpy*, in the following sense: there is some (unknown) nonzero $\mathbf{s}' \in \mathbb{Z}_q^n$ such that the distribution of $\langle \mathbf{a}_i, \mathbf{s}' \rangle \bmod q$ is relatively concentrated around 0. (Concretely, $\mathbf{s}'$ is the coefficient vector, reduced modulo q, of a short vector in Λ). For a sufficiently wide error distribution χ over $\mathbb{Z}_q$, the noisy inner products then *statistically hide* the reduction's choice of $\mathbf{s}$, i.e., it is about as likely to be $\mathbf{s} + \mathbf{s}'$, conditioned on the view of the oracle. The oracle must therefore guess incorrectly with noticeable probability, and the reduction outputs "small" as desired. (Using a more technical argument, we also show that a particular *predicate* on $\mathbf{s}$ is essentially uniform, hence hard-core, conditioned on the view.)

Additional details. The modulus q must be large enough so that in the lumpy case, the distribution of $\langle \mathbf{a}_i, \mathbf{s} \rangle$ is well-concentrated relative to the size of q. The degree of concentration is dictated by the tightness of the reduction's main sampling algorithm, which in turn is governed by the "quality" of the input basis. Using an LLL-reduced basis [LLL82] (which may be computed in polynomial time), the value $q = 2^n$ suffices. However, if the reduction is given a basis of better quality, then a smaller q may be used; this is where the new variant of GapSVP comes into play.

The reduction we have outlined above, while technically correct, is still not quite as strong as we would like. This is because in the lumpy case, the amount of noise required to hide $\mathbf{s}$ *grows* with the number of samples m that the oracle uses, whereas ideally it should be independent of m. This is important for optimizing the underlying worst-case approximation factors (especially for chosen-ciphertext security, which uses more samples), and is also needed for the LWE search/decision equivalence for prime $q = \text{poly}(n)$.

To address this issue, our reduction actually generates each pair $(\mathbf{a}_i, b_i)$ *together at once* for any desired number of samples, by adding noise *a priori* to a known vector $\mathbf{v} \in \Lambda$ in the input lattice, rather than *a posteriori* to the inner products $\langle \mathbf{a}_i, \mathbf{s} \rangle$. When the minimum distance is large, the LWE oracle can be used to recover $\mathbf{v}$, whereas when the minimum distance is small, $\mathbf{v}$ is statistically hidden. In the end, our reduction relies heavily upon the classical component of Regev's reduction [Reg05], though in our case the $\mathbf{a}_i$ are generated by a classical sampling algorithm of [GPV08] rather than by a quantum step, and we follow a different approach for solving GapSVP.

1.3 Discussion and Open Problems

Note that the (simplified) reduction above essentially chooses a function $g_{\mathbf{A}}$ under an unknown distribution on $\mathbf{A}$, evaluates it on a known input, and checks whether the oracle recovers that same input. The uniform distribution on $\mathbf{A}$ induces an injective (trapdoor) function, whereas a lumpy distribution induces a function that statistically hides its input. This is essentially the notion of a *lossy trapdoor function* from [PW08], but in a slightly relaxed sense: in our case, there is no *single, fully-specified* (and efficiently-sampleable) distribution that induces a lossy function — but *any* lattice with small minimum distance does so.

It is worth pointing out explicitly how our reduction avoids quantum computation. Recall that the LWE oracle solves for a secret $\mathbf{s}$ (alternately, a vector $\mathbf{v}$ in the input lattice) that the reduction chooses itself. In [Reg05], this allowed the quantum part of the reduction to "uncompute" $\mathbf{s}$ and create a useful quantum state, but it was unclear whether such an oracle was of any use classically. Here we avoid quantum computation by introducing, as a complementary case, a lattice with small minimum distance that statistically hides the reduction's random choices. In this case, the inputs provided to the oracle (in particular, the $\mathbf{a}_i$s) are *not* faithful to the LWE distribution, but this is of absolutely no consequence! We mention that related forms of statistical hiding via small minimum distance have also appeared in the context of interactive proofs for lattice problems [GG00, MV03] and algorithms for the shortest vector problem [AKS01].

Currently, our core reductions are *non-adaptive* (all queries to the LWE oracle can be prepared in advance), and seem to be limited to solving the *decision* version GapSVP of the shortest vector problem. It would be very interesting if the reductions could be made "iterative" and/or extended to solve *search* problems such as SIVP, like the quantum reduction of [Reg05] and prior reductions for "Minicrypt" primitives (e.g., [Ajt04, MR07]). Another open problem is to design a reduction that solves the *search* version of the shortest vector problem; such a result would be quite surprising, because even the prior reductions mentioned above have also been limited to the decision version.

Finally, we believe that it may be very fruitful to study the complexity of our new variant of GapSVP (and related lattice problems), in which a gap of intermediate quality is already promised and a tighter approximation is desired.

2 Preliminaries

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

The main security parameter throughout the paper is n, and all other quantities are implicitly functions of n. We use standard $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, and $\omega(\cdot)$ notation to describe the growth of functions, and write $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n) \cdot \log^c n)$ for some fixed constant c. We let $\mathrm{poly}(n)$ 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. For a set $S \subseteq \mathbb{R}^n$, point $\mathbf{x} \in \mathbb{R}^n$, and scalar $c \in \mathbb{R}$, we define $S + \mathbf{x} = \{\mathbf{y} + \mathbf{x} \, : \, \mathbf{y} \in S\}$ and $cS = \{c\mathbf{y} \, : \, \mathbf{y} \in S\}$.

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

For any (ordered) set $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathbb{R}^n$ of linearly independent vectors, let $\tilde{\mathbf{S}} = \{\tilde{\mathbf{s}_1}, \ldots, \tilde{\mathbf{s}_n}\}$ denote its *Gram-Schmidt orthogonalization*, defined iteratively as follows: let $\tilde{\mathbf{s}_1} = \mathbf{s}_1$, and for each $i = 2, \ldots, n$, let $\tilde{\mathbf{s}_i}$ be the projection of $\mathbf{s}_i$ onto $\operatorname{span}^{\perp}(\mathbf{s}_1, \ldots, \mathbf{s}_{i-1})$, i.e., $\tilde{\mathbf{s}_i} = \mathbf{s}_i - \sum_{j=1}^{i-1} \mu_{i,j} \tilde{\mathbf{s}_j}$, where $\mu_{i,j} = \langle \mathbf{s}_i, \tilde{\mathbf{s}_j} \rangle / \langle \tilde{\mathbf{s}_j}, \tilde{\mathbf{s}_j} \rangle$. Observe that $\|\tilde{\mathbf{s}_i}\| \leq \|\mathbf{s}_i\|$ for all i.

Probability. For a probability distribution X over a domain D, let $f_X : D \to \mathbb{R}$ denote its density function. Let X^n denote the n-fold product distribution over D^n, which has density function $f_{X^n}(\mathbf{x}) = f_X^n(\mathbf{x}) := f_X(x_1) \cdots f_X(x_n)$. 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 $U(D)$.

Let X and Y be two distributions, and let D be a probabilistic algorithm. We say that the *advantage* of D in distinguishing X from Y is $|\Pr[D(X) = 1] - \Pr[D(Y) = 1]|$. We say that two ensembles $\{X_n\}$ and $\{Y_n\}$ of distributions indexed by n are *computationally indistinguishable* if every probabilistic polynomial-time D has negligible advantage $\operatorname{negl}(n)$ in distinguishing X_n from Y_n.

For any $r > 0$, define the one-dimensional Gaussian function $\rho_r : \mathbb{R} \to \mathbb{R}$ with parameter r as

$$\rho_r(x) = \exp(-\pi(x/r)^2).$$

(We take $r = 1$ when it is omitted.) The total measure associated to ρ_r is $\int_{\mathbb{R}^n} \rho_r(x)\, dx = r$, so we can define a continuous Gaussian probability distribution over $\mathbb{R}$ by its density function $D_r(x) = \rho_r(x)/r$ (as before, we may omit r). These extend to $\mathbb{R}^n$ in the usual way as $\rho_r^n(\mathbf{x}) = \rho_r(x_1) \cdots \rho_r(x_n) = \exp(-\pi(\|\mathbf{x}\|/r)^2)$ and $D_r(\mathbf{x}) = \rho_r(\mathbf{x})/r^n$. We also define the Gaussian norm distribution $S_r^{(n)}$, which is obtained by sampling a vector $\mathbf{x} \in \mathbb{R}^n$ from D_r^n and outputting $\|\mathbf{x}\|$.

The Gaussian distribution D_r^n is spherically symmetric, so for $\mathbf{x}$ distributed according to D_r^n and any unit vector $\mathbf{u} \in \mathbb{R}^n$, $\langle \mathbf{u}, \mathbf{x} \rangle$ is distributed according to D_r. For $x \in \mathbb{R}$ distributed according to D_r and any $t \geq 1$, a standard tail inequality says that $|x| < r \cdot t$ except with probability at most $\exp(-\pi t^2)$. In addition, for $\mathbf{x} \in \mathbb{R}^n$ distributed according to D_r^n, we have $\|\mathbf{x}\| < r\sqrt{n}$ except with probability at most 2^{-n}.

It is possible to sample efficiently from D_r (hence D_r^n) to within any desired level of precision. It is possible to sample efficiently from $U(\mathcal{B}_n)$ by first choosing an $\mathbf{x}$ according to D^n to select a random direction, then scaling $\mathbf{x}$ to have (Euclidean) norm $r \in [0, 1)$ with probability proportional to r^{n-1}. For simplicity, we use real numbers in this work and assume that we can sample from D_r^n exactly; all the arguments can be made rigorous by using a suitable amount of precision.

To prove the hardness of search-LWE, we need the following lemma about the statistical distance between the uniform distributions over two n-dimensional balls whose centers are relatively close.

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

2.1 Learning with Errors

Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ be the additive group on the real interval $[0, 1)$ with modulo 1 addition. For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution ϕ on $\mathbb{T}$, define $A_{\mathbf{s},\phi}$ to be the distribution on

$\mathbb{Z}_q^n \times \mathbb{T}$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing an error term $e \in \mathbb{T}$ according to ϕ, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle / q + e)$, where the addition is performed in $\mathbb{T}$.

We are primarily concerned with error distributions ϕ over $\mathbb{T}$ that are derived from Gaussians. For $\alpha > 0$, define Ψ_α to be the distribution on $\mathbb{T}$ obtained by taking a sample from the one-dimensional Gaussian $D_\alpha^{(1)}$ and reducing modulo 1. At times we consider an error distribution ϕ that is *itself* a random variable, e.g., Ψ_β for β chosen according to some distribution. We point out such cases explicitly when they arise, but retain the same notation as when ϕ is a fixed distribution.

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

The above definition of LWE is for a "worst-case" search problem. As shown in [Reg05], it is equivalent (up to a polynomial factor in the number of samples used) to an "average-case" version in which the goal is to find a *uniformly random* $\mathbf{s} \in \mathbb{Z}_q^n$ with *non-negligible* probability given $A_{\mathbf{s},\phi}$ (where the probability is taken over all the randomness in the experiment). This equivalence follows by a simple reduction from arbitrary $\mathbf{s}$ to uniformly random $\mathbf{s}' \in \mathbb{Z}_q^n$ [Reg05, Lemma 4.1], and the ability to verify a correct value of $\mathbf{s}'$ once it is found [Reg05, Lemma 3.6]). Specifically, suppose W is an oracle that solves the average-case version of LWE. To find an arbitrary $\mathbf{s}$ with overwhelming probability given $A_{\mathbf{s},\phi}$, we transform it into $A_{\mathbf{s}',\phi}$ for a uniformly random $\mathbf{s}' = \mathbf{s} + \mathbf{t}$ by choosing random $\mathbf{t} \in \mathbb{Z}_q^n$ and mapping pairs $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle / q)$. By invoking W, we obtain a candidate solution $\tilde{\mathbf{s}}$, check whether $\tilde{\mathbf{s}} = \mathbf{s}'$, and output $\mathbf{s} = \tilde{\mathbf{s}} - \mathbf{t}$ if so. By repeating a polynomial number of times, we find $\mathbf{s}$ with overwhelming probability.

For a function $\pi : \mathbb{Z}_q^n \to \{0,1\}^\ell$, we say that π is *hard-core* for $\mathsf{LWE}_{q,\phi}$ (in n dimensions) if, given access to $A_{\mathbf{s},\phi}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$, $\pi(\mathbf{s})$ is computationally indistinguishable from $U(\{0,1\}^\ell)$. When $\ell = 1$, this is equivalent (via standard reductions) to saying that no probabilistic polynomial-time algorithm computes $\pi(\mathbf{s})$ with probability better than $1/2 + \operatorname{negl}(n)$. We are interested in a particular candidate collection of hard-core functions for LWE. For *even* q and any $\ell \geq 1$, define

$$h^\ell : \mathbb{Z}_q^{\geq \ell} \to \{0,1\}^\ell \qquad \text{as} \qquad h^\ell(\mathbf{s}) = h(s_1) \circ \cdots \circ h(s_\ell),$$

where $h(s)$ for $s = \bar{s} + q\mathbb{Z} \in \mathbb{Z}_q$ denotes the *parity* of the integer residue $\bar{s} \in \mathbb{Z}$, and $\circ$ denotes concatenation. (Note that because q is even, any choice of residue $\bar{s}$ for s has the same parity.)

2.2 Lattices

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

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

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

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

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, it can be seen that the dual basis $\mathbf{B}^* = (\mathbf{B}^{-1})^t$ is in fact a basis of Λ^*. The following standard fact relates the Gram-Schmidt orthogonalizations of a basis and its dual (a proof can be found in [Reg04a, Lecture 8]).

Lemma 2.3. *Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be an (ordered) basis, and let $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ be its dual basis in reversed order (i.e., $\mathbf{d}_i = \mathbf{b}^*_{n-i+1}$). Then $\tilde{\mathbf{d}_i} = \tilde{\mathbf{b}_i}/\|\tilde{\mathbf{b}_i}\|^2$ for all $i \in [n]$. In particular, $\|\tilde{\mathbf{d}_i}\| = 1/\|\tilde{\mathbf{b}_i}\|$.*

Computational problems. We are mainly interested in the shortest vector problem on lattices.

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

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

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

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

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

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

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

The first condition is the interesting one. For any $\zeta(n) \geq 2^{(n-1)/2}$, $\mathsf{GapSVP}_{\zeta,\gamma}$ is actually *equivalent* to the standard GapSVP_γ problem, because an arbitrary basis $\mathbf{B}'$ of Λ can be reduced in polynomial time using the LLL algorithm [LLL82] to another basis $\mathbf{B}$ of Λ so that $\lambda_1(\Lambda) \leq \|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \min_i \|\tilde{\mathbf{b}_i}\|$. (In fact, alternate parameters and analysis of the LLL algorithm imply that we can even take $\zeta(n) \approx (2/\sqrt{3})^n$.) For smaller functions $\zeta(n)$, particularly $\zeta(n) = \mathrm{poly}(n)$, the condition is nontrivial and more interesting. The nature of the problem is to approximate the minimum distance to within a gap $\gamma(n)$, given a promise that it lies within a looser range having a gap $\zeta(n)$. The promise could be made efficiently verifiable by restricting to "high quality" bases that contain (or guarantee the existence of) a vector of length at most $\zeta(n)$, though this could potentially make the problem easier. To our knowledge, none of the lattice algorithms in the literature are able to solve $\mathsf{GapSVP}_{\zeta,\gamma}$ for $\gamma(n) < \zeta(n) = \mathrm{poly}(n)$ in time better than exponential $2^{\Omega(n)}$, even when the promise is verifiable efficiently, and even when, say, $\zeta(n) = 2\gamma(n)$.

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

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

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

For an n-dimensional lattice Λ, real $r > 0$, and $\mathbf{c} \in \mathbb{R}^n$, define the *discrete Gaussian probability distribution over* Λ (with parameter r, centered at $\mathbf{c}$) as:

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

(As above, r and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$, respectively, when omitted.) Note that the denominator in the above expression is merely a normalization factor.

Our reductions use, as a subroutine, an efficient algorithm that generates samples from discrete Gaussian distributions.

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

To demonstrate a particular hard-core predicate for LWE, we also need the following simple (but new, to our knowledge) fact about discrete Gaussians.

Lemma 2.9. *Let $\mathbf{B}$ be a basis of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, and let $\mathbf{v} = \mathbf{B}\mathbf{z} \in \Lambda$ be a nonzero lattice vector whose ith coefficient z_i is* odd*. Let $r \geq \|\mathbf{v}\| \cdot \omega(\sqrt{\log n})$, let $\mathbf{c} \in \mathbb{R}^n$ be arbitrary, and let $\mathbf{x} = \mathbf{B}\mathbf{z}'$ be a random variable having distribution $D_{\Lambda,r,\mathbf{c}}$. Then the* parity *of coefficient z'_i (i.e., $z'_i \bmod 2$) is negligibly close to uniform over $\{0,1\}$.*

We remark that the lemma easily generalizes to any prime modulus p, where for $z_i \neq 0 \bmod p$ and $r \geq p \cdot \|\mathbf{v}\| \cdot \omega(\sqrt{\log n})$, we have that $z'_i \bmod p$ is negligibly close to uniform over $\mathbb{Z}_p$.

Proof. Define a basis $\mathbf{B}'$ of a sublattice $\Lambda' = \mathcal{L}(\mathbf{B}') \subset \Lambda$ as $\mathbf{b}'_i = 2\mathbf{b}_i$ and $\mathbf{b}'_j = \mathbf{b}_j$ for all $j \neq i$. Then we have $\mathbf{v} = \mathbf{B}\mathbf{z} \notin \Lambda'$, and $\Lambda = \Lambda' \cup (\Lambda' + \mathbf{v})$. Observe that for $\mathbf{x} = \mathbf{B}\mathbf{z}' \in \Lambda$, the parity of z'_i is zero if $\mathbf{x} \in \Lambda'$, and is one if $\mathbf{x} \in \Lambda' + \mathbf{v}$.

For $\mathbf{x}$ distributed according to $D_{\Lambda,r,\mathbf{c}}$, the probability that z'_i is even or odd is therefore proportional to $P_0 = \rho_r(\Lambda' - \mathbf{c})$ or $P_1 = \rho_r(\Lambda' + \mathbf{v} - \mathbf{c})$, respectively. A routine argument (using techniques from [MR07]) shows that for $r \geq \|\mathbf{v}\| \cdot \omega(\sqrt{\log n})$, the quantities P_0 and P_1 are within a $(1 \pm \operatorname{negl}(n))$ factor of each other, which proves the claim. We defer a complete proof to the full version. □

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. In Section 3.1 we give a reduction establishing the hardness of LWE in its search version. This proves that the injective trapdoor functions from [GPV08] are indeed one-way, hence have a generic hard-core predicate that can be used to encrypt a single bit at a time. In Section 3.2 we give a more technical proof showing that LWE admits a *specific* natural hard-core predicate, which has the advantage that it can be easily extended into many *simultaneously* hard bits (as shown in Section 4.1); this leads to more efficient multi-bit cryptosystems.

3.1 Hardness of Search-LWE

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

There is a (classical) probabilistic polynomial-time reduction from solving $\mathsf{GapSVP}_{\zeta,\gamma}$ *in the worst case (with overwhelming probability) to solving* $\mathsf{LWE}_{q,\Psi_\alpha}$ *with non-negligible probability (for uniformly random* $\mathbf{s} \in \mathbb{Z}_q^n$*) using a polynomial number of samples.*

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

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

We also mention that using results from [Pei08], Theorem 3.1 can easily be generalized to work for $\mathsf{GapSVP}_{\zeta,\gamma}$ in any ℓ_p norm, $2 \le p \le \infty$, for essentially the same approximation factor γ.

Our proof of Theorem 3.1 relies on the core classical component of Regev's reduction.

Proposition 3.2 ([Reg05, Lemma 3.4]). *Let* $\epsilon = \epsilon(n)$ *be a negligible function,* $q = q(n) \ge 2$ *be an integer,* $\alpha = \alpha(n) \in (0,1)$ *and* $\phi = \Psi_\alpha$*, and* Λ *be any* n*-dimensional lattice. There is a classical probabilistic polynomial-time reduction* R *that solves* $\mathsf{CVP}_{\alpha q/(\sqrt{2}r)}$ *on* Λ *in the worst case (with overwhelming probability), given:*

1. *an oracle* W *that solves* $\mathsf{LWE}_{q,\phi}$ *with non-negligible probability (for uniformly random* $\mathbf{s} \in \mathbb{Z}_q^n$*) using a polynomial number of samples, and*
2. *an oracle that samples from* $D_{\Lambda^*,r}$ *for a given number* $r \ge \sqrt{2}q \cdot \eta_\epsilon(\Lambda^*)$.

For completeness, we give a brief description of the reduction claimed in Proposition 3.2 (however, this is not required to understand the proof of Theorem 3.1 and may be safely skipped). It is given a basis $\mathbf{B}$ of Λ and a point $\mathbf{x} \in \mathbb{R}^n$ within distance $\alpha q/(\sqrt{2}r)$ of some vector $\mathbf{v} \in \Lambda$. Suppose $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$ is the coefficient vector of $\mathbf{v}$ reduced modulo q. To generate a sample from $A_{\mathbf{s},\phi}$, the reduction obtains a sample $\mathbf{y}$ from $D_{\Lambda^*,r}$, lets $\mathbf{a} = (\mathbf{B}^*)^{-1}\mathbf{y} = \mathbf{B}^t\mathbf{y} \bmod q$, and outputs

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

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

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

The oracle W solves for $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$ by hypothesis, and the entire vector $\mathbf{v}$ can be obtained by iterating the procedure as described in [Reg05, Lemma 3.5].

We stress that the precise error distribution in the $\langle \mathbf{y}, \mathbf{x} \rangle$ term requires some care to analyze precisely; the exact distance between $\mathbf{x}$ and $\mathbf{v}$ and the extra error term e both play an important role. The details are not relevant at this point, though they will be more important later on in Section 3.2 when we analyze specific hard-core predicates.

Proving the theorem. We are now ready to prove Theorem 3.1. Essentially, the reduction works as follows: given a lattice Λ, it perturbs a point $\mathbf{v} \in \Lambda$, invokes the reduction R from Proposition 3.2 on the perturbed point, and checks whether R successfully recovers $\mathbf{v}$. When $\lambda_1(\Lambda)$ is large, R must indeed recover $\mathbf{v}$ by hypothesis. When $\lambda_1(\Lambda)$ is small, $\mathbf{v}$ is *statistically hidden* and R must guess incorrectly with some non-negligible probability. (The same basic principle underlies the interactive proofs of Goldreich and Goldwasser [GG00], where here the reduction R is playing the role of the unbounded prover.)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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. □

3.2 Hard-Core Predicate

Here we demonstrate a particular hard-core predicate for LWE (assuming the worst-case hardness of GapSVP), namely, the *parity* of the first entry s_1 of the secret $\mathbf{s} \in \mathbb{Z}_q^n$. (By symmetry, it follows that the parity of *any* single entry s_i is hard-core).

Our strategy is similar to the one used above in the proof of Theorem 3.1, but is more technically involved. Given a lattice Λ, the reduction perturbs a point $\mathbf{v} \in \Lambda$ (this time using a sufficiently wide Gaussian), uses the perturbed point to simulate an LWE distribution to an oracle P that predicts the predicate, and checks whether P's output matches a corresponding predicate on $\mathbf{v}$. When $\lambda_1(\Lambda)$ is large, the simulation is faithful to an LWE distribution and P's prediction is correct (with non-negligible advantage over $1/2$) by hypothesis. When $\lambda_1(\Lambda)$ is small, the predicate on $\mathbf{v}$ is (almost) uniform conditioned on P's input, hence P has essentially no advantage over $1/2$.

For technical reasons, we need to impose two extra conditions on the $\mathsf{LWE}_{q,\phi}$ problem in order to make the proof work. The first is that q must be an *even* integer; otherwise, the notion of parity in $\mathbb{Z}_q$ is ill-defined. The second is that the noise distribution $\phi = \Psi_\beta$ is *itself* is a random variable; more precisely, the parameter β is chosen from a certain distribution and kept secret (and fixed). This condition is an artifact of the main proof technique in the context of hard-core predicates; we elaborate below.

When reducing to the *search* problem $\mathsf{LWE}_{q,\Psi_\alpha}$, the main step in the reduction from Proposition 3.2 above actually generates samples from a distribution $A_{\mathbf{s},\Psi_\beta}$ for some *unknown* $\beta \leq \alpha$. The reduction then emulates $A_{\mathbf{s},\Psi_{\beta'}}$ for many different values of $\beta' \geq \beta$ by adding different amounts of extra noise to $A_{\mathbf{s},\Psi_\beta}$. In at least one of these instances, β' is sufficiently close to α that the oracle for $\mathsf{LWE}_{q,\Psi_\alpha}$ is obliged to return the correct solution $\mathbf{s}$. Because candidate solutions to the LWE problem can be checked efficiently, the reduction can therefore recognize the correct $\mathbf{s}$ and continue on.

When attempting to prove that a predicate π is *hard-core* for $\mathsf{LWE}_{q,\Psi_\alpha}$, however, this kind of strategy breaks down. Here we have an oracle that predicts $\pi(\mathbf{s})$ given $A_{\mathbf{s},\Psi_\alpha}$, but it appears that the correct value of $\pi(\mathbf{s})$ *cannot* be recognized efficiently on its own. So even though the reduction may emulate different instances of $A_{\mathbf{s},\Psi_{\beta'}}$, it has no way of checking *which* of the oracle's predictions is correct (and the oracle may intentionally give bad predictions under noise distributions other than Ψ_α). Our solution to this difficulty is to strengthen the hypothesis by requiring the oracle to predict $\pi(\mathbf{s})$ under error distribution Ψ_β, where β itself is a random variable that emerges from the main reduction technique. The distribution of β is somewhat unnatural, but presents no problems in usage.

Theorem 3.3. *Let $\alpha = \alpha(n) \in (0,1)$ be a real number and $\gamma = \gamma(n) \geq \omega(n\sqrt{\log n}/\alpha)$. Let $\zeta = \zeta(n) \geq \gamma$ and $q = q(n) \geq (\zeta/\sqrt{n}) \cdot \omega(\sqrt{\log n})$ be an* even *integer.*

There is a classical probabilistic polynomial-time reduction from solving $\mathsf{GapSVP}_{\zeta,\gamma}$ in the worst case (with overwhelming probability) to distinguishing $h^1(\mathbf{s})$ from $U(\{0,1\})$ (with non-negligible advantage) given $A_{\mathbf{s},\Psi_\beta}$, for $\mathbf{s} \in \mathbb{Z}_q^n$ chosen uniformly at random and (secret) $\beta = \sqrt{\alpha^2/2 + l^2}$, where l is distributed according to $S^{(n)}_{\alpha/\sqrt{2n}}$.

In other words, h^1 is hard-core for $\mathsf{LWE}_{q,\Psi_\beta}$ assuming that $\mathsf{GapSVP}_{\zeta,\gamma}$ is hard in the worst case.

We start with a couple of elementary reductions that make the proof of Theorem 3.3 simpler. First, define a variant problem $\mathsf{GapSVP}'_{\zeta,\gamma}$ whose input, just as for $\mathsf{GapSVP}_{\zeta,\gamma}$, is a pair $(\mathbf{B}, d)$ such that $\lambda_1(\mathcal{L}(\mathbf{B})) \leq \zeta(n)$, $\min_i \|\tilde{\mathbf{b}}_i\| \geq 1$, and $1 \leq d \leq \zeta(n)/\gamma(n)$. It is a YES instance if there exists a $\mathbf{z} \in \mathbb{Z}^n$ such that z_1 is *odd* and $\|\mathbf{Bz}\| \leq d$; it is a NO instance if $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$.

Lemma 3.4. *For any $\zeta(n) \geq \gamma(n) \geq 1$, there is a deterministic polynomial-time Cook reduction from $\mathsf{GapSVP}_{\zeta,\gamma}$ to $\mathsf{GapSVP}'_{\zeta,\gamma}$.*

Proof. Given an input instance $(\mathbf{B}, d)$ of $\mathsf{GapSVP}_{\zeta,\gamma}$, the reduction generates n instances $(\mathbf{B}^{(i)}, d)$ for $i \in [n]$ as described below, and invokes the $\mathsf{GapSVP}'_{\zeta,\gamma}$ oracle on each of them. If the oracle accepts any of the instances, the reduction accepts, otherwise it rejects.

The instances $(\mathbf{B}^{(i)}, d)$ are defined as follows: for $i = 1$, let $\mathbf{B}^{(1)} = \mathbf{B}$. For $i = 2, \ldots, n$, let $\mathbf{b}_i^{(i)} = \mathbf{b}_i + \mathbf{b}_1$, and let $\mathbf{b}_j^{(i)} = \mathbf{b}_j$ for all $j \neq i$. Observe that $\mathcal{L}(\mathbf{B}^{(i)}) = \mathcal{L}(\mathbf{B})$ and that the Gram-Schmidt orthogonalizations of $\mathbf{B}$ and $\mathbf{B}^{(i)}$ are identical, for every $i \in [n]$. Therefore, the instances $(\mathbf{B}^{(i)}, d)$ satisfy the requirements of the GapSVP' problem.

If $(\mathbf{B}, d)$ is a NO instance of GapSVP, then by the first observations above, every $(\mathbf{B}^{(i)}, d)$ is a NO instance of GapSVP'.

If $(\mathbf{B}, d)$ is a YES instance of GapSVP, then there exists some $\mathbf{z} \in \mathbb{Z}^n$ such that $\mathbf{Bz}$ is a shortest nonzero vector in $\mathcal{L}(\mathbf{B})$ (i.e., $\|\mathbf{Bz}\| \leq d$) and an $i \in [n]$ such that z_i is *odd*; for if not, then $\mathbf{z} \in (2\mathbb{Z})^n$ and $\mathbf{Bz}/2 \in \mathcal{L}(\mathbf{B})$ is nonzero and shorter than $\mathbf{Bz}$, a contradiction. We claim that $(\mathbf{B}^{(i)}, d)$ is a YES instance of GapSVP'. If z_1 is odd, then we may take $i = 1$ and the claim holds trivially. Now suppose that z_1 is even. Letting $\mathbf{z}' \in \mathbb{Z}^n$ be such that $\mathbf{B}^{(i)}\mathbf{z}' = \mathbf{Bz}$, we have $z'_1 = z_1 - z_i$, which is odd, and the claim follows. □

Next, observe that $A_{\mathbf{s},\phi}$ for an *arbitrary* $\mathbf{s} \in \mathbb{Z}_q^n$ can be transformed into $A_{\mathbf{s}',\phi}$ for a *uniformly random* $\mathbf{s}' = \mathbf{s} + \mathbf{t} \in \mathbb{Z}_q^n$, simply by choosing $\mathbf{t} \in \mathbb{Z}_q^n$ uniformly at random and mapping each pair $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle / q) \in \mathbb{Z}_q^n \times \mathbb{T}$. Moreover, $h^1(\mathbf{s}') = h^1(\mathbf{s}) \oplus h^1(\mathbf{t})$ when q is even. Therefore, if we have an oracle D that distinguishes $h^1(\mathbf{s})$ from uniform with advantage δ given $A_{\mathbf{s},\phi}$ for *uniform* $\mathbf{s} \in \mathbb{Z}_q^n$, then we have an efficient predictor P that computes $h^1(\mathbf{s})$ with probability $1/2 + \delta$ given $A_{\mathbf{s},\phi}$ for *arbitrary* $\mathbf{s} \in \mathbb{Z}_q^n$.

The final tool we need is a technical lemma relating to the generation of samples from an LWE distribution.

Lemma 3.5 ([Reg05, Proof of Lemma 3.8]). *Let $\epsilon = \epsilon(n)$ be a negligible function, $q = q(n) \geq 2$ be an integer, and $\alpha = \alpha(n) \in (0, 1)$ be a real number. Let $\mathbf{B}$ be a basis for an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, let $r \geq \sqrt{2}q \cdot \eta_\epsilon(\Lambda)$, and let $\mathbf{x} \in \mathbb{R}^n$ be at distance d' from some $\mathbf{v} \in \Lambda$.*

Consider the following experiment: let $\mathbf{y}$ be drawn from $D_{\Lambda^*,r}$ and let $e \in \mathbb{R}$ be drawn from $D^1_{\alpha/\sqrt{2}}$. Then the distribution of

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

is within negligible statistical distance of $A_{\mathbf{s},\Psi_\beta}$, where $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$ and $\beta = \sqrt{\alpha^2/2 + (d'r/q)^2}$.

We are now ready to prove the theorem.

Proof of Theorem 3.3. By Lemma 3.4, we can say that the input to our reduction is an instance of $\mathsf{GapSVP}'_{\zeta,\gamma}$, i.e., a pair $(\mathbf{B}, d)$ where $\min\|\tilde{\mathbf{b}}_i\| \geq 1$, the minimum distance $\lambda_1(\mathcal{L}(\mathbf{B})) \leq \zeta$, and $1 \leq d \leq \zeta/\gamma$. Let $\Lambda = \mathcal{L}(\mathbf{B})$.

By the discussion above, we may hypothesize a predictor P that computes $h^1(\mathbf{s})$ with probability $1/2+\delta$ for some non-negligible $\delta = \delta(n)$ given $A_{\mathbf{s},\Psi_\beta}$ for *arbitrary* $\mathbf{s} \in \mathbb{Z}_q^n$, and β chosen as described in the theorem statement.

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

1. Choose a point $\mathbf{w} \in \mathbb{R}^n$ from distribution $D^n_{d\cdot\omega}$ for

$$\omega = \frac{\alpha \cdot \gamma}{2n} = \omega(\sqrt{\log n}),$$

   let $\mathbf{x} = \mathbf{w} \bmod \mathbf{B}$, and let $\mathbf{v} = \mathbf{x} - \mathbf{w} \in \Lambda$.

2. Invoke the hypothesized predictor P, simulating each desired sample from the LWE distribution as follows: using the algorithm from Proposition 2.8 on the reversed dual basis of $\mathbf{B}$, sample $\mathbf{y}$ from $D_{\Lambda^*,r}$ for
$$r = \frac{q \cdot \sqrt{2n}}{\gamma \cdot d}.$$
Next, sample $e \in \mathbb{R}$ from $D^1_{\alpha/\sqrt{2}}$ and give P the pair
$$(\mathbf{a} = \mathbf{B}^t\mathbf{y} \bmod q\ ,\ b = \langle \mathbf{y}, \mathbf{x}\rangle/q + e) \in \mathbb{Z}_q^n \times \mathbb{T}.$$

3. When P outputs a prediction, check whether the prediction equals $h^1(\mathbf{s})$, where $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$.

If P's prediction is correct in at least $(1/2 + \delta/2)N$ of the iterations, then *reject*, otherwise *accept*.

We now analyze the reduction. Just as in the proof of Theorem 3.1, for the definition of r above, the algorithm from Proposition 2.8 correctly samples from a distribution that is within negligible statistical distance of $D_{\Lambda^*,r}$.

Now consider the case when $(\mathbf{B}, d)$ is a NO instance of GapSVP', i.e., $\lambda_1(\Lambda) > \gamma \cdot d$. Just as in the proof of Theorem 3.1, we have $r \geq \sqrt{2}q \cdot \eta_\epsilon(\Lambda^*)$ as required by Lemma 3.5. Now because $\mathbf{v} = \mathbf{x} - \mathbf{w} \in \Lambda$, the distance between $\mathbf{x}$ and $\mathbf{v}$ is $d' = \|\mathbf{w}\|$, which means that $d'r/q$ is distributed according to $S_t^{(n)}$, where
$$t = d \cdot \omega \cdot r/q = \alpha/\sqrt{2n}.$$

By Lemma 3.5, it follows that the reduction simulates $A_{\mathbf{s},\Psi_\beta}$ (up to negligible statistical distance), where $\mathbf{s} = \mathbf{B}^{-1}\mathbf{v} \bmod q$ and $\beta = \sqrt{\alpha^2/2 + l^2}$, and l is distributed according to $S^{(n)}_{\alpha/\sqrt{2n}}$. By hypothesis, P predicts $h^1(\mathbf{s})$ with probability negligibly close to $1/2 + \delta$, so by a standard application of the Chernoff bound (for sufficiently large $N = \mathrm{poly}(n)$), P predicts correctly in at least $(1/2 + \delta/2)N$ iterations, and the reduction rejects as desired.

Finally, consider the case when $(\mathbf{B}, d)$ is a YES instance of GapSVP', i.e., there exists a $\mathbf{z} \in \mathbb{Z}^n$ such that z_1 is odd and $\|\mathbf{Bz}\| \leq d$. Observe that Step 2, which provides all the input to the predictor P, depends only on the fixed value of $\mathbf{x}$ and additional randomness that is independent of $\mathbf{w}$. Also observe that conditioned on the fixed value of $\mathbf{x}$, the random variable $\mathbf{v} = \mathbf{x} - \mathbf{w} \in \Lambda$ is distributed according to $D_{\Lambda,d\cdot\omega,\mathbf{x}}$. By Lemma 2.9, the parity of the first entry of $\mathbf{B}^{-1}\mathbf{v}$ is negligibly close to uniform, conditioned on the entire fixed input to P. Because q is even, the predicate $h^1(\mathbf{s})$ is also negligibly close to uniform, and P's prediction is correct with probability at most $1/2 + \mathrm{negl}(n)$. By the Chernoff bound, P predicts correctly in fewer than $(1/2 + \delta)N$ iterations, and the reduction accepts as desired. □

4 Public-Key Cryptosystems

Here we construct public-key cryptosystems (for multi-bit messages) that are based on the search version of LWE. We start in Section 4.1 by showing how to extend the particular hard-core predicate for LWE (shown in Section 3.2) into many simultaneously hard bits. Then in Section 4.2 we construct a semantically secure cryptosystem, followed in Section 4.3 by an extension that enjoys chosen-ciphertext security.

4.1 Simultaneous Hard-Core Bits for LWE

Lemma 4.1. *Let $\ell = \mathrm{poly}(n)$, $q = q(n) \geq 2$ be even, and ϕ be a distribution (itself possibly a random variable) on $\mathbb{T}$. If h^1 is hard-core for $\mathsf{LWE}_{q,\phi}$ in n dimensions, then h^ℓ is hard-core for $\mathsf{LWE}_{q,\phi}$ in $n + \ell - 1$ dimensions.*

More precisely, there is an efficient reduction from distinguishing $h^1(\mathbf{s})$ from $U(\{0,1\})$ (with non-negligible advantage) given $A_{\mathbf{s},\phi}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ to distinguishing $h^\ell(\mathbf{s}')$ from $U(\{0,1\}^\ell)$ (with non-negligible advantage) given $A_{\mathbf{s}',\phi}$ for uniformly random $\mathbf{s}' \in \mathbb{Z}_q^{n+\ell-1}$.

Proof. We proceed by a hybrid argument. If some D distinguishes between $h^\ell(\mathbf{s}')$ and U_ℓ given $A_{\mathbf{s}',\phi}$ (for uniform $\mathbf{s}' \in \mathbb{Z}_q^{n+\ell-1}$) with non-negligible advantage $\delta = \delta(n)$, then there is some $j \in [\ell]$ such that D distinguishes between $h^{j-1}(\mathbf{s}') \circ U_{\ell-j+1}$ and $h^j(\mathbf{s}') \circ U_{\ell-j}$ given $A_{\mathbf{s}',\phi}$ with non-negligible advantage $\delta'(n) = \delta(n)/\ell$.

We describe a reduction that, given $A_{\mathbf{s},\phi}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ and an input bit h, uses D to distinguish whether h is $h^1(\mathbf{s})$ or U_1. The reduction chooses $\mathbf{s}_l \in \mathbb{Z}_q^{j-1}$ and $\mathbf{s}_r \in \mathbb{Z}_q^{\ell-j}$ uniformly at random, and lets $h' = h^{j-1}(\mathbf{s}_l) \circ h \circ U_{\ell-j} \in \{0,1\}^\ell$. It invokes D on h', simulating $A_{\mathbf{s}',\phi}$ in the manner described below, and copies D's output.

Letting $\mathbf{s}' = \mathbf{s}_l \circ \mathbf{s} \circ \mathbf{s}_r$, we see that $\mathbf{s}'$ is distributed uniformly over $\mathbb{Z}_q^{n+\ell-1}$. It is also apparent that if the reduction's input bit h is uniform, then h' is distributed as $h^{j-1}(\mathbf{s}') \circ U_{\ell-j+1}$, whereas if $h = h^1(\mathbf{s})$, then h' is distributed as $h^j(\mathbf{s}') \circ U_{\ell-j}$. Therefore the reduction distinguishes between these two cases with non-negligible advantage δ'.

The reduction simulates $A_{\mathbf{s}',\phi}$ using $A_{\mathbf{s},\phi}$ as follows. Given a pair $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle / q + x) \in \mathbb{Z}_q^n \times \mathbb{T}$ from $A_{\mathbf{s},\phi}$, it chooses $\mathbf{a}_l \in \mathbb{Z}_q^{j-1}$ and $\mathbf{a}_r \in \mathbb{Z}_q^{\ell-j}$ uniformly at random and outputs the pair

$$(\mathbf{a}_l \circ \mathbf{a} \circ \mathbf{a}_r \ , \ \langle \mathbf{a}_l, \mathbf{s}_l \rangle / q + b + \langle \mathbf{a}_r, \mathbf{s}_r \rangle / q) = (\mathbf{a}', b' = \langle \mathbf{a}', \mathbf{s}' \rangle / q + x) \in \mathbb{Z}_q^{n+\ell-1} \times \mathbb{T}.$$

It is apparent that $\mathbf{a}'$ is distributed uniformly over $\mathbb{Z}_q^{n+\ell-1}$, thus, the simulation is faithful to $A_{\mathbf{s}',\phi}$. □

4.2 Trapdoor Functions and Basic Cryptosystem

Here we recall the collection of LWE-based injective trapdoor functions given in [GPV08], which build on ideas due to Goldreich, Goldwasser, and Halevi [GGH97]. For completeness, and due to some modifications and enhancements, we present a full description of the collection along with proofs of correctness and security. We then design a semantically secure cryptosystem around these trapdoor functions.

For consistency and simplicity of notation, we continue use n as the main parameter and hypothesize $\ell \geq 1$ simultaneous parity bits for LWE in n dimensions, with the understanding that this is based on a single parity predicate for the LWE problem in $n - \ell + 1$ dimensions by Lemma 4.1.

4.2.1 Generation

The first component is a special algorithm for generating a (nearly) uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ that serves as the index of the public function $g_{\mathbf{A}}$, together with a trapdoor $\mathbf{T}$ made up of vectors whose lengths are bounded by some relatively small L.[2] Ajtai [Ajt99] gave the first such generation algorithm for *odd* q, which yielded a bound $L = m^{2.5}$; recently, Alwen and Peikert [AP08] improved the algorithm to yield a tighter

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

bound $L \approx m$ for *arbitrary* q (recall that we use an even q in Theorem 3.3 and Lemma 4.1 for our particular choice of hard-core functions).

Proposition 4.2 ([Ajt99, AP08]). *For any positive integers n and $q \geq 3$, any $\delta > 0$ and $m \geq (2+\delta)n \lg q$, there is a probabilistic polynomial-time algorithm that outputs a pair $(\mathbf{T} \in \mathbb{Z}^{m \times m}, \mathbf{A} \in \mathbb{Z}_q^{n \times m})$ such that: the distribution of $\mathbf{A}$ is within negligible statistical distance of uniform over $\mathbb{Z}_q^{n \times m}$, $\mathbf{T}$ is nonsingular (over the rationals), $\|\mathbf{t}_i\| \leq L = O(m \log m)$ for every $i \in [m]$, and $\mathbf{AT} = \mathbf{0} \bmod q$.*

4.2.2 Evaluation

On index $\mathbf{A}$ and inputs $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{x} \in \mathbb{T}^m$, compute

$$\mathbf{b} = \mathbf{A}^t\mathbf{s}/q + \mathbf{x} \in \mathbb{T}^m.$$

Round each entry of $\mathbf{b}$ to the nearest multiple of $1/q'$ modulo 1, i.e., let $\mathbf{b}' = \lfloor q' \cdot \mathbf{b} \rceil / q' \in \mathbb{T}^m$. Output $g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \mathbf{b}'$, which may alternately be represented as $q' \cdot \mathbf{b}' \in \mathbb{Z}_{q'}^m$.

Lemma 4.3. *Let $\pi : \mathbb{Z}_q^n \to \{0,1\}^\ell$ be a function (e.g., $\pi = h^\ell$) and ϕ be a distribution (itself possibly a random variable) over $\mathbb{T}$. If π is hard-core for $\mathsf{LWE}_{q,\phi}$, then π is hard-core for the collection $\{g_{\mathbf{A}}\}$ under the input distribution where $\mathbf{s} \in \mathbb{Z}_q^n$ is uniformly random and $\mathbf{x}$ is drawn from ϕ^m.*

Proof. The proof follows immediately from the fact that $\mathbf{A}$ is negligibly close to uniform, and that an adversary given samples $(\mathbf{a}_i, b_i)$ from $A_{\mathbf{s},\phi}$ can round off each $b_i \in \mathbb{T}$ to the nearest multiple of $1/q'$ to simulate the output $\mathbf{b}'$ of $g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$. □

4.2.3 Inversion

A standard counting argument reveals that a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is full-rank (i.e., its rows are linearly independent modulo q) except with probability at most $q^n/2^m$, which is negligible in n when $m \geq (1+\delta)n \lg q$. For the remainder of the paper we implicitly assume that such an $\mathbf{A}$ is full-rank.

Observe that $\mathbf{A}^+ = \mathbf{A}^t(\mathbf{A}\mathbf{A}^t)^{-1} \in \mathbb{Z}_q^{m \times n}$ is the *right inverse* of $\mathbf{A}$ modulo q, because $\mathbf{A}\mathbf{A}^+ = \mathbf{I}_n$, the n-dimensional identity matrix modulo q. (Note that the Gram matrix $\mathbf{A}\mathbf{A}^t$ is invertible modulo q because $\mathbf{A}$ is full-rank.) Therefore, given $\mathbf{y} \in \mathbb{T}^m$ where $\mathbf{y} = (\mathbf{A}^t\mathbf{s})/q \bmod 1$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, we can recover $\mathbf{s}$ by computing

$$(\mathbf{A}^+)^t(q \cdot \mathbf{y}) = (\mathbf{A}\mathbf{A}^+)^t\mathbf{s} = \mathbf{s} \bmod q.$$

To invert $\mathbf{b}' = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) \in \mathbb{T}^m$ given the trapdoor $\mathbf{T}$, treat $\mathbf{b}'$ as an element of $\mathbb{R}^m$ and compute

$$\mathbf{y} = \mathbf{T}^{-t} \cdot \lfloor \mathbf{T}^t \cdot \mathbf{b}' \rceil \bmod 1,$$

and recover $\mathbf{s}$ from $\mathbf{y}$ as described above. (The exact value of $\mathbf{x}$ cannot always be recovered from $\mathbf{b}'$ due to rounding, but it is not needed in our applications.)

Lemma 4.4. *Let $q' = q'(n) \geq 2L\sqrt{m}$ and $\alpha = \alpha(n) \leq 1/(L \cdot \omega(\sqrt{\log n}))$. Then for any $\mathbf{s} \in \mathbb{Z}_q^n$ and for $\mathbf{x}$ chosen from Ψ_β^m for any $\beta \leq \alpha$, the inversion algorithm on $\mathbf{b}' = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$ correctly outputs $\mathbf{s}$ with overwhelming probability over the choice of $\mathbf{x}$.*

Proof. We start with a few facts that we later use to analyze the rounding step. First, let $\mathbf{w} \in \mathbb{R}^m$ be such that $|w_i| \leq 1/(2q')$ for all $i \in [m]$. Then for all $i \in [m]$, we have

$$|\langle \mathbf{t}_i, \mathbf{w} \rangle| \leq \|\mathbf{t}_i\| \cdot \|\mathbf{w}\| \leq L \cdot \sqrt{m}/(2q') \leq 1/4$$

by the Cauchy-Schwarz inequality and by hypothesis on $\|\mathbf{t}_i\|$ and q'. Second, suppose $\mathbf{x}' \in \mathbb{R}^m$ is distributed according to D_β^m for some $\beta \leq \alpha$. Then for all $i \in [m]$, the inner product $\langle \mathbf{t}_i, \mathbf{x}' \rangle$ is distributed according to D_r for $r = \|\mathbf{t}_i\| \cdot \beta \leq 1/\omega(\sqrt{\log n})$ by hypothesis on $\|\mathbf{t}_i\|$, α, and β. By the tail bound on Gaussian distributions, $|\langle \mathbf{t}_i, \mathbf{x}' \rangle| < 1/4$ except with probability $\exp(-\Omega(1/r^2)) = \mathrm{negl}(n)$.

Now consider the inversion algorithm on $b' = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$ where $\mathbf{x}$ is chosen from Ψ_β^m. By the definition of $g_{\mathbf{A}}$, there exist $\mathbf{w} \in \mathbb{R}^m$ with $|w_i| \leq 1/(2q')$ for all $i \in [m]$ and an $\mathbf{x}'$ distributed according to D_β^m such that

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

Thus,

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

Observe that $(\mathbf{AT}/q)$ is an integer matrix by hypothesis on $\mathbf{T}$, and $\mathcal{L}(\mathbf{T}^t) \subseteq \mathbb{Z}^m$ because $\mathbf{T}$ is an integer matrix. Therefore,

$$\lfloor \mathbf{T}^t \cdot \mathbf{b}' \rceil = (\mathbf{AT}/q)^t \cdot \mathbf{s} + \lfloor \mathbf{T}^t \cdot (\mathbf{x}' + \mathbf{w}) \rceil = \mathbf{T}^t(\mathbf{A}^t\mathbf{s}/q) \bmod \mathcal{L}(\mathbf{T}^t),$$

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

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

4.2.4 Cryptosystem and Analysis

Using the above collection of trapdoor functions, a public-key cryptosystem based on $\mathsf{GapSVP}_{\zeta,\gamma}$ (for γ determined below) is conceptually straightforward: to encrypt, evaluate $g_{\mathbf{A}}$ on a suitably random input, and mask the message by a hard-core function applied to the input. To decrypt, invert $g_{\mathbf{A}}$ to recover the input and remove the mask.

In detail, set the parameters as follows. Let $q = (\zeta/\sqrt{n}) \cdot \omega(\sqrt{\log n})$ be even, let $m = (2 + \delta)n \lg q$ for some $\delta > 0$, let $q' = 2L\sqrt{m} = \mathrm{poly}(n)$, and let $\alpha = 1/(L \cdot \omega(\sqrt{\log n}))$. Recall that $\mathsf{GapSVP}_\gamma^\zeta$ is equivalent to GapSVP_γ when $\zeta(n) = 2^{n/2}$, which implies $\log q = O(n)$. The other most interesting case is when $\zeta(n) = \mathrm{poly}(n)$, which implies $\log q = O(\log n)$.

- To generate a key pair, sample a function index $\mathbf{A}$ (the public key) with its trapdoor $\mathbf{T}$ (the secret key).
- To encrypt, choose $\mathbf{s} \in \mathbb{Z}_q^n$ uniformly at random and $\mathbf{x}$ according to Ψ_β^m for $\beta = \sqrt{\alpha^2/2 + l^2}$, where l is distributed according to $S^{(n-\ell+1)}_{\alpha/\sqrt{2(n-\ell+1)}}$. The encryption of message $\mu \in \{0,1\}^\ell$ is

$$(\mathbf{b}' = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) \ , \ c = h^\ell(\mathbf{s}) \oplus \mu).$$

- To decrypt a ciphertext $(\mathbf{b}', c)$ using $\mathbf{T}$, invert $\mathbf{b}'$ to find $\mathbf{s}$ and output $h^\ell(\mathbf{s}) \oplus c$.

The size of the public key $\mathbf{A}$ is $O(mn \log q) = O(n^2 \log^2 q)$ bits, and the trapdoor $\mathbf{T}$ has size $O(m^2 \log m)$. The size of the ciphertext is dominated by $\mathbf{b}'$, which requires $O(m \log q') = O(n \log q \log n)$ bits. By taking (say) $\ell = n/2$, the ciphertext is therefore an $O(\log q \log n)$ factor larger than the plaintext.

Proposition 4.5. *The cryptosystem described above is complete and semantically secure, assuming that* $\mathsf{GapSVP}_{\zeta,\gamma}$ *is hard in the worst case for some* $\gamma(n) = \tilde{O}(n^2 \log q)$.

Proof. Correctness of decryption (with overwhelming probability over the encryption randomness) is immediate by the fact that $\beta \leq \alpha$ with overwhelming probability, and by Lemma 4.4. Semantic security (assuming the worst-case hardness of $\mathsf{GapSVP}_{\zeta,\gamma}$) follows directly from the fact that h^ℓ is hard-core for $g_{\mathbf{A}}$ under the input distribution used for encryption, which follows by the sequence of Lemma 4.3, Lemma 4.1, and Theorem 3.3. We may therefore take the underlying worst-case approximation factor γ to be

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

Note that an improved bound L (or its Gram-Schmidt counterpart $\tilde{L}$ as described in Section 4.2.3 above) yields a tighter approximation factor γ. For example, if L (or $\tilde{L}$) were improved to the asymptotically optimal $O(\sqrt{m})$, the factor γ could be reduced to $\tilde{O}(n^{1.5}\sqrt{\log q})$.

4.3 Chosen-Ciphertext Security

To construct a cryptosystem that enjoys security under chosen-ciphertext attacks, we use a paradigm recently proposed by Peikert and Waters [PW08], and additional perspectives due to Rosen and Segev [RS08]. We discuss all the important technical ideas here, but defer a complete description and proof to the full version.

The main observation is that any $k = \text{poly}(n)$ independently chosen functions $g_{\mathbf{A}_1}, \ldots, g_{\mathbf{A}_k}$ remain one-way (assuming LWE is hard) even when evaluated on the *same* input $\mathbf{s}$ and independent $\mathbf{x}_1, \ldots, \mathbf{x}_k$ (respectively) from the appropriate error distribution ϕ. This is because the indices $\mathbf{A}_1, \ldots, \mathbf{A}_k$ and outputs $\mathbf{b}'_1 = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}_1), \ldots, \mathbf{b}'_k = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}_k)$ can be assembled simply by drawing $k \cdot m$ samples from $A_{\mathbf{s},\phi}$. Similarly, the function $h^\ell(\mathbf{s})$ remains hard-core given all these values, if it was hard-core for LWE in the first place. (We remark that these facts were also observed independently by Goldwasser and Vaikuntanathan [GV08], who construct similar chosen ciphertext-secure cryptosystems.) Essentially, the properties described above constitute security under "correlated inputs," as defined in [RS08].[3]

There is a simple (and black-box) chosen ciphertext-secure cryptosystem based on any collection of injective trapdoor functions that is secure under a suitable form of input correlation (including the one described above). Crucially, the proof of security requires the functions to be *injective*. More precisely, the following properties must hold with overwhelming probability over the choice of function g from the collection:

1. Each value y in the range has *at most one* legal preimage x under g.
2. Given any y and any candidate preimage x (and the description of g), one can efficiently check whether x is the legal preimage of y (without knowledge of the trapdoor).
3. Given any y and the trapdoor for g, the inverter *always* finds the legal preimage x of y (if it exists).

[3] These observations can also be used to construct a relaxed kind of "all-but-one" function as defined in [PW08], but we find the terminology of correlated inputs to be more natural in this context.

These properties ensure that for any y (possibly constructed adversarially), the following two algorithms behave *identically*: (1) on input x, y, accept if x is the preimage of y; (2) on input y and the trapdoor, run the inverter to get some x, and accept if x is indeed the preimage of y. This identical behavior is the crux of the security proof.

Making our functions injective. Note that in the above description of the trapdoor functions $g_{\mathbf{A}}$, *any* value $\mathbf{s} \in \mathbb{Z}_q^n$ is a potential preimage of $\mathbf{b}' \in \mathbb{T}^m$, under the (possibly very unlikely) error vector $\mathbf{x} = \mathbf{b}' - (\mathbf{A}^t\mathbf{s})/q \in \mathbb{T}^m$. Therefore, we need to restrict the notion of a legal preimage and prove that it satisfies the three properties listed above. In particular, must carefully deal with the behavior of the inversion algorithm on *arbitrary* (possibly adversarial) values $\mathbf{b}' \in \mathbb{T}^m$, as opposed to those generated honestly. We stress that in our context, the error component $\mathbf{x}$ of the input need not be considered as part of the preimage, because it is not needed to check validity, nor is it used in the encryption.

We now define the notion of legal preimages for a function $g_{\mathbf{A}}$, which depends on the parameter $\alpha = \alpha(n) \in (0, 1)$ associated with the collection, and some arbitrary $t = t(n) = \omega(\sqrt{\log n})$. Define the absolute value $|\cdot|$ on $\mathbb{T} = [0, 1)$ as $|x| = \min\{x, 1 - x\}$, and extend it coordinate-wise to $\mathbb{T}^m$.

Definition 4.6. We say that $\mathbf{s} \in \mathbb{Z}_q^n$ is a legal preimage of $\mathbf{b}' \in \mathbb{T}^m$ under $g_{\mathbf{A}}$ if and only if every entry of $\left|\mathbf{b}' - (\mathbf{A}^t\mathbf{s})/q\right|$ is strictly less than $\alpha \cdot t$.

Let $q' \geq 1/(\alpha \cdot t)$. First, we observe that $\mathbf{s}$ is indeed a legal preimage of an honestly-generated $\mathbf{b}' = g_{\mathbf{A}}(\mathbf{s}, \mathbf{x})$, with overwhelming probability over the choice of $\mathbf{x}$ from any Ψ_β^m where $\beta \leq \alpha$ (this is required for completeness of the cryptosystem). Indeed, for every $i \in [m]$, we have $|x_i| < \alpha \cdot t/2$ with overwhelming probability by the Gaussian tail bound, and after the rounding step,

$$\left|b_i' - b_i\right| \leq 1/(2q') \leq \alpha \cdot t/2.$$

Proposition 4.7. *The three properties listed above are satisfied under Definition 4.6.*

Proof. Property 2 holds trivially by definition. Property 1 follows by a simple fact that holds with all but $q^n/2^m = \mathrm{negl}(n)$ probability over the choice of $\mathbf{A}$: for every nonzero $\mathbf{s} \in \mathbb{Z}_q^n$, $(\mathbf{A}^t\mathbf{s})/q \bmod 1$ has at least one entry with absolute value greater than $1/4$. (This can be seen by analyzing the probability for any fixed nonzero $\mathbf{s}$, then invoking the union bound.) Then for $\alpha < 1/(8t)$, every $\mathbf{b}'$ has at most one legal preimage by the triangle inequality.

For Property 3, we observe that for any $\mathbf{b}'$ that has a legal preimage $\mathbf{s}$, there is a vector $\mathbf{w} \in \mathbb{R}^m$ such that $\|\mathbf{w}\| \leq \sqrt{m} \cdot \alpha \cdot t$ and

$$\mathbf{b}' = (\mathbf{A}^t\mathbf{s})/q + \mathbf{w} \bmod \mathbb{Z}^m.$$

Then by following the proof of Lemma 4.4 (without the randomized component $\mathbf{x}'$), we see that the inversion algorithm *always* correctly recovers $\mathbf{s}$ as long as $\alpha \leq 1/(L \cdot \sqrt{m} \cdot t) = 1/(L \cdot \sqrt{m} \cdot \omega(\sqrt{\log n}))$. Note that the parameter α here is smaller than the one in Lemma 4.4 by a factor of $\sqrt{m}$, due to the "worst-case" inversion requirement. This allows for an underlying worst-case approximation factor

$$\gamma(n) = \tilde{O}(n/\alpha) = \tilde{O}(L \cdot n \cdot \sqrt{m}) = \tilde{O}(n^{2.5} \log^{1.5} q). \qquad \square$$

Acknowledgments

We thank Oded Regev for helpful discussions and observations.

References

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

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

[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.

[AP08] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. Manuscript, 2008.

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

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

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

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

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

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

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

[GV08] Shafi Goldwasser and Vinod Vaikuntanathan. Correlation-secure trapdoor functions from lattices, 2008. Manuscript.

[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.

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

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

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

[Pei08] 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.

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

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.

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

[Reg04b] Oded Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004.

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

[RS08] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. Cryptology ePrint Archive, Report 2008/116, 2008. `http://eprint.iacr.org/2008/116`.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/