The Learning with Errors Problem

Oded Regev[*]

Abstract

In this survey we describe the Learning with Errors (LWE) problem, discuss its properties, its hardness, and its cryptographic applications.

1 Introduction

In recent years, the *Learning with Errors* (LWE) problem, introduced in [Reg05], has turned out to be an amazingly versatile basis for cryptographic constructions. Its main claim to fame is being as hard as worst-case lattice problems, hence rendering all cryptographic constructions based on it secure under the assumption that worst-case lattice problems are hard. Our goal in this survey is to present the state-of-the-art in our understanding of this problem. Although all results presented here already appear in the literature (except for the observation in Appendix A), we tried to make our presentation somewhat simpler than that in the original papers. For more information on LWE and related problems, see some of the recent surveys on lattice-based cryptography [MR08, Pei09b, Mic07, Reg06].

LWE. The LWE problem asks to recover a secret $\mathbf{s} \in \mathbb{Z}_q^n$ given a sequence of 'approximate' random linear equations on $\mathbf{s}$. For instance, the input might be

$$\begin{aligned}
14s_1 + 15s_2 + 5s_3 + 2s_4 &\approx 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 &\approx 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + 1s_4 &\approx 3 \pmod{17} \\
10s_1 + 4s_2 + 12s_3 + 16s_4 &\approx 12 \pmod{17} \\
9s_1 + 5s_2 + 9s_3 + 6s_4 &\approx 9 \pmod{17} \\
3s_1 + 6s_2 + 4s_3 + 5s_4 &\approx 16 \pmod{17} \\
&\vdots \\
6s_1 + 7s_2 + 16s_3 + 2s_4 &\approx 3 \pmod{17}
\end{aligned}$$

[*] Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. Supported by the Israel Science Foundation, by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, by a European Research Council (ERC) Starting Grant, and by The Wolfson Family Charitable Trust.

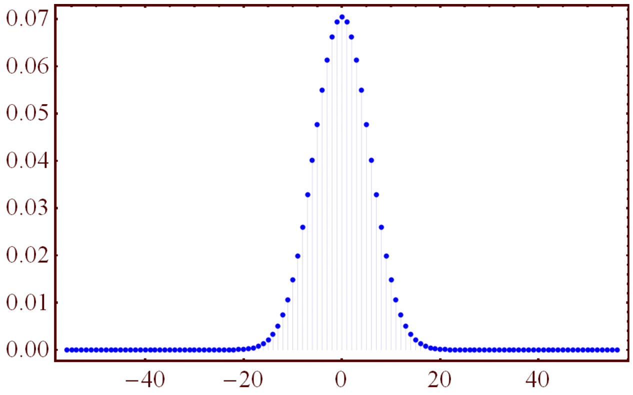

Figure 1: The error distribution with $q = 113$ and $\alpha = 0.05$.

where each equation is correct up to some small additive error (say, ±1), and our goal is to recover $\mathbf{s}$. (The answer in this case is $\mathbf{s} = (0, 13, 9, 11)$.)

If not for the error, finding $\mathbf{s}$ would be very easy: after about n equations, we can recover $\mathbf{s}$ in polynomial time using Gaussian elimination. Introducing the error seems to make the problem significantly more difficult. For instance, the Gaussian elimination algorithm takes linear combinations of n equations, thereby amplifying the error to unmanageable levels, leaving essentially no information in the resulting equations.

Let us define the problem more precisely. Fix a size parameter $n \geq 1$, a modulus $q \geq 2$, and an 'error' probability distribution χ on $\mathbb{Z}_q$. Let $A_{\mathbf{s},\chi}$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ be the probability distribution obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}_q$ according to χ, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where additions are performed in $\mathbb{Z}_q$, i.e., modulo q.[1] We say that an algorithm solves LWE with modulus q and error distribution χ if, for any $\mathbf{s} \in \mathbb{Z}_q^n$, given an arbitrary number of independent samples from $A_{\mathbf{s},\chi}$ it outputs $\mathbf{s}$ (with high probability). We note that one can equivalently view LWE as the problem of decoding from random linear codes, or as a random bounded distance decoding problem on lattices. Also, we note that the special case $q = 2$ corresponds to the well-known *learning parity with noise* (LPN) problem. In this case the error distribution χ can be specified by just one parameter $\varepsilon > 0$, namely, the probability of error (i.e., the probability of 1).

Parameters. In all applications and throughout this survey, the error distribution χ is chosen to be a normal distribution rounded to the nearest integer (and reduced modulo q) of standard deviation αq where $\alpha > 0$ is typically taken to be $1/\text{poly}(n)$ (see Figure 1). In fact, as we shall mention later, it can be shown that this distribution is in some sense 'LWE-complete'. The modulus q is typically taken to be polynomial in n. Choosing an exponential modulus q has the disadvantage of significantly increasing the size of the input (as a function of n), thereby making cryptographic applications considerably less efficient, but does have the advantage that the hardness of the problem is somewhat better understood; see below for the details. The number of equations seems to

[1] It is usually more natural and mathematically cleaner to consider a *continuous* error distribution; for simplicity we ignore this issue in this survey.

be, for most purposes, insignificant. For instance, the known hardness results are essentially independent of it. This can be partly explained by the fact that from a given fixed polynomial number of equations, one can generate an arbitrary number of additional equations that are essentially as good as new, with only a slight worsening in the error distribution. This property was shown in [GPV08, ACPS09] (see also [Lyu05]) and will be discussed again in Section 2.

Algorithms. One naïve way to solve LWE is through a maximum likelihood algorithm. Assume for simplicity that q is polynomial and that the error distribution is normal, as above. Then, it is not difficult to prove that after about $O(n)$ equations, the only assignment to $\mathbf{s}$ that 'approximately satisfies' the equations is the correct one. This can be shown by a standard argument based on Chernoff's bound and a union bound over all $\mathbf{s} \in \mathbb{Z}_q^n$. This leads to an algorithm that uses only $O(n)$ samples, and runs in time $2^{O(n \log n)}$. As a corollary we obtain that LWE is 'well-defined' in the sense that with high probability the solution $\mathbf{s}$ is unique, assuming the number of equations is $\Omega(n)$.

Another, even more naïve algorithm is the following: keep asking for LWE samples until seeing poly(n) equations of the form $s_1 \approx \ldots$ (i.e., a pair $(\mathbf{a}, b)$ where $\mathbf{a} = (1, 0, \ldots, 0)$), at which point we can recover the value of s_1. We then repeat this for all s_i. The probability of seeing such an equation is q^{-n}, leading to an algorithm requiring $2^{O(n \log n)}$ equations, and with a similar running time.

A much more interesting algorithm follows from the work of Blum, Kalai, and Wasserman [BKW03], and requires only $2^{O(n)}$ samples and time. It is based on a clever idea that allows to find a small set S of equations (say, of size n) among $2^{O(n)}$ equations, such that $\sum_S \mathbf{a}_i$ is, say, $(1, 0, \ldots, 0)$ (in brief: partition the n coordinates into $\log n$ blocks of size $n/\log n$ each, and construct S recursively by finding collisions in blocks). By summing these equations we can recover the first coordinate of $\mathbf{s}$ (and similarly for all other coordinates).

Somewhat surprisingly, the Blum et al. algorithm is the best known algorithm for the LWE problem. This is closely related to the fact that the best known algorithms for lattice problems [AKS01, MV10] require $2^{O(n)}$ time. Any algorithmic improvement on LWE is likely to lead to a breakthrough in lattice algorithms.

Hardness. There are several reasons to believe the LWE problem is hard. First, because the best known algorithms for LWE run in exponential time (and even quantum algorithms don't seem to help). Second, because it is a natural extension of the LPN problem, which is itself an extensively studied problem in learning theory that is widely believed to be hard. Moreover, LPN can be formulated as the problem of decoding from random linear binary codes, hence any algorithmic progress on LPN is likely to lead to a breakthrough in coding theory.

Third, and most importantly, because LWE is known to be hard based on certain assumptions regarding the worst-case hardness of standard lattice problems such as GapSVP (the decision version of the shortest vector problem) and SIVP (the shortest independent vectors problem) [Reg05, Pei09a]. More precisely, when the modulus q is exponential, hardness is based on the standard assumption that GapSVP is hard to approximate to within polynomial factors. For polynomial moduli q (which is the more interesting setting for cryptographic applications), the hardness is based on slightly less standard (but still quite believable) assumptions. Namely, either that GAPSVP is hard to approximate even given a 'hint' in the form of a short basis, or that GAPSVP or SIVP are hard to approximate to within polynomial factors even with a quantum computer.

We note that SIVP is in some sense harder than GAPSVP, and the fact that we have hardness based on SIVP is crucial for establishing the hardness of the ring-LWE problem described below. We also note that the above results require q to be somewhat large (typically, at least $\sqrt{n}$) and hence do not apply to LPN, i.e., the case $q = 2$. Finally, we remark that the approximation factors obtained for the worst-case lattice problems are typically of the form $\tilde{O}(n/\alpha)$, explaining why we prefer to have $\alpha \geq 1/\text{poly}(n)$. A further discussion of the known hardness results will be given in Section 2.

Thanks to these strong hardness results, a reader skeptical of the hardness of LWE can view the work in this area as attempts to find (possibly quantum) algorithms for worst-case lattice problems (and admittedly, this was the original motivation for [Reg05]).

Variants. As we will describe in Section 3, the LWE problem can be reduced to many, apparently easier, problems. These reductions are one of the main reason the LWE problem finds so many applications in cryptography. Among these reductions is a search to decision reduction, showing that it suffices to distinguish LWE samples from entirely uniform samples, and a worst-case to average-case reduction, showing that it suffices to solve this distinguishing task for a *uniform* secret $\mathbf{s} \in \mathbb{Z}_q^n$. Both reductions are quite simple, yet extremely useful. We will also describe some recent reductions that are somewhat less trivial.

Cryptographic applications. The LWE problem has turned out to be amazingly versatile as a basis for cryptographic constructions, partly due to its extreme flexibility as evidenced by the variants of LWE described above. Among other things, LWE was used as the basis of public-key encryption schemes secure under chosen-plaintext attacks [Reg05, KTX07, PVW08][2] and chosen-ciphertext attacks [PW08, Pei09a], oblivious transfer protocols [PVW08], identity-based encryption (IBE) schemes [GPV08, CHKP10, ABB10], various forms of leakage-resilient encryption (e.g., [AGV09, ACPS09, DGK$^+$10, GKPV10]), and more. The LWE problem was also used to show hardness results in learning theory [KS06]. (We note that cryptographic applications of LPN, i.e., the case $q = 2$, also exist but seem to be much more limited; see, e.g., [BFKL93, HB01, Ale03, JW05, GRS08, ACPS09].)

All the above results rely on the LWE problem; through the known hardness results, one immediately obtains hardness based on worst-case lattice problems. This gives LWE-based cryptography strong security guarantees not shared by most other cryptographic constructions, such as conjectured security against quantum computers. In addition, LWE is attractive as it typically leads to efficient implementations, involving low complexity operations (often mainly additions).

[2] See also [Ajt05] for a public key cryptosystem that seems to have many properties in common with the LWE-based one in [Reg05].

Origins. The origins of the LWE problem can be traced back to the celebrated work of Ajtai and Dwork [AD97], which describes the first public key cryptosystem whose security is based on worst-case lattice problems, and to the simplifications and improvements that followed it [GGH97, Reg03] (see also [AD07]). Although the LWE problem does not appear in any of those papers, a careful inspection of the Ajtai-Dwork construction in the simplified form given in [Reg03] reveals that a hardness result for the LWE problem is already implicit there. We provide a few more details on this observation in Appendix A.

This might be a good opportunity to point out that early work on the topic (including [AD97, Reg03]) based the hardness on a not so well known lattice problem called unique-SVP, and for a long time it was not clear whether it could be replaced with more standard lattice problems. Recently, Peikert [Pei09a] and Lyubashevsky and Micciancio [LM09] realized that unique-SVP is essentially equivalent to the standard lattice problem GAPSVP, leading to a considerably cleaner picture of the area. Our presentation here incorporates this observation.

The SIS **problem.** The LWE problem has a 'dual' problem known as the *SIS problem* (which stands for Small Integer Solution). In the SIS problem, we are given a sequence of vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots$ chosen uniformly from $\mathbb{Z}_q^n$ and asked to find a subset of them (or more generally, a combination with small coefficients) that sums to zero (modulo q). One can equivalently think of SIS as the problem of finding short vectors in a random lattice or code. The SIS problem was introduced in [MR04], and has its origins in the ground-breaking work of Ajtai [Ajt96], where for the first time, a cryptographic primitive based on the worst-case hardness of lattice problems was shown. The SIS problem is used in the construction of so-called 'minicrypt' primitives, such as one-way functions [Ajt96], collision resistant hash functions [GGH96], digital signature schemes [GPV08, CHKP10], and identification schemes [MV03, Lyu08, KTX08]; this is in contrast to LWE, whose cryptographic applications are typically of the 'cryptomania' type (i.e., public-key encryption and beyond). The hardness of the SIS problem is quite well understood [MR04]: it is known that for any q that is at least some polynomial in n, solving the SIS problem implies a solution to standard worst-case lattice problems such as SIVP and GAPSVP.

Ring-LWE**.** Cryptographic schemes based on the SIS and LWE problems tend to require rather large key sizes, typically on the order of n^2. This is because for cryptographic applications, one typically needs to provide at least n vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{Z}_q^n$ leading to key sizes of order n^2. Reducing this to almost linear size is highly desirable from a practical point of view, and as we shall see below, might also lead to interesting theoretical developments.

One natural way to achieve this goal (which is related to the idea underlying the heuristic design of the NTRU cryptosystem [HPS98]) is to assume that there is some structure in the LWE (or SIS) samples. More specifically, one kind of 'structure' that is often considered is the following. Assume n is a power of two (for reasons that will be explained shortly), and assume that the $\mathbf{a}$ vectors arrive in blocks of n samples $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{Z}_q^n$, where $\mathbf{a}_1 = (a_1, \ldots, a_n)$ is chosen uniformly as before, and the remaining vectors are given by $\mathbf{a}_i = (a_i, \ldots, a_n, -a_1, \ldots, -a_{i-1})$. Notice that representing n vectors now requires only $O(n)$ elements of $\mathbb{Z}_q$, as opposed to $O(n^2)$. Also, using the fast Fourier transform, operations on such vectors can be significantly sped up, leading to cryptographic constructions that not only have smaller keys, but are also considerably faster.

Mathematically speaking, the above can be interpreted as replacing the group $\mathbb{Z}_q^n$ with the ring $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$. For this reason, these structured variants are called ring-SIS and ring-LWE. Other choices of rings are possible, but for simplicity we focus here on the one above. The requirement that n is a power of 2 is meant to guarantee that $x^n + 1$ is irreducible over the rationals, as otherwise certain things start behaving badly (e.g., it is easy to see that both ring-SIS and the decision version of ring-LWE over the somewhat more familiar 'cyclic ring' $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$ are easy, and this can be seen as a consequence of the factorization $x^n - 1 = (x-1)(1 + x + \cdots + x^{n-1})$).

But are these special cases of SIS and LWE still hard? In the case of SIS, this was established in a sequence of papers starting with Micciancio's work [Mic02], which placed this study on firm theoretical grounds, and continuing with work by Peikert and Rosen [PR06], and Lyubashevsky and Micciancio [LM06].[3] Just like the hardness results for standard SIS, these results show that solving the ring-SIS problem implies a solution to worst-case instances of lattice problems. However, the worst-case lattice problems are restricted to the family of *ideal lattices*, which are lattices that satisfy a certain symmetry requirement (for instance, that if $(x_1, \ldots, x_n)$ is a lattice vector, then so is $(x_2, \ldots, x_n, -x_1)$). It is reasonable to conjecture that lattice problems on such lattices are still hard; there is currently no known way to take advantage of that extra structure, and the running time required to solve lattice problems on such lattices is the same as that for general lattices. One caveat is that GAPSVP is actually *easy* on ideal lattices (due to what seems like 'technical' reasons); this is why hardness results based on ideal lattices always use problems like SIVP (or the search version of GAPSVP, called SVP, which over ideal lattice is more or less equivalent to SIVP).

Several cryptographic constructions based on the hardness of the ring-SIS problem were developed, including collision resistant hash functions [PR06, LM06, LMPR08], identification schemes [Lyu09], and signature schemes [LM08, Lyu09]. As mentioned above, these systems typically boast small key sizes, and extremely fast computations.

Obtaining analogous hardness results for ring-LWE turned out to be quite nontrivial, and was only achieved very recently. Stehlé, Steinfeld, Tanaka, and Xagawa [SSTX09] showed (based on [Reg05]) a quantum reduction from (ring-)SIS to (ring-)LWE. Although this way of establishing the hardness of LWE through that of SIS seems quite elegant and attractive, the resulting hardness unfortunately deteriorates with the number of LWE samples m and even becomes vacuous when m is greater than q^2; in contrast, in the hardness result for standard LWE, the number of LWE samples plays an essentially insignificant role. It is not clear if this disadvantage is an inherent limitation of any SIS to LWE reduction.

A hardness result for ring-LWE that is independent of the number of samples was given in [LPR10]. Instead of showing a reduction from the SIS problem, the proof in [LPR10] follows the outline of the original LWE hardness proof of [Reg05]. We note that in order for the hardness not to deteriorate with the number of samples, the error distribution in the LWE instances is required to have a somewhat unusual 'ellipsoidal' shape. This might be an artifact of the proof, although it

[3] Specifically, Micciancio initiated the theoretical study of ring-SIS by proving that its inhomogeneous variant over $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$ is hard. Later work showed that in order to extend this result to (the homogeneous) ring-SIS, one needs to either restrict the domain of the solutions [PR06], or switch to a different ring, such as the ring $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ considered in this survey [LM06].

might also be the case that ring-LWE with the more natural spherical error is somehow easier.

It was also shown in [LPR10] that ring-LWE shares some of the nice properties of the standard LWE problem, most notably an equivalence between the decision version and the search version. This is particularly useful for efficient cryptographic applications, as without such an equivalence, one is forced to rely on hard-core bit constructions, which typically damage the efficiency (or alternatively, force us to make very strong hardness assumptions). In Section 5 we give a self-contained description of this reduction as well as other details on the ring-LWE problem.

Open questions. Being a relatively new problem, there are still many open questions surrounding the LWE problem. The first is regarding the hardness of LWE. As mentioned above, the known LWE hardness results (for the polynomial modulus case) are based on somewhat non-standard assumptions. It would be most desirable to improve this and reach a situation similar to the one known for the SIS problem. The bottleneck is that the lattice problem solved using LWE is somewhat non-standard, and the only known way to reduce standard problems to it is using a quantum reduction.

A related problem is to understand the hardness of the LPN problem. The hardness proof of LWE does not work for small moduli. Or perhaps there is an efficient algorithm for LPN that awaits to be discovered? Such an algorithm must make use of the small modulus.

The hardness of ring-LWE and of lattice problems on ideal lattices is still not too well understood. Can we reduce lattice problems on general lattices to problems on ideal lattices (possibly with some increase in the dimension)? Another intriguing possibility is that problems on ideal lattices are easier than those on general lattices. The reductions to ring-LWE and its variants might lead to the discovery of such algorithms.

The range of cryptographic applications of the LWE problem is already quite impressive, and will surely continue to grow in the next few years. One open question is to give a direct construction of a family of efficient pseudorandom functions. Another outstanding question is whether LWE (or more likely ring-LWE) can be used as the security assumption underlying the construction of a fully-homomorphic encryption scheme. The recent breakthrough construction by Gentry [Gen09] is not based on the LWE problem, but is instead based on quite strong assumptions on the hardness of lattice problems and subset sum. Building a scheme based on the LWE problem would give us more confidence in its security, and might have some other advantages, such as higher efficiency and practicality.

One ambitious goal is to show reductions from LWE to classical problems in the literature, akin to Feige's work on implications of the conjectured hardness of refuting random 3SAT instances [Fei02]. See Alekhnovich's paper [Ale03] for some related work.

Outline. In the next section we describe the known hardness results for LWE. That section is somewhat more technical than the rest of this survey and the reader might wish to skip some parts of it on first reading. In Section 3 we describe several equivalent variants of LWE that demonstrate its flexibility. We give an example of one very simple cryptographic application in Section 4, and in Section 5 we discuss recent work on ring-LWE.

Acknowledgments: I thank Vadim Lyubashevsky, Daniele Micciancio, and Chris Peikert for many helpful comments and discussions.

2 Hardness

In this section we provide a simplified description of the LWE hardness results from [Reg05, Pei09a].

As mentioned above, Stehlé et al. [SSTX09] recently suggested an alternative way to establish the hardness of LWE, namely, through a quantum reduction from the SIS problem (whose hardness is quite well understood). Since this reduction unfortunately leads to a qualitatively weaker hardness result for LWE with an undesirable dependence on the number of samples m, we will not discuss it here further. It would be interesting to see if their approach can lead to a hardness result that is qualitatively similar to those described below.

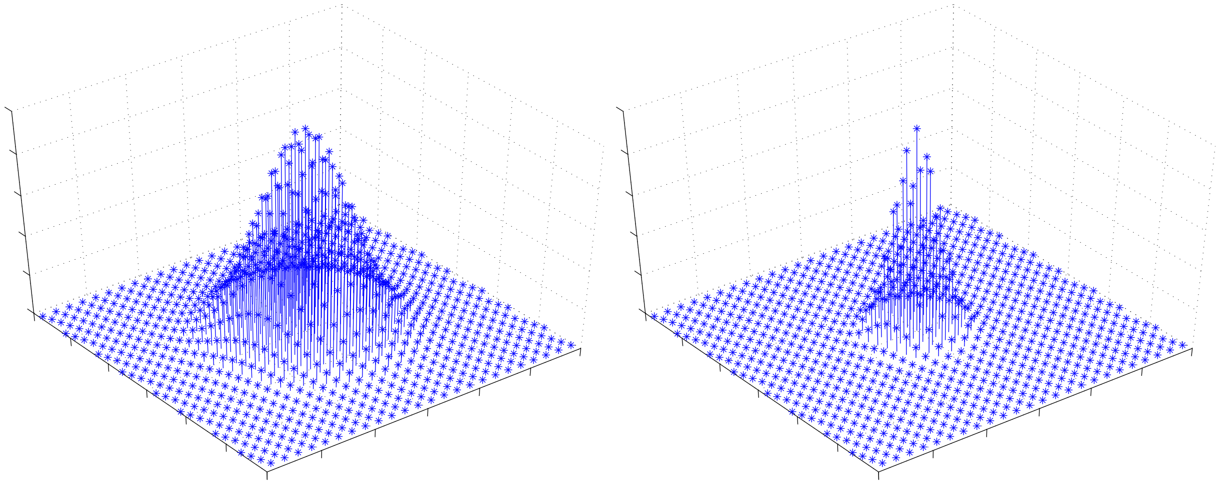

Figure 2: $D_{\Lambda,2}$ (left) and $D_{\Lambda,1}$ (right) for a two-dimensional lattice Λ. The z-axis represents probability.

Before presenting the LWE hardness result, we need a few very basic definitions related to lattices and lattice problems. A *lattice* in $\mathbb{R}^n$ is defined as the set of all integer combinations of n linearly independent vectors. This set of vectors is known as a *basis* of the lattice and is not unique. The *dual* of a lattice Λ in $\mathbb{R}^n$, denoted Λ^*, is the lattice given by the set of all vectors $\mathbf{y} \in \mathbb{R}^n$ such that $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$ for all vectors $\mathbf{x} \in \Lambda$. For instance, for any $t > 0$, the dual of $t\mathbb{Z}^n$ is $\mathbb{Z}^n/t$. We let $\lambda_1(\Lambda)$ denote the length of the shortest nonzero vector in the lattice Λ. We define the *discrete Gaussian distribution* on Λ with parameter r, denoted $D_{\Lambda,r}$, as the distribution that assigns mass proportional to $e^{-\pi\|\mathbf{x}/r\|^2}$ to each point $\mathbf{x} \in \Lambda$ (see Figure 2). Samples from $D_{\Lambda,r}$ are lattice vectors of norm roughly $\sqrt{n}r$ (assuming r is not too small).

In computational problems involving lattices, lattices are always represented in terms of an (arbitrary) basis. As a rule of thumb, 'algebraic' questions regarding lattices are easy, e.g., deciding if a vector is contained in a lattice, computing the dual lattice, etc. Geometric questions are typically hard. One of the most well-known geometric questions is $\textsc{GapSVP}_\gamma$. Here, we are given a lattice Λ and are asked to approximate $\lambda_1(\Lambda)$ to within a multiplicative factor of γ. The problem is known to be NP-hard for small γ, and easy for exponential approximation factors $\gamma = 2^{O(n)}$ [LLL82]. But our focus (and of the entire lattice-based cryptography literature) is on polynomial approximation factors of the form n^c where c is a constant that is typically at least 1. For these approximation factors, the problem is believed not to be NP-hard, but is nevertheless believed to be hard. For instance, the best known algorithms for it run in exponential time $2^{O(n)}$, and even quantum computers don't seem to improve this in any way. In order to avoid unnecessary technicalities, we will typically ignore here the exact approximation factor (be it n, n^2, etc.) and just say that, e.g., the hardness is based on $\textsc{GapSVP}$.

A second well-known geometric problem is SIVP_γ; here we are asked to find a set of n linearly independent vectors in a given lattice, such that the length of the longest vector in the set is at most γ times longer than the shortest possible for any such set. All the properties of $\textsc{GapSVP}$ mentioned above also apply to SIVP. Another problem that will appear below is the bounded distance decoding problem (BDD); here, for some distance parameter $d > 0$, we are given a lattice Λ and a point $\mathbf{x}$ within distance at most d of Λ, and asked to find the closest lattice vector to $\mathbf{x}$. Notice that as long as $d < \lambda_1(\Lambda)/2$, there is a unique correct answer.

The core of the LWE hardness results is the following proposition from [Reg05], whose proof will be given towards the end of this section.

Proposition 2.1. *Let $q \geq 2$ be an integer and α be a real number in $(0,1)$. Assume we are given access to an oracle that solves the* LWE *problem with modulus q and error parameter α. Then, given as input any lattice Λ, a large enough polynomial number of samples from the discrete Gaussian distribution $D_{\Lambda^*,r}$ for some (not too small) r, and a point $\mathbf{x}$ within distance $\alpha q/(\sqrt{2}r)$ of Λ, we can output the (unique) closest lattice point to $\mathbf{x}$ in polynomial time.*

The requirement that r is not too small is very mild; the precise condition is $r \geq \sqrt{2}q \cdot \eta_\varepsilon(\Lambda^*)$ where η is the so-called smoothing parameter and ε is some negligible amount; for our purposes, one can instead use the slightly stronger condition $r \geq q\sqrt{2n}/\lambda_1(\Lambda)$, or even ignore this condition altogether. (Notice that some kind of lower bound on r is necessary as otherwise, if r is too small, the distribution $D_{\Lambda^*,r}$ is essentially a deterministic distribution on the origin, samples from which can be trivially obtained. Moreover, if $r < \sqrt{2}\alpha q/\lambda_1(\Lambda)$, the above bounded distance decoding problem does not always have a unique solution.)

In order to understand the significance of this proposition, it is useful to contrast it with a result from [AR04, LLM06] which says the following: given as input a lattice Λ, a large enough polynomial number of samples from the discrete Gaussian distribution $D_{\Lambda^*,r}$ for some (not too small) r, and a point $\mathbf{x}$ within distance $O(\sqrt{\log n}/r)$ of Λ, we can output the (unique) closest lattice point to $\mathbf{x}$ in polynomial time. Notice that no LWE oracle is required in this algorithm. So what the proposition shows is that using an LWE oracle, the decoding radius can be increased from $O(\sqrt{\log n}/r)$ to $\alpha q/(\sqrt{2}r)$.

The above can already be interpreted as an indication that the LWE problem is hard, as it allows us to solve a worst-case lattice problem (BDD given a hint in the form of samples from the discrete Gaussian distribution) that we do not know how to solve otherwise. More specifically, for

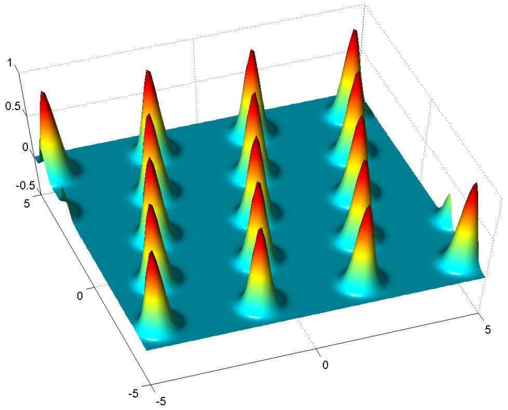

Figure 3: The quantum state.

$\alpha q = \Omega(\sqrt{n})$, the best algorithms we have to solve the problem require exponential time.

It would be, however, preferable to relate this problem to more standard lattice problems. Exactly such a result was obtained by Peikert [Pei09a]. His main realization (which was made explicit by [LM09]) is that there is a polynomial time reduction from the standard lattice problem GAPSVP (with a $\mathrm{poly}(n)$ approximation factor) to BDD (to within distance $\lambda_1/\mathrm{poly}(n)$).[4] This is quite reassuring, as it tells us that as long as $\alpha q/r = \lambda_1(\Lambda)/\mathrm{poly}(n)$, the LWE problem is as hard as the variant of the worst-case lattice problem GAPSVP in which we are given samples $D_{\Lambda^*,r}$.

It would be even nicer if we could replace the somewhat unusual assumption regarding the discrete Gaussian samples with a more familiar one. This can be done using the sampling procedure of [GPV08] which roughly speaking, is able to efficiently produce such samples given a basis of Λ^* all of whose vectors are of length at most r. This leads to a hardness result for LWE based on the assumption that GAPSVP is hard even given an unusually good basis for it (see [Pei09a] for the exact statement). Alternatively, using the LLL algorithm [LLL82] we can efficiently produce a basis of Λ^* whose vectors are of length at most $2^n/\lambda_1(\Lambda)$; this implies that LWE for *exponential moduli* $q = 2^{O(n)}$ is as hard as the standard worst-case lattice problem GAPSVP.

The approach originally taken in [Reg05] is different. The main additional ingredient proved there is that for any $d > 0$, there is an efficient *quantum* reduction from the problem of sampling from $D_{\Lambda^*,\sqrt{n}/d}$ to the problem of solving BDD on Λ to within distance d.[5] The reduction is a relatively simple quantum procedure: by using a BDD oracle, one can create a quantum state corresponding to a periodic Gaussian distribution (see Figure 3), whose Fourier transform (which

[4] The basic idea of the reduction is the following. We check the BDD oracle's ability to correctly recover lattice points after adding to them a perturbation of norm d, for various values of d. For $d < \lambda_1/\mathrm{poly}(n)$ the BDD oracle must always succeed, by assumption. For $d > \sqrt{n}\lambda_1$, one can prove that the oracle must err with noticeable probability because there is not enough statistical information in the query to determine the original lattice point (a similar idea was used in [GG00]). This allows us to deduce a $\mathrm{poly}(n)$ approximation to λ_1, as desired.

[5] Essentially the same reduction (with an improved analysis) was used by Stehlé et al. [SSTX09] to establish their SIS to LWE reduction.

can be computed efficiently using the quantum Fourier transform) turns out to be exactly the quantum state corresponding to the distribution $D_{\Lambda^*,\sqrt{n}/d}$; the latter can be measured to produce a sample from $D_{\Lambda^*,\sqrt{n}/d}$, as desired.

By combining this quantum reduction with the proposition above, we can now show that solving LWE with a polynomial modulus q implies a quantum solution to standard worst-case lattice problems. More precisely, assume that $\alpha q \geq 2\sqrt{n}$. Start by creating a bunch of samples from $D_{\Lambda^*,r}$ for a large r (which, as mentioned above, one can do efficiently). Now apply the proposition to obtain a solution to BDD on Λ to within distance $\sqrt{2n}/r$. Next, apply the quantum reduction to obtain a bunch of samples from $D_{\Lambda^*,r/\sqrt{2}}$. Now repeat the above two steps using these new samples as input, leading to samples from $D_{\Lambda^*,r/2}$, $D_{\Lambda^*,r/2\sqrt{2}}$, etc. After a polynomial number of steps, we end up with samples from $D_{\Lambda^*,r'}$ for a small $r' = \mathrm{poly}(n)/\lambda_1(\Lambda)$. From this it is easy to solve GAPSVP by applying once more the first step above, and using the reduction from GAPSVP to BDD. Alternatively, by taking about n samples from $D_{\Lambda^*,r'}$, we obtain a solution to the worst-case lattice problem SIVP.

Proof of Proposition 2.1. We demonstrate the main idea of the proof with the lattice $\Lambda = \mathbb{Z}^n$. The general argument is essentially identical, but might be confusing to those not familiar with lattices, and is therefore omitted; see [Pei09a] for more details and [Reg05] for the full details. (The main disadvantage of our presentation below is that the BDD problem is trivial on $\mathbb{Z}^n$, so one should keep in mind that the use of $\mathbb{Z}^n$ is just in order to clarify the main ideas.)

Assume we are given a point $\mathbf{x}$ close to some unknown lattice point $\mathbf{v} \in \mathbb{Z}^n$. We will show below how to generate samples from the LWE distribution with secret $\mathbf{s} = \mathbf{v} \bmod q$. Using the LWE oracle, we can recover $\mathbf{s}$, which gives us the least significant digits of $\mathbf{v}$ in base q. Recovering the entire vector $\mathbf{v}$ can now be done using a straightforward reduction; namely, run the same process on the point $(\mathbf{x} - \mathbf{s})/q$ (which is close to the lattice point $(\mathbf{v} - \mathbf{s})/q \in \mathbb{Z}^n$) to recover the second digits of $\mathbf{v}$ in base q, etc.

The core of the proof is, therefore, in producing LWE samples with secret $\mathbf{s}$. This is done as follows. Take a sample $\mathbf{y}$ from $D_{\mathbb{Z}^n,r}$ (using the samples given to us as input), and output the pair

$$(\mathbf{a} = \mathbf{y} \bmod q, b = \lfloor \langle \mathbf{y}, \mathbf{x} \rangle \rceil \bmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

Since r is not too small, the distribution of $\mathbf{a}$ is essentially uniform over $\mathbb{Z}_q^n$, as required. To analyze the second coordinate, condition on any fixed value of $\mathbf{a}$, so now $\mathbf{y}$ is distributed according to a discrete Gaussian distribution on the set of points $q\mathbb{Z}^n + \mathbf{a}$. Notice that if $\mathbf{x} = \mathbf{v}$, then b is exactly $\langle \mathbf{a}, \mathbf{s} \rangle$, corresponding to LWE equations with no error. In the case that $\mathbf{x} = \mathbf{v} + \mathbf{e}$ for some vector $\mathbf{e}$ of norm at most $\alpha q/(\sqrt{2}r)$, we obtain an error term in the second coordinate of the form $\langle \mathbf{e}, \mathbf{y} \rangle$. Being the inner product of a fixed vector of norm at most $\alpha q/(\sqrt{2}r)$ with a discrete Gaussian vector of norm roughly r, this error term is essentially normally distributed with standard deviation at most roughly αq, as required.

The above description hides two technical details. The first is that the error term $\langle \mathbf{e}, \mathbf{y} \rangle$ still has some of the discrete structure of $\mathbf{y}$; in order to get a true normal distribution, one needs to add a small amount of extra normal noise, by taking $b = \lfloor \langle \mathbf{y}, \mathbf{x} \rangle + e \rceil \bmod q$ where e is chosen from a continuous normal distribution. (This is why we assumed the distance of $\mathbf{x}$ is at most $\alpha q/(\sqrt{2}r)$, and not $\alpha q/r$.) The second minor detail is that the amount of noise in the resulting LWE samples depends on the distance of $\mathbf{x}$ from the lattice, i.e., the closer $\mathbf{x}$ is, the less noise there is in the LWE samples. The problem with this is that our LWE oracle might only work correctly with its specified amount of noise. Luckily, the solution is very easy: we simply add some extra noise to b of various amounts, until we hit the right amount of noise, allowing us to recover $\mathbf{s}$. (This minor issue turns out to be much less benign in the case of ring-LWE; see Section 5 for details.)

Other implications. In addition to its role in hardness results, Proposition 2.1 has a nice implication regarding the complexity of the LWE problem [GPV08, ACPS09]: It says that from a given small fixed polynomial number of LWE samples with secret $\mathbf{s}$ one can generate an arbitrary number of further LWE samples that are 'as good as new', i.e., LWE samples that are independently distributed according to $A_{\mathbf{s},\chi}$ for a normal noise χ. This result explains why the hardness of the LWE problem is independent of the number of samples. In a high level, this result follows easily from the proof of Proposition 2.1; all one has to notice is that the LWE problem is itself an instance of the BDD problem, and hence the proof above converts it into a legal instance of LWE.[6]

Notice that in the argument above we never used any properties of the input error distribution, beyond it being small — the LWE samples produced by the reduction are still guaranteed to be properly distributed, with the usual normal distribution of the error. This observation indicates that LWE with a normal error distribution is 'LWE-complete': LWE instances with an arbitrary error distribution can be reduced to LWE with a normal error distribution.

3 Variants of the Problem

We start by showing a reduction from the (search version of) LWE problem to the decision problem of distinguishing between LWE samples and samples from the uniform distribution U on $\mathbb{Z}_q^n \times \mathbb{Z}_q$. Although quite simple, this reduction is extremely helpful in cryptographic applications. The proof below is taken from [Reg05], and is similar to a proof in [BFKL93].

Lemma 3.1 (Decision to Search)**.** *Let $n \geq 1$ be some integer, $2 \leq q \leq \mathrm{poly}(n)$ be a prime, and χ be some distribution on $\mathbb{Z}_q$. Assume we have access to a procedure W that for all $\mathbf{s}$ accepts with probability exponentially close to 1 on inputs from $A_{\mathbf{s},\chi}$ and rejects with probability exponentially close to 1 on inputs from U. Then, there exists an efficient algorithm W' that, given samples from $A_{\mathbf{s},\chi}$ for some unknown $\mathbf{s}$, outputs $\mathbf{s}$ with probability exponentially close to 1.*

Proof: Let us show how W' finds $s_1 \in \mathbb{Z}_q$, the first coordinate of $\mathbf{s}$. Finding the other coordinates is similar. For each $k \in \mathbb{Z}_q$, consider the following transformation. Given a pair $(\mathbf{a}, b)$ we output the pair $(\mathbf{a} + (r, 0, \ldots, 0), b + r \cdot k)$ where $r \in \mathbb{Z}_q$ is chosen uniformly at random. It is easy to see that if $k = s_1$ then this transformation takes $A_{\mathbf{s},\chi}$ to itself. Moreover, if $k \neq s_1$ then it takes $A_{\mathbf{s},\chi}$

[6] To be precise, this would lead to LWE instances whose number of variables has grown from the original n to the number of equations m. The proof in [GPV08, ACPS09] is slightly different and uses the observation that the BDD instance corresponding to the input LWE instance is on a lattice that contains $q\mathbb{Z}^m$ as a sublattice, and our task is to find which of the q^n cosets of $q\mathbb{Z}^m$ contains the nearest lattice point.

to the uniform distribution (note that this requires q to be prime). Hence, using W, we can test whether $k = s_1$. Since there are only $q < \text{poly}(n)$ possibilities for s_1 we can try all of them. ∎

In [Pei09a], this lemma was extended to the case that q is a product of distinct primes, each of which is large enough but still polynomial, under the assumption that the noise distribution is normal. The idea is to find the value of s_1 modulo each of the prime factor of q using a transformation similar to the one above. As in the above proof, if our guess is correct, the LWE distribution is taken to itself. If, on the other hand, our guess is incorrect then, using the assumption that the noise distribution is normal, one can prove that the resulting distribution is extremely close to uniform.

We next show that solving the decision problem with a non-negligible probability over a *uniform choice* of secret $\mathbf{s} \in \mathbb{Z}_q^n$ suffices to solve the problem for *all* secrets $\mathbf{s}$.

Lemma 3.2 (Average-case to Worst-case)**.** *Let $n, q \geq 1$ be some integers and χ be some distribution on $\mathbb{Z}_q$. Assume that we have access to a distinguisher W that distinguishes $A_{\mathbf{s},\chi}$ from U for a non-negligible fraction of all possible $\mathbf{s}$. Then there exists an efficient algorithm W' that for* all $\mathbf{s}$ *accepts with probability exponentially close to* 1 *on inputs from $A_{\mathbf{s},\chi}$ and rejects with probability exponentially close to* 1 *on inputs from U.*

Proof: The proof is based on the following transformation. For any $\mathbf{t} \in \mathbb{Z}_q^n$ consider the function $f_{\mathbf{t}} : \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{Z}_q^n \times \mathbb{Z}_q$ defined by

$$f_{\mathbf{t}}(\mathbf{a}, b) = (\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle).$$

It is easy to see that this function transforms the distribution $A_{\mathbf{s},\chi}$ into $A_{\mathbf{s}+\mathbf{t},\chi}$. Moreover, it transforms the uniform distribution U into itself.

The algorithm W' repeats the following a large enough $\text{poly}(n)$ times. Choose a vector $\mathbf{t} \in \mathbb{Z}_q^n$ uniformly at random, and estimate to within $\pm 1/\text{poly}(n)$ the acceptance probability of W, both on U and on the distribution obtained by applying $f_{\mathbf{t}}$ to our input distribution (by simply applying W to each distribution $\text{poly}(n)$ times). If the two estimates differ by a noticeable amount, stop and accept. Otherwise, repeat with another $\mathbf{t}$. If, no noticeable difference is ever observed, reject.

To prove correctness, observe that if the input distribution is U then for each $\mathbf{t}$, both our estimates are of exactly the same quantity, hence (by setting parameters correctly) the probability we accept is exponentially small. If, on the other hand, the input distribution is $A_{\mathbf{s},\chi}$ then with high probability in one of our attempts we will hit a $\mathbf{t}$ for which W distinguishes $A_{\mathbf{s}+\mathbf{t},\chi}$ from U, in which case we'll accept with probability exponentially close to 1. ∎

Two recent results show further ways to modify the distribution of secrets $\mathbf{s}$ in the LWE problem, without compromising its hardness. The first result, shown by Applebaum, Cash, Peikert, and Sahai [ACPS09], says that we can choose the coordinates of the secret $\mathbf{s}$ from the same distribution as that of the noise distribution χ. (This result does not follow from the previous lemma since χ^n typically has a negligible 'mass' under the uniform distribution.) Roughly speaking, their proof is based on the observation that any set of n (linearly independent) approximate equations essentially give us an approximation of $\mathbf{s}$ up to the noise distribution, so by subtracting that approximation from the LWE secret, we obtain an LWE instance whose secret is distributed like the noise distribution. More precisely, assume we are given samples from the distribution $A_{\mathbf{s},\chi}$ for some arbitrary unknown $\mathbf{s} \in \mathbb{Z}_q^n$. Take n samples from the distribution, and write them as $\bar{\mathbf{b}} = \bar{\mathbf{A}}^T\mathbf{s} + \bar{\mathbf{x}}$ where each coordinate of (the unknown) $\bar{\mathbf{x}} \in \mathbb{Z}_q^n$ is chosen independently from χ. Assume for simplicity that $\bar{\mathbf{A}}$ is invertible. (This happens with good probability; alternatively, we can accumulate samples until we obtain an invertible matrix). Now, it is easy to check that by replacing each sample $(\mathbf{a}, b)$ from $A_{\mathbf{s},\chi}$ with $(-\bar{\mathbf{A}}^{-1}\mathbf{a}, b - \langle \bar{\mathbf{A}}^{-1}\mathbf{a}, \bar{\mathbf{b}} \rangle)$, we obtain samples from the distribution $A_{\bar{\mathbf{x}},\chi}$. Noticing that this transformation maps the uniform distribution to itself completes the proof.

A second related result, which we will not describe in detail, was recently shown by Goldwasser, Kalai, Peikert, and Vaikuntanathan [GKPV10]. Their result is much more general than the two described above: it essentially shows that as long as the secret is chosen from a distribution whose entropy is not too small, the LWE problem remains hard. There are, however, some subtleties, most notably the fact that the hardness of this restricted-secret variant of LWE is based on the hardness of (standard) LWE whose noise parameter α is taken to be negligibly small (which, in turn, is as hard as approximating worst-case lattice problems to within super-polynomial factors). Strengthening this to a hardness result based on LWE with the usual setting of $\alpha = 1/\mathrm{poly}(n)$ is an interesting open question.

4 Cryptographic Applications

The range of cryptographic applications of the LWE problem has by now become very wide, and it would be outside the scope of this survey to describe all known applications. Instead, we choose to include one very simple cryptographic application, in order to give a taste of this area, and in order to motivate the rest of this paper, especially Section 5. We emphasize that the example below is quite inefficient; for more efficient schemes, see, e.g., [PVW08, MR08]. Moreover, using the ring-LWE problem described in Section 5, the system can be made truly practical [LPR10].

Our cryptosystem is parameterized by integers n (the security parameter), m (number of equations), q (modulus), and a real $\alpha > 0$ (noise parameter). One possible choice that guarantees both security and correctness is the following. Choose q to be a prime between n^2 and $2n^2$, $m = 1.1 \cdot n \log q$, and $\alpha = 1/(\sqrt{n} \log^2 n)$. In the following description, all additions are performed modulo q.

- **Private key:** The private key is a vector $\mathbf{s}$ chosen uniformly from $\mathbb{Z}_q^n$.
- **Public Key:** The public key consists of m samples $(\mathbf{a}_i, b_i)_{i=1}^m$ from the LWE distribution with secret $\mathbf{s}$, modulus q, and error parameter α.
- **Encryption:** For each bit of the message, do the following. Choose a random set S uniformly among all 2^m subsets of $[m]$. The encryption is $(\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i)$ if the bit is 0 and $(\sum_{i \in S} \mathbf{a}_i, \lfloor \frac{q}{2} \rfloor + \sum_{i \in S} b_i)$ if the bit is 1.
- **Decryption:** The decryption of a pair $(\mathbf{a}, b)$ is 0 if $b - \langle \mathbf{a}, \mathbf{s} \rangle$ is closer to 0 than to $\lfloor \frac{q}{2} \rfloor$ modulo q, and 1 otherwise.

As promised, this system is quite inefficient: its public key size is $O(mn \log q) = \tilde{O}(n^2)$ and encryption increases the size of a message by a factor of $O(n \log q) = \tilde{O}(n)$.

Correctness. Note that if not for the error in the LWE samples, $b - \langle \mathbf{a}, \mathbf{s} \rangle$ would be either 0 or $\lfloor \frac{q}{2} \rfloor$ depending on the encrypted bit, and decryption would always be correct. Hence we see that a decryption error occurs only if the sum of the error terms over all S is greater than $q/4$. Since we are summing at most m normal error terms, each with standard deviation αq, the standard deviation of the sum is at most $\sqrt{m}\alpha q < q/\log n$; a standard calculation shows that the probability that such a normal variable is greater than $q/4$ is negligible.

Security. We now sketch the security proof, showing that the system is secure based on the LWE assumption. Assume that the system is not secure against chosen plaintext attacks; i.e., that there exists an efficient algorithm that given a public key $(\mathbf{a}_i, b_i)_{i=1}^m$ chosen from the LWE distribution with some secret $\mathbf{s}$ and an encryption of a random bit generated as above, can correctly guess the encrypted bit with probability at least $1/2 + 1/\text{poly}(n)$ for a non-negligible fraction of secrets $\mathbf{s}$.

Now imagine we provide the algorithm with pairs $(\mathbf{a}_i, b_i)_{i=1}^m$ chosen uniformly from $\mathbb{Z}_q^n \times \mathbb{Z}_q$ (instead of from the LWE distribution), and a random bit encrypted as before using these pairs (as if they were a public key). It follows from the leftover hash lemma [IZ89] (or alternatively, an argument based on Fourier analysis as in [NS99]) that with very high probability over the choice of $(\mathbf{a}_i, b_i)_{i=1}^m$, the distribution (over S) of a random subset sum $(\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i)$ is extremely close to uniform in statistical distance. Intuitively, this follows because the number of possible subsets is 2^m, which is much larger than the number of possible pairs, q^{n+1}. As a result, encryptions of 0 and of 1 are essentially identically distributed and the algorithm simply has no way to guess the encrypted bit beyond random guessing.

Therefore, by checking the algorithm's ability to correctly guess the value of an encrypted random bit, we can distinguish LWE samples from uniform samples for a non-negligible fractions of secrets $\mathbf{s}$. Using Lemmas 3.1 and 3.2, this implies a solution to the LWE problem, and we are done.

5 Ring-LWE

As described in the introduction, the ring-LWE problem holds a great promise to make lattice-based cryptography truly practical, as well as to lead to theoretical advances in the area. In this section we describe some of the recent work on establishing results for ring-LWE that are analogous to those known for LWE [LPR10].

First, let us start by giving a slightly more convenient, yet equivalent, definition of the ring-LWE problem. Let n be a power of two, and let q be a prime modulus satisfying $q = 1 \bmod 2n$. Define R_q as the ring $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ containing all polynomials over the field $\mathbb{Z}_q$ in which x^n is identified with -1. In ring-LWE we are given samples of the form $(\mathbf{a}, \mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}) \in R_q \times R_q$ where $\mathbf{s} \in R_q$ is a fixed secret, $\mathbf{a} \in R_q$ is chosen uniformly, and $\mathbf{e}$ is an error term chosen independently from some error distribution over R_q. The most natural choice of error distribution is to choose the coordinates of the polynomial $\mathbf{e}$ to be i.i.d. normal, i.e., a spherical Gaussian distribution. Our goal is to recover the secret $\mathbf{s}$ from these samples (for all $\mathbf{s}$, with high probability). This definition can be extended to somewhat more general rings, but for simplicity we focus on the choice above.

A hardness result for the above search version of ring-LWE was established in [LPR10]. The proof follows the same outline as that in [Reg05], namely, proving an analogue of Proposition 2.1 for ideal lattices, and then combining it with the quantum reduction from Gaussian sampling to BDD, leading to a solution to worst-case problems like SIVP on ideal lattices. Luckily, the quantum reduction of [Reg05] can be used nearly as is; the main effort is in adapting Proposition 2.1.

One of the main issues that arise during this adaptation is the error distribution. Recall that in the proof of Proposition 2.1, the error distribution in the LWE samples we produce is a normal variable whose variance depends on the norm of the offset vector $\mathbf{e}$. Since the error is specified by just one real parameter (the variance), we could easily take care of this issue by adding varying amounts of extra normal error, until we 'hit' the amount of error for which the oracle is promised to output the solution of the LWE instance. In the case of ring-LWE, because we replace the inner product $\langle \mathbf{e}, \mathbf{y} \rangle$ with a ring product, the error distribution turns out to be a (not necessarily spherical) Gaussian distribution whose variances depend on the entire vector $\mathbf{e}$. Because the error distribution has n parameters (and not one as before), we cannot hope to hit any fixed noise distribution by simply adding more noise. Instead, we must assume that the ring-LWE oracle works for a whole range of noise parameters (or some distribution on such parameters). The upshot of this is that the hardness of ring-LWE obtained in [LPR10] only applies if the error distribution is itself chosen from a certain distribution on non-spherical Gaussian variables. While somewhat undesirable, luckily this issue does not seem to cause any trouble in most of the applications, and we will therefore ignore it in the sequel, instead assuming for simplicity that the error is taken from a spherical Gaussian distribution. We remark that a hardness result for the spherical error case is shown in [SSTX09] (or can be derived from [LPR10]), but unfortunately it depends on the number of samples m.

In the rest of this section we sketch the reduction from the search version of ring-LWE to the decision problem of distinguishing valid ring-LWE samples from uniform samples in $R_q \times R_q$, which is the problem most suited for cryptographic applications. Our description hides many technical (but crucial) steps in this reduction, such as amplification, and worst-case to average-case reductions; see [LPR10] for the full details. Before we go on, the reader might wish to reread the proof of the analogous statement for LWE, as given in Lemma 3.1. To recall, the main idea there was to guess a part of the secret (namely, the first coordinate) and to modify the LWE samples in such a way that: (1) a correct guess keeps the distribution as is, and (2) an incorrect guess transforms it into the uniform distribution.

Let us try to apply the same idea to ring-LWE, and for now just focus on requirement (1). Given a sample $(\mathbf{a}, \mathbf{b}) \in R_q \times R_q$, we would like to somehow sample an $\mathbf{r} \in R_q$ and output a pair $(\mathbf{a} + \mathbf{r}, \mathbf{b} + \mathbf{k})$ where $\mathbf{k} \in R_q$ is our guess for $\mathbf{r} \cdot \mathbf{s}$. However, $\mathbf{r} \cdot \mathbf{s}$ might take q^n possible values, so how can we hope to guess it?

In order to solve this problem, we need to use some basic facts regarding the ring R_q. First, notice that if $t \in \mathbb{Z}_q$ is such that $t^n = -1$ (i.e., a root of $x^n + 1$), then for any elements $\mathbf{p}_1, \mathbf{p}_2 \in R_q$

we have $\mathbf{p}_1(t) \cdot \mathbf{p}_2(t) = (\mathbf{p}_1 \cdot \mathbf{p}_2)(t)$, i.e., the mapping that takes each polynomial $\mathbf{p} \in R_q$ to its evaluation $\mathbf{p}(t) \in \mathbb{Z}_q$ is a ring homomorphism. Next, notice that since $q = 1 \bmod 2n$, the polynomial $x^n + 1$ has all n roots in the field $\mathbb{Z}_q$. Namely, if g is a generator of the multiplicative group, then the roots are

$$t_1 = g^{(q-1)/2n}, t_3 = g^{3(q-1)/2n}, \ldots, t_{2n-1} = g^{(2n-1)(q-1)/2n}.$$

We can now define the mapping $\varphi : R_q \to \mathbb{Z}_q^n$ that maps each $\mathbf{p} \in R_q$ to $(\mathbf{p}(t_1), \mathbf{p}(t_3) \ldots, \mathbf{p}(t_{2n-1})) \in \mathbb{Z}_q^n$. Using our previous observation, this mapping is a ring homomorphism, with the operations in $\mathbb{Z}_q^n$ being coordinate-wise. Moreover, this mapping is actually a ring isomorphism, since given any element in $\mathbb{Z}_q^n$ we can find using interpolation a degree $n-1$ polynomial mapped to it. To summarize, the isomorphism φ allows us to think of R_q equivalently as the ring $\mathbb{Z}_q^n$ with coordinate-wise addition and multiplication.

The solution to our problem above should now be clear. Choose $\mathbf{r} = \varphi^{-1}(r, 0, \ldots, 0)$ for a random $r \in \mathbb{Z}_q$, and then transform each sample $(\mathbf{a}, \mathbf{b})$ to $(\mathbf{a} + \mathbf{r}, \mathbf{b} + \varphi^{-1}(r \cdot k, 0, \ldots, 0))$ where k is our guess for the first coordinate of $\varphi(\mathbf{s})$. The point is that by taking $\mathbf{r}$ to be of the above form, we managed to restrict the number of possible values of $\mathbf{r} \cdot \mathbf{s}$ to only q, hence allowing us to efficiently enumerate all possible values.

There is still one important piece missing from the above description. While our transformation satisfies requirement (1) above, it does not satisfy requirement (2). Indeed, when our guess k is incorrect, the distribution we obtain is far from uniform: if $(\mathbf{a}, \mathbf{b})$ is a sample from that distribution, then the first coordinate of $\varphi(\mathbf{b})$ is uniform, but the remaining $n-1$ coordinates are distributed as in the original ring-LWE samples. The problem with this is that the decision oracle might behave on such a distribution in exactly the same way as on the original ring-LWE distribution, thereby not allowing us to check if our guess k is correct.

We solve this problem using a hybrid argument, as follows. For $i \in \{0, \ldots, n\}$, consider the distribution obtained from the input samples $(\mathbf{a}, \mathbf{b})$ by replacing the first i coordinate of $\varphi(\mathbf{b})$ with uniform values. This sequence interpolates between the original ring-LWE distribution ($i = 0$) and the completely uniform distribution ($i = n$). Since by assumption our oracle is able to distinguish these two cases, there must exist an i for which the oracle distinguishes the $i-1$st distribution from the ith distribution. Using an argument similar to the one above, we are now able to correctly recover the ith coordinate of $\varphi(\mathbf{s})$ (the only extra step being to make coordinates 1 through $i-1$ uniform by simply adding to the $\mathbf{b}$ component an element whose first $i-1$ coordinates are chosen uniformly).

So we are now able to recover the ith coordinate of $\varphi(\mathbf{s})$, but unfortunately i is determined by the oracle, over which we have no control. How can we recover all the coordinate of $\varphi(\mathbf{s})$? One thing we can try to do is permute the coordinates of $\varphi(\mathbf{s})$. More precisely, for any permutation $\pi : [n] \to [n]$, let $\tau_\pi : R_q \to R_q$ be the operation that permutes the coordinates (in the $\mathbb{Z}_q^n$ representation) according to π. Since multiplication and addition in $\mathbb{Z}_q^n$ are coordinate-wise, τ_π is clearly an automorphism. So now assume we transform each input sample $(\mathbf{a}, \mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e})$ to

$$(\tau_\pi(\mathbf{a}), \tau_\pi(\mathbf{b}) = \tau_\pi(\mathbf{a}) \cdot \tau_\pi(\mathbf{s}) + \tau_\pi(\mathbf{e})).$$

These samples look a lot like legal ring-LWE samples with the secret $\tau_\pi(\mathbf{s})$. If this were really the case, we would be done, as we could recover the ith coordinate of $\tau_\pi(\mathbf{s})$, from which we can recover all of $\mathbf{s}$ by using several different permutations. However, applying τ_π to $\mathbf{e}$ can completely ruin the error distribution, which, to recall, was defined as a Gaussian distribution of the polynomial's coefficients, and hence is likely to be quite 'strange' when viewed in the $\mathbb{Z}_q^n$ representation. Luckily, it turns out that certain permutations π leave the error distribution invariant. Namely, for each $j \in \{1, 3, \ldots, 2n-1\}$, consider the permutation π_j that sends the coordinate corresponding to t_i to that corresponding to $t_{i \cdot j^{-1} \bmod 2n}$. It is not difficult to see that when viewed in R_q, the operation τ_{π_j} can be described as taking each polynomial $\mathbf{p} = \mathbf{p}(x) \in R_q$ and mapping it to the polynomial $\mathbf{p}(x^j) \in R_q$. Hence, all τ_{π_j} does is simply permute the coefficients of the polynomial and possibly negate some, and in particular, it preserves the Gaussian error distribution. Using these n permutations, we can now recover all n coordinates of $\varphi(\mathbf{s})$, and complete the proof.

The reduction above might seem mysterious and ad-hoc: where do φ and τ come from? how come the τ_{π_j} preserve the error distribution? The answer is that underlying the entire reduction are some algebraic structures, such as the cyclotomic number field $\mathbb{Q}(\zeta_{2n})$, its canonical embedding, its n automorphisms, and the factorization of the ideal $\langle q \rangle$ into ideals of small norm. Viewed this way, the reduction easily extends to all cyclotomic polynomials (and not just $x^n + 1$), as was done in [LPR10].

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*. 2010.

[ACPS09] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618. 2009.

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th Annual ACM Symp. on Theory of Computing (STOC)*, pages 284–293. 1997.

[AD07] M. Ajtai and C. Dwork. The first and fourth public-key cryptosystems with worst-case/average-case equivalence, 2007. Available from ECCC at `http://www.uni-trier.de/eccc/`.

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

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 1–32. Dept. Math., Seconda Univ. Napoli, Caserta, 2004. Preliminary version in STOC 1996.

[Ajt05] M. Ajtai. Representing hard lattices with $O(n \log n)$ bits. In *Proc. 37th Annual ACM Symp. on Theory of Computing (STOC)*, pages 94–103. 2005.

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd Annual ACM Symp. on Theory of Computing (STOC)*, pages 601–610. 2001.

[Ale03] M. Alekhnovich. More on average case vs approximation complexity. In *Proc. 44th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 298–307. 2003.

[AR04] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. *Journal of the ACM*, 52(5):749–765, 2005. Preliminary version in FOCS'04.

[BFKL93] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO*, pages 278–291. 1993.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*. 2010.

[DGK$^+$10] Y. Dodis, S. Goldwasser, Y. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*. 2010.

[Fei02] U. Feige. Relations between average case complexity and approximation complexity. In *Proc. 34th Annual ACM Symp. on Theory of Computing (STOC)*, pages 534–543. 2002.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. 2009.

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

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

[GGH97] O. Goldreich, S. Goldwasser, and S. Halevi. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In *CRYPTO*, volume 1294 of *Lecture Notes in Comput. Sci.*, pages 105–111. 1997.

[GKPV10] S. Goldwasser, Y. Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*. 2010.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. 40th ACM Symp. on Theory of Computing (STOC)*, pages 197–206. 2008.

[GRS08] H. Gilbert, M. J. B. Robshaw, and Y. Seurin. How to encrypt with the LPN problem. In *ICALP*, pages 679–690. 2008.

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

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288. 1998.

[IZ89] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proc. 30th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 248–253. 1989.

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

[KS06] A. R. Klivans and A. A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. System Sci.*, 75(1):2–12, 2009. Preliminary version in FOCS'06.

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

[KTX08] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, pages 372–389. 2008.

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

[LLM06] Y.-K. Liu, V. Lyubashevsky, and D. Micciancio. On bounded distance decoding for general lattices. In *RANDOM*, pages 450–461. 2006.

[LM06] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP*, pages 144–155. 2006.

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

[LM09] V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *CRYPTO*, pages 577–594. 2009.

[LMPR08] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In *FSE*, pages 54–72. 2008.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*. 2010.

[Lyu05] V. Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *RANDOM*, pages 378–389. 2005.

[Lyu08] V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography*, pages 162–179. 2008.

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616. 2009.

[Mic02] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Preliminary version in FOCS 2002.

[Mic07] D. Micciancio. Cryptographic functions from worst-case complexity assumptions. In P. Q. Nguyen and B. Vallée, editors, *The LLL Algorithm: Survey and Applications*, Information Security and Cryptography, pages 427–452. Springer, 2008. Prelim. version in LLL25, 2007.

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[MR08] D. Micciancio and O. Regev. Lattice-based cryptography. In D. J. Bernstein and J. Buchmann, editors, *Post-quantum Cryprography*. Springer, 2008.

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

[MV10] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In *STOC*. 2010.

[NS99] P. Q. Nguyen and J. Stern. The hardness of the hidden subset sum problem and its cryptographic implications. In *CRYPTO*, pages 31–46. 1999.

[Pei09a] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. 41st ACM Symp. on Theory of Computing (STOC)*, pages 333–342. 2009.

[Pei09b] C. Peikert. Some recent progress in lattice-based cryptography. Slides for invited tutorial at TCC'09, 2009.

[PR06] C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, pages 145–166. 2006.

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

[PW08] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc. 40th ACM Symp. on Theory of Computing (STOC)*, pages 187–196. 2008.

[Reg03] O. Regev. New lattice-based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, 2004. Preliminary version in STOC'03.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34, 2009. Preliminary version in STOC'05.

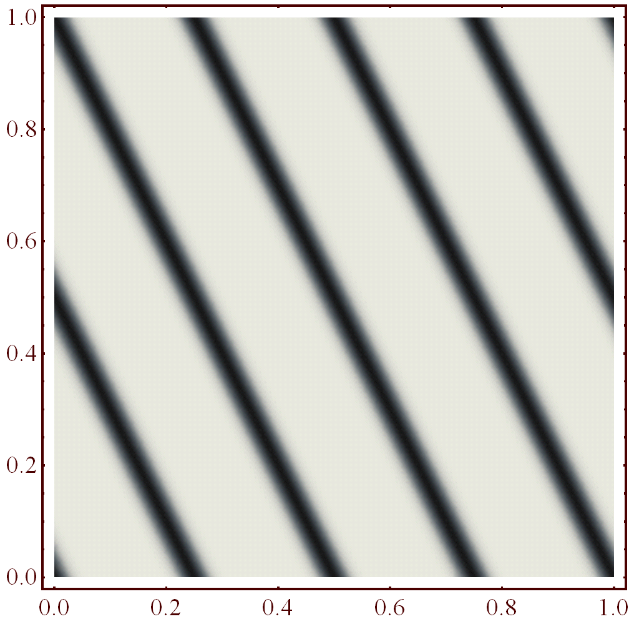

Figure 4: A two-dimensional wavy distribution with $\mathbf{s} = (4, 2)$.

[Reg06] O. Regev. Lattice-based cryptography. In *CRYPTO*, pages 131–141. 2006.

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, pages 617–635. 2009.

A The hardness of LWE and Ajtai-Dwork

Here we observe that a certain hardness result for the LWE problem is already implicit in Ajtai and Dwork's work [AD97] as presented in its simplified form [Reg03]. Even though the hardness results for LWE described in the body of this survey seem to us superior in all respects, we decided to include this observation for its historical value, and because we view it as an indication that LWE is the correct unifying idea behind all work on lattice-based public-key cryptography.

For $\mathbf{s} \in \mathbb{Z}^n$ define the 'wavy' distribution with parameter $\mathbf{s}$ as the distribution on $[0, 1)^n$ in which the probability of obtaining $\mathbf{x}$ (or more precisely, its probability density) is proportional to $\exp(-(n \cdot \mathrm{dist}(\langle \mathbf{x}, \mathbf{s} \rangle, \mathbb{Z}))^2)$ where $\mathrm{dist}(\cdot, \mathbb{Z})$ denotes the distance from the nearest integer (see Figure 4). So vectors $\mathbf{x} \in [0, 1)^n$ for which $\langle \mathbf{x}, \mathbf{s} \rangle$ is integer are most likely to appear, and typical vectors $\mathbf{x}$ from the distribution will have $\langle \mathbf{x}, \mathbf{s} \rangle$ that is within $\pm \frac{1}{n}$ from an integer. In other words, one can think of each sample $\mathbf{x}$ as an approximate equation

$$s_1 x_1 + \cdots + s_n x_n \approx 0 \pmod 1$$

with normal error.

In the next-to-last step of the chain of reduction in [Reg03], the following problem was shown to be as hard as the worst-case lattice problem unique-SVP (and thanks to the reductions in [Pei09a, LM09], also as hard as GAPSVP): given an unlimited number of samples from the wavy distribution with an unknown parameter $\mathbf{s} \in \{-2^n, \ldots, 2^n\}^n \setminus \{0^n\}$, find $\mathbf{s}$. In fact, the hardness result in [Reg03] even applies to the *decision* version of distinguishing wavy distributions from the uniform distribution over $[0,1)^n$, and as a result requires more effort to prove; for completeness, at the end of this section we sketch the hardness proof of the search version.

We now sketch a reduction from the above problem to LWE by showing how to convert samples from the wavy distribution to LWE samples. Assume we know an index i for which $s_i \neq 0$; the reduction can obtain such an i be trying all n indices. Moreover, assume without loss of generality that $i = n$. Then it follows from the definition of the wavy distribution that $x_1, \ldots, x_{n-1}$ are uniformly distributed, and conditioned on any fixed value of them, x_n is distributed according to a 'one-dimensional wavy distribution' with s_n periods, whose phase is shifted by $s_1 x_1 + \cdots + s_{n-1} x_{n-1}$ mod 1 (to see this, imagine taking vertical slices of Figure 4). Next, assume we have an estimate s'_n of s_n to within a multiplicative error of at most $1 \pm 1/n^c$ for a large enough c. The reduction can obtain such an estimate by trying all values on an exponential scale. Consider now the distribution of $(\mathbf{a} = (x_1, \ldots, x_{n-1}, x_n + t/s'_n), b = t)$ where $\mathbf{x}$ is a sample from the wavy distribution and t is uniformly chosen in $[0,1)$. Then it is not hard to show that this distribution is within statistical distance about n^{-c} of distribution on pairs $(\mathbf{a}, b)$ obtained by choosing $\mathbf{a}$ uniformly in $[0,1)^n$ and then choosing b from a normal variable centered around $\langle \mathbf{a}, \mathbf{s} \rangle$ mod 1. Now let $\mathbf{a}' = \lfloor q\mathbf{a} \rceil$ and similarly $b' = \lfloor qb \rceil$. If q is large enough, say, 2^{2n}, then the error introduced by rounding is negligible, and the pairs $(\mathbf{a}', b')$ are essentially distributed like valid LWE samples, as required.

For completeness, let us briefly sketch the idea of the reduction in [Reg03]. Assume $\mathbf{v}$ is the unique shortest vector of the input lattice $\mathcal{L}(\mathbf{B})$. Let $\mathbf{D}$ be the dual basis to $\mathbf{B}$. By definition, all vectors $\mathbf{u}$ in the dual lattice $\mathcal{L}(\mathbf{B})^* = \mathcal{L}(\mathbf{D})$ satisfy $\langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z}$, i.e., all vectors in the dual lattice are contained in a set of hyperplanes orthogonal to $\mathbf{v}$ and spaced $1/\|\mathbf{u}\|$ from each other. Moreover, since the vector $\mathbf{u}$ is a unique shortest vector, the set of lattice points inside each such hyperplane is quite dense. The main idea of the reduction is that by taking a random point in $\mathcal{L}(\mathbf{D})$ and adding Gaussian noise to it, we eliminate the fine structure inside the hyperplanes, and end up with a distribution that is essentially uniform on any hyperplane orthogonal to $\mathbf{v}$ and looks like a periodic Gaussian in the direction of $\mathbf{v}$. By considering the coefficients modulo 1 in the basis $\mathbf{D}$ of a point chosen from this distribution, we end up with the wavy distribution whose parameter is the vector of coefficients of $\mathbf{v}$ in the basis $\mathbf{B}$.