On Lattices, Learning with Errors, Random Linear Codes, and Cryptography

Oded Regev *

November 4, 2004

Abstract

Our main result is a reduction from worst-case lattice problems such as SVP and SIVP to a certain learning problem. This learning problem is a natural extension of the 'learning from parity with error' problem to higher moduli. It can also be viewed as the problem of decoding from a random linear code. This, we believe, gives a strong indication that these problems are hard. Our reduction, however, is quantum. Hence, an efficient solution to the learning problem implies a *quantum* algorithm for SVP and SIVP. A main open question is whether this reduction can be made classical.

Using the main result, we obtain a public-key cryptosystem whose hardness is based on the worst-case quantum hardness of SVP and SIVP. Previous lattice-based public-key cryptosystems such as the one by Ajtai and Dwork were only based on unique-SVP, a special case of SVP. The new cryptosystem is much more efficient than previous cryptosystems: the public key is of size $\tilde{O}(n^2)$ and encrypting a message increases its size by $\tilde{O}(n)$ (in previous cryptosystems these values are $\tilde{O}(n^4)$ and $\tilde{O}(n^2)$, respectively).

1 Introduction

Main theorem. Let n be some integer and let $\varepsilon \geq 0$ be some real. Consider the 'learning from parity with error' problem, defined as follows: find $\mathbf{s} \in \mathbb{Z}_2^n$ given a list of 'equations with errors'

$$\begin{aligned} \langle \mathbf{s}, \mathbf{a}_1 \rangle &\approx_\varepsilon b_1 \pmod 2 \\ \langle \mathbf{s}, \mathbf{a}_2 \rangle &\approx_\varepsilon b_2 \pmod 2 \\ &\vdots \end{aligned}$$

where the $\mathbf{a}_i$'s are chosen independently from the uniform distribution on $\mathbb{Z}_2^n$ and $\langle \mathbf{s}, \mathbf{a}_i \rangle = \sum_j s_j (a_i)_j$ is the inner product modulo 2 of $\mathbf{s}$ and $\mathbf{a}_i$. The input to the problem consists of the pairs $(\mathbf{a}_i, b_i)$ and the output is a guess for $\mathbf{s}$. By the $\approx_\varepsilon$ symbol we mean that each equation is independently chosen to be correct with probability $1 - \varepsilon$ and incorrect with probability ε. Notice that the case $\varepsilon = 0$ can be solved efficiently by, say, Gaussian elimination. This requires $O(n)$ equations and $\mathrm{poly}(n)$ time.

The problem seems to become significantly harder when we take any positive $\varepsilon > 0$. For example, let us consider again the Gaussian elimination process and assume we are interested in recovering only the first

*Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Supported by an Alon Fellowship, by the Israel Science Foundation, and by the Army Research Office grant DAAD19-03-1-0082.

bit of $\mathbf{s}$. Using Gaussian elimination, we can find a set S of $O(n)$ equations such that $\sum_S \mathbf{a}_i$ is $(1, 0, \ldots, 0)$. Summing the corresponding values b_i gives us a guess for the first bit of $\mathbf{s}$. However, a standard calculation shows that this guess is correct with probability $\frac{1}{2} + 2^{-\Theta(n)}$. Hence, in order to obtain the first bit with good confidence, we have to repeat the whole procedure $2^{\Theta(n)}$ times. This yields an algorithm that uses $2^{O(n)}$ equations and $2^{O(n)}$ time. In fact, it can be shown that given only $O(n)$ equations, the $\mathbf{s}' \in \mathbb{Z}_2^n$ that maximizes the number of satisfied equations is with high probability $\mathbf{s}$. This yields a simple maximum likelihood algorithm that requires only $O(n)$ equations and runs in time $2^{O(n)}$.

Blum, Kalai, and Wasserman [9] provided the first subexponential algorithm for this problem. Their algorithm requires only $2^{O(n/\log n)}$ equations/time and is currently the best known algorithm for the problem. It is based on a clever idea that allows to find a small set S of equations (say, $O(\sqrt{n})$) among $2^{O(n/\log n)}$ equations, such that $\sum_S \mathbf{a}_i$ is, say, $(1, 0, \ldots, 0)$. This gives us a guess for the first bit of $\mathbf{s}$ that is correct with probability $\frac{1}{2} + 2^{-\Theta(\sqrt{n})}$. We can obtain the correct value with high probability by repeating the whole procedure only $2^{O(\sqrt{n})}$ times. Their algorithm was later shown to have other important applications, such as the first $2^{O(n)}$-time algorithm for solving the shortest vector problem in a lattice [13, 4].

An important open question is to explain the apparent difficulty in finding efficient algorithms for this learning problem. Our main theorem explains this difficulty for a natural extension of this problem to higher moduli, defined next.

Let $p = p(n) \leq \mathrm{poly}(n)$ be some prime integer and consider a list of 'equations with error'

$$\begin{aligned} \langle \mathbf{s}, \mathbf{a}_1 \rangle &\approx_\chi b_1 \pmod p \\ \langle \mathbf{s}, \mathbf{a}_2 \rangle &\approx_\chi b_2 \pmod p \\ &\vdots \end{aligned}$$

where this time $\mathbf{s} \in \mathbb{Z}_p^n$, $\mathbf{a}_i$ are chosen independently and uniformly from $\mathbb{Z}_p^n$, and $b_i \in \mathbb{Z}_p$. The error in the equations is now specified by a probability distribution $\chi : \mathbb{Z}_p \to \mathbb{R}^+$ on $\mathbb{Z}_p$. Namely, for each equation i, $b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i$ where each $e_i \in \mathbb{Z}_p$ is chosen independently according to χ. We denote the problem of recovering $\mathbf{s}$ from such equations by $\mathsf{LWE}_{p,\chi}$ (learning with error). For example, the learning from parity problem with error ε is the special case where $p = 2$, $\chi(0) = 1 - \varepsilon$, and $\chi(1) = \varepsilon$. Under a reasonable assumption on χ (namely, that $\chi(0) > 1/p + 1/\mathrm{poly}(n)$), the maximum likelihood algorithm described above solves $\mathsf{LWE}_{p,\chi}$ for $p \leq \mathrm{poly}(n)$ using $\mathrm{poly}(n)$ equations and $2^{O(n \log n)}$ time. Under a similar assumption, an algorithm resembling the one by Blum et al. [9] requires only $2^{O(n)}$ equations/time. This is the best known algorithm for the LWE problem.

Our main theorem shows that for certain choices of p and χ, a solution to $\mathsf{LWE}_{p,\chi}$ implies a quantum solution to worst-case lattice problems.

Theorem 1.1 (Informal) *Let n, p be integers and $\alpha \in (0, 1)$ be some real such that $\alpha p > 2\sqrt{n}$. If there exists a polynomial time algorithm that solves $\mathsf{LWE}_{p,\bar{\Psi}_\alpha}$ then there exists a quantum algorithm that approximates the shortest vector problem (SVP) and the shortest independent vectors problem (SIVP) to within $\tilde{O}(n/\alpha)$ in the worst case.*

The exact definition of $\bar{\Psi}_\alpha$ will be given later. For now, it is enough to know that it is a distribution on $\mathbb{Z}_p$ that has the shape of a discrete Gaussian centered around 0 with standard deviation αp, as in Figure 1. Also, the probability of 0 (i.e., no error) is roughly $1/(\alpha p)$. A possible setting for the parameters is $p = O(n^2)$ and $\alpha = 1/(\sqrt{n} \log n)$ (in fact, these are the parameters that we use in our cryptographic application).

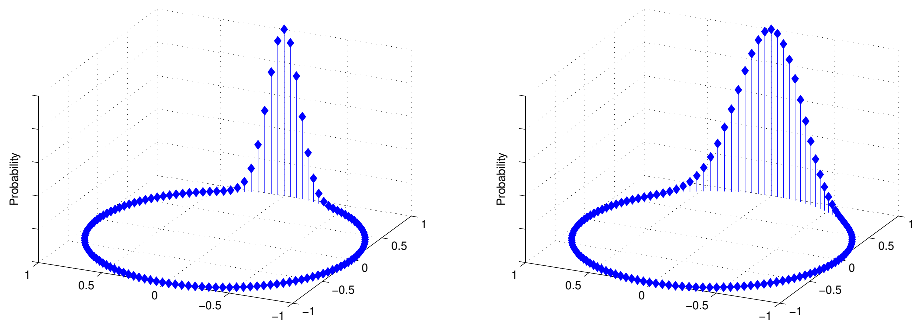

Figure 1: $\bar{\Psi}_\alpha$ for $p = 127$ with $\alpha = 0.05$ (left) and $\alpha = 0.1$ (right). The elements of $\mathbb{Z}_p$ are arranged on the circle.

The SVP and SIVP are two of the main computational problems on lattices. The best known polynomial time algorithms yield only mildly subexponential approximation factors. It is conjectured that there is no classical polynomial time algorithm that approximates them to within any polynomial factor. Lattice-based constructions of one-way functions, such as the one by Ajtai [2], are based on this conjecture.

One might guess that the same conjecture holds in the quantum world, i.e., there is no quantum polynomial time algorithm that approximates SVP (or SIVP) to within any polynomial factor. Thus one can interpret the main theorem as saying that based on this conjecture, the LWE problem is hard. The only evidence supporting this conjecture is that there are no quantum algorithms for lattice problems that are known to outperform classical algorithms, even though this is probably one of the most important open questions in the field of quantum computing. We do not know, however, if this conjecture is true.

In fact, one could also interpret our main theorem as a way to disprove this conjecture: if one finds an efficient algorithm for LWE, then one also obtains a quantum algorithm for approximating worst-case lattice problems. Such a result would be of tremendous importance on its own. Finally, we would like to stress that it is possible that our result can be made classical. This would make all our results stronger and the above discussion unnecessary.

The LWE problem can be equivalently presented as the problem of decoding random linear codes. More specifically, let $m = \mathrm{poly}(n)$ be arbitrary and let $\mathbf{s} \in \mathbb{Z}_p^n$ be some vector. Then, consider the following problem: given a random matrix $Q \in \mathbb{Z}_p^{m\times n}$ and the vector $\mathbf{t} = Q\mathbf{s} + \mathbf{e} \in \mathbb{Z}_p^m$ where each coordinate of the error vector $\mathbf{e} \in \mathbb{Z}_p^m$ is chosen independently from $\bar{\Psi}_\alpha$, recover $\mathbf{s}$. The Hamming weight of $\mathbf{e}$ is roughly $m(1 - 1/(\alpha p))$ (since a value chosen from $\bar{\Psi}_\alpha$ is 0 with probability roughly $1/(\alpha p)$). Hence, the Hamming distance of $\mathbf{t}$ from $Q\mathbf{s}$ is roughly $m(1 - 1/(\alpha p))$. Moreover, it can be seen that for large enough m, for any other word $\mathbf{s}'$, the Hamming distance of $\mathbf{t}$ from $Q\mathbf{s}'$ is roughly $m(1 - 1/p)$. Hence, we obtain that approximating the nearest codeword problem to within factors smaller than $(1 - 1/p)/(1 - 1/(\alpha p))$ on random codes is as hard as quantumly approximating worst-case lattice problems. This gives a partial answer to the important open question of understanding the hardness of decoding from random linear codes.

It turns out that certain problems, which are seemingly easier than the LWE problem, are in fact equivalent to the LWE problem. We establish these equivalences using elementary reductions. For example, being able to distinguish a set of equations as above from a set of equations in which the b_i's are chosen uniformly from $\mathbb{Z}_p$ is equivalent to solving LWE. Moreover, it is enough to correctly distinguish these two distributions for some non-negligible fraction of all $\mathbf{s}$. The latter formulation is the one we use in our cryptographic applications.

Cryptosystem. Using our main theorem, we obtain a public key cryptosystem whose security is based on the worst-cast quantum hardness of approximating SIVP and SVP to within $\tilde{O}(n^{1.5})$. In other words, breaking our cryptosystem implies an efficient quantum algorithm for approximating SIVP and SVP to within $\tilde{O}(n^{1.5})$. We should emphasize that the cryptosystem is completely classical; it is only in the proof of its security that we use quantum computation. The cryptosystem itself is quite simple; the reader is encouraged to glimpse at the beginning of Section 5.

Previous cryptosystems, such as the Ajtai-Dwork cryptosystem [3] and the one by Regev [19], were only based on the worst-case hardness of the unique-SVP problem, a special case of SVP whose hardness is not so well understood. Basing a cryptosystem on the worst-case hardness of SVP and SIVP is an important open problem. Our cryptosystem does not quite solve this open problem because of its quantum aspects.

Another important feature of our cryptosystem is its improved efficiency. In previous cryptosystems, the public key size is $\tilde{O}(n^4)$ and the encryption increases the size of messages by a factor of $\tilde{O}(n^2)$. In our cryptosystem, the public key size is only $\tilde{O}(n^2)$ and encryption increases the size of messages by a factor of only $\tilde{O}(n)$. This possibly makes our cryptosystem practical.

Finally, we mention that learning problems related to ours were already suggested as possible sources of cryptographic hardness, e.g., [8, 5], although this was done without establishing any connection to lattice problems.

Why quantum? This paper is almost entirely classical. In fact, quantum is needed only in one step in the proof of the main theorem. Making this step classical would make the entire reduction classical. To demonstrate the difficulty, consider the following situation. Let L be some lattice and let $d = \lambda_1(L)/n^{10}$ where $\lambda_1(L)$ is the length of the shortest nonzero vector in L. We are given an oracle that for any point $\mathbf{x} \in \mathbb{R}^n$ within distance d of L finds the closest lattice vector to $\mathbf{x}$. If $\mathbf{x}$ is not within distance d of L, the output of the oracle is undefined. Intuitively, such an oracle seems quite powerful; the best known algorithms for performing such a task require exponential time. Nevertheless, we do not see any way to use this oracle classically. Indeed, it seems to us that the only way to generate inputs to the oracle is the following: somehow choose a lattice point $\mathbf{y} \in L$ and let $\mathbf{x} = \mathbf{y} + \mathbf{z}$ for some perturbation vector $\mathbf{z}$ of length at most d. Clearly, on input $\mathbf{x}$ the oracle outputs $\mathbf{y}$. But this is useless since we already know $\mathbf{y}$!

It turns out that quantumly, such an oracle is quite useful. Indeed, being able to compute $\mathbf{y}$ from $\mathbf{x}$ allows us to *uncompute* $\mathbf{y}$. More precisely, it allows us to transform the quantum state $|\mathbf{x}, \mathbf{y}\rangle$ to the state $|\mathbf{x}, 0\rangle$ in a reversible (i.e., unitary) way. This ability to erase the contents of a memory cell in a reversible way seems useful only in the quantum setting.

Techniques. Our cryptosystem and its analysis are similar to those in the Ajtai-Dwork cryptosystem [3] and the one by Regev [19] (in fact, our analysis is somewhat simpler technically). We believe that the novel part of the paper is the main theorem and in particular, its proof. We now describe some of the new techniques in our proof.

All lattice-based constructions of one-way functions [2, 10, 15, 16, 18] use a construction known as an *iterative reduction*. Essentially, this means that instead of 'immediately' finding very short vectors in a lattice, the reduction proceeds in steps where in each step shorter lattice vectors are found. Typically, in each step the length of the vectors found decreases by some factor, say 2, so that after a polynomial number of steps we end up with vectors that are within some polynomial factor of the shortest. Iterative reductions, however, were so far never used in the construction of lattice-based public-key cryptosystems. Our main theorem is the first to apply iterative reductions in this context. Moreover, we believe that the use of iterative reductions is what allows us to relate the LWE problem to lattices, and in particular to problems like SVP and SIVP (as opposed to unique-SVP).

Our proof is based on the Fourier transform of Gaussian measures, a technique that was developed in previous papers [19, 18, 1]. More specifically, we use a parameter known as the smoothing parameter, as introduced in [18]. We also use the discrete Gaussian distribution and approximations to its Fourier transform, ideas that were developed in [1].

Open questions. The main open question raised by this work is whether our results can be dequantized, leading to stronger results based on the classical hardness of SIVP and SVP. We see no reason why this should be impossible. However, despite our efforts, we were not able to show this. As mentioned above, the difficulty is that there seems to be no classical way to use an oracle that solves the closest vector problem within small distances. Quantumly, however, such an oracle turns out to be quite useful.

Another important open question is to determine the hardness of the learning from parity with errors problem (i.e., the case $p = 2$). Our theorem only works for $p > 2\sqrt{n}$. It seems that in order to prove similar results for smaller values of p, substantially new ideas are required. Alternatively, one might interpret our inability to prove hardness for small p as an indication that substantially better algorithms should exist for this case. Currently, the best known algorithms do not perform significantly better for constant p.

Finally, let us mention another promising direction for future research. In a recent work of Feige [12], a conjecture was presented regarding the hardness of certain random instances of 3SAT. Based on this conjecture, Feige showed some interesting hardness of approximation results. In a follow-up work, Alekhnovich [5] showed strong inapproximability results for the nearest codeword problem and the problem of approximation matrix rigidity based on several other conjectures. His conjectures are in fact closely related to the LWE problem and perhaps using our main theorem, some inapproximability results can be derived based on the quantum-hardness of lattices.

1.1 Overview

In this subsection, we give a brief informal overview of the proof of our main theorem. We do not include a description of the reductions in Section 4 and the cryptosystem (Section 5) as these parts of the paper are more similar to previous work. A complete list of definitions is given in Section 2. The only definition required for this overview is that of the discrete Gaussian distribution $D_{L,r}$. This is the distribution in which the probability of each $\mathbf{x} \in L$ is proportional to $\exp\left(-\pi\|\mathbf{x}/r\|^2\right)$ (see Eq. (4) and Figure 2). Its support is the lattice L and for large enough r (which is the case in all our applications), vectors chosen from it have length roughly $r\sqrt{n}$ with high probability.

The proof of the main theorem shows a quantum algorithm that solves worst-case lattice problems using an oracle for the $\mathsf{LWE}_{p,\bar{\Psi}_\alpha}$ problem. For simplicity, we consider the SIVP, defined as follows. For a rank n lattice L, let $\lambda_n(L)$ denote the minimum length of a set of n linearly independent lattice vectors from L, where the length of a set is defined as the length of longest vector in it. Our goal is to find a set of n linearly independent vectors whose length is at most $\mathrm{poly}(n)\lambda_n(L)$. In the actual algorithm, an attempt is made to make this polynomial as small as possible. In this overview, we will not be concerned with this issue.

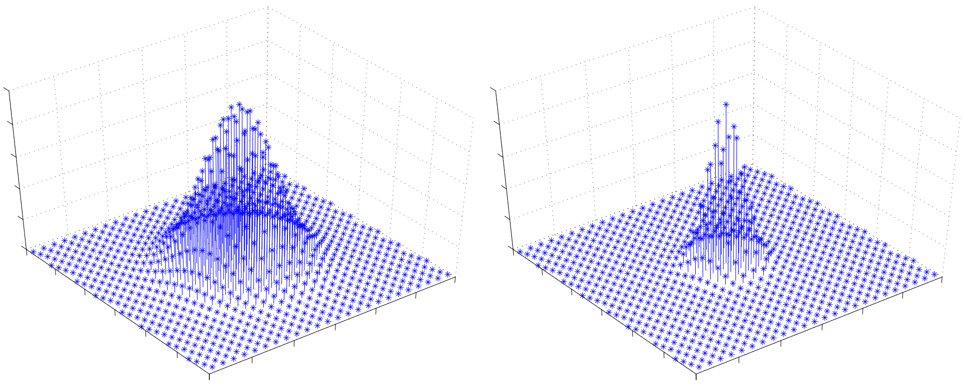

Figure 2: $D_{L,2}$ (left) and $D_{L,1}$ (right) for a two-dimensional lattice L. The z-axis represents probability.

Our algorithm is iterative, in the sense that it starts with some long vectors and repeatedly finds shorter and shorter vectors. More precisely, let $c > 0$ be some constant. Our algorithm starts with n^c samples from $D_{L,r}$ for some large enough r. Each iteration takes as input n^c samples from $D_{L,r}$ and generates n^c samples from $D_{L,r'}$ for $r' = r\sqrt{n}/(\alpha p)$. Notice that since $\alpha p > 2\sqrt{n}$, $r' < r/2$. Hence, by repeating this iterative step, we obtain samples from $D_{L,r}$ for smaller and smaller values of r. The algorithm stops when we no longer find n linearly vectors of length at most $r\sqrt{n}$ among the n^c samples from $D_{L,r}$.

There are two technical conditions that need to be checked. First, we should be able to efficiently produce samples from $D_{L,r}$ for some large enough r. This is indeed possible for $r \geq 2^{O(n)}\lambda_n(L)$ using a simple algorithm. Second, we should show that when the algorithm stops, $r \leq \mathrm{poly}(n)\lambda_n(L)$. This implies that we have a set of n linearly independent vectors of length at most $r\sqrt{n} \leq \mathrm{poly}(n)\lambda_n(L)$, as required. In the rest of this subsection, we ignore these issues and concentrate on the heart of the algorithm: the iterative step.

In the iterative step, we have as input n^c samples from $D_{L,r}$ and we are supposed to generate n^c samples from $D_{L,r'}$. The parameter r can be assumed to be large enough, say $r \geq n^{10}\lambda_n(L)$. We do this in two steps (see Figure 3). In the first step, we use these samples to construct an algorithm that solves $\mathsf{CVP}_{L^*,\alpha p/r}$, i.e., solves the closest vector problem on L^* for points that are within distance $\alpha p/r$ of the lattice. This algorithm is classical and uses the LWE oracle. In the second step, we use this algorithm to generate samples from $D_{L,r'}$. This step is quantum (and in fact, the only quantum part of our proof). In the following, we describe each of these steps briefly.

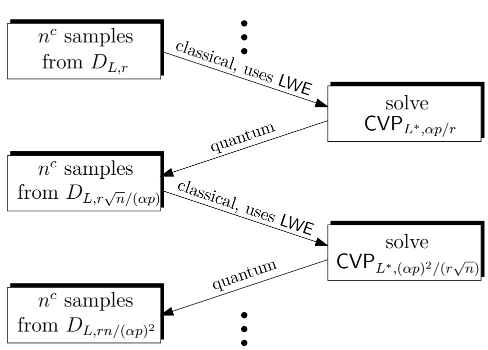

Figure 3: Two iterations of the algorithm

Step 1: Before describing this step, it would be helpful to recall the main idea in [1]. Consider some probability distribution D on some lattice L and consider its Fourier transform, call it f. This is a function from $\mathbb{R}^n$ to $\mathbb{R}$ that is L^*-periodic, i.e., $f(\mathbf{x}) = f(\mathbf{x}+\mathbf{y})$ for any $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in L^*$. In [1] it was shown that given a polynomial number of samples from D, one can compute an approximation to f to within $\pm 1/\mathrm{poly}(n)$. To see this, we use the definition of the Fourier transform to write f as

$$f(\mathbf{x}) = \sum_{\mathbf{y}\in L} D(\mathbf{y}) \exp\left(2\pi i \langle \mathbf{x}, \mathbf{y}\rangle\right) = \mathop{\mathrm{Exp}}_{\mathbf{y}\sim D}[\exp\left(2\pi i \langle \mathbf{x}, \mathbf{y}\rangle\right)]$$

where in the second equality we simply rewrite the sum as an expectation. By the Chernoff-Hoeffding bound, we see that if $\mathbf{y}_1, \ldots, \mathbf{y}_N$ are $N = \mathrm{poly}(n)$ independent samples from D, then

$$f(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} \exp\left(2\pi i \langle \mathbf{x}, \mathbf{y}_i\rangle\right)$$

where the approximation is to within $\pm 1/\mathrm{poly}(n)$, given that N is a large enough polynomial.

If we take our samples from the distribution $D_{L,r}$, we obtain a good approximation to its Fourier transform, which we denote by $f_{1/r}$. We omit the exact expression for $f_{1/r}$. It is enough to know that since r is large enough, $f_{1/r}$ is closely approximately by

$$f_{1/r}(\mathbf{x}) \approx \exp\left(-\pi (r \cdot \mathrm{dist}(L^*, \mathbf{x}))^2\right). \tag{1}$$

Hence, $f_{1/r}(\mathbf{x}) = 1$ for any $\mathbf{x} \in L^*$ and as one gets away from L^*, its value decreases. For points within distance, say, $1/r$ from the lattice, its value is still some positive constant $\exp(-\pi)$. However, as the distance from L^* increases, the value of the function soon becomes negligible (see Figure 4). We remark that for our choice of r, the distance between any two vectors in L^* is much larger than $1/r$ and hence these Gaussians are well-separated.

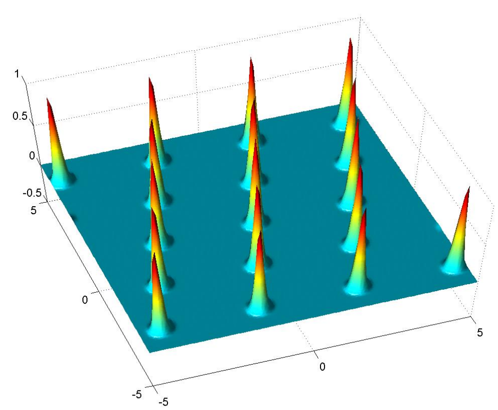

Figure 4: $f_{1/r}$ for a two-dimensional lattice

Although not needed in this paper, let us briefly outline how one can solve $\mathsf{CVP}_{L^*,1/r}$ using samples from $D_{L,r}$ (recall that our final goal is to solve the more difficult $\mathsf{CVP}_{L^*,\alpha p/r}$). Assume we are given some point $\mathbf{x}$ within distance $1/r$ of L^*. Intuitively, this $\mathbf{x}$ is located on one of the Gaussians of $f_{1/r}$. We start 'walking uphill' on $f_{1/r}$ in an attempt to find its 'peak'. This peak corresponds to the closest lattice point to $\mathbf{x}$. This procedure does not quite work: due to the error in our approximation of $f_{1/r}$, we cannot find the closest lattice point exactly. It is possible to overcome this difficulty; we omit further details. Notice that this procedure, unlike our solution to $\mathsf{CVP}_{L^*,\alpha p/r}$, does not need an LWE oracle. It is the LWE oracle that allows us to gain this extra factor of αp in the radius.

It turns out that the $\mathsf{CVP}_{L^*,\alpha p/r}$ can be reduced to the following problem. Given some input vector $\mathbf{x}$ within distance at most $\alpha p/r$ of L^*, let $\mathbf{y} \in L^*$ be the closest vector to $\mathbf{x}$ (for our choice of r, $\mathbf{y}$ is unique). Then the goal is to find the vector of coefficients of $\mathbf{y}$ reduced modulo p. Equivalently, our goal is to find $\tau(\mathbf{x}) := (L^*)^{-1}\mathbf{y} \bmod p \in \mathbb{Z}_p^n$. Hence, in the following it is enough to show how to find $\tau(\mathbf{x})$ given $\mathbf{x}$.

We first notice that the lattice L can be partitioned into p^n translates of the lattice pL. Namely, for each $\mathbf{a} \in \mathbb{Z}_p^n$, consider the set

$$pL + L\mathbf{a} = \{L\mathbf{b} \mid \mathbf{b} \in \mathbb{Z}^n,\ \ \mathbf{b} \bmod p = \mathbf{a}\}.$$

Then the set $\{pL + L\mathbf{a} \mid \mathbf{a} \in \mathbb{Z}_p^n\}$ forms a partition of L. We show that for any $\mathbf{a} \in \mathbb{Z}_p^n$, the probability that a point sampled from $D_{L,r}$ is in $pL + L\mathbf{a}$ is very close to p^{-n}. This holds because r is large enough, and each of the translated lattice $pL + L\mathbf{a}$ has roughly the same probability under $D_{L,r}$.

Consider the following simple procedure, call it SAMPLEPAIR: sample $\mathbf{y}'$ from $D_{L,r}$, let $\mathbf{a} \in \mathbb{Z}_p^n$ be such that $\mathbf{y}' \in pL + L\mathbf{a}$, let $\mathbf{y} = \mathbf{y}'/p$, and output the pair $(\mathbf{a}, \mathbf{y})$. Notice that we can perform this procedure efficiently since we assumed we are able to sample from $D_{L,r}$. From the above discussion we have that the marginal distribution of $\mathbf{a}$ is essentially uniform. Moreover, by definition we have that the distribution of $\mathbf{y}'$ conditioned on any $\mathbf{a}$ is $D_{pL+L\mathbf{a},r}$. Hence the distribution of $\mathbf{y}$ conditioned on any $\mathbf{a}$ is $D_{L+L\mathbf{a}/p,r/p}$. These facts imply that the distribution produced by SAMPLEPAIR is essentially identical to the following distribution: choose $\mathbf{a} \in \mathbb{Z}_p^n$ uniformly at random, sample $\mathbf{y}$ from $D_{L+L\mathbf{a}/p,r/p}$ and output $(\mathbf{a}, \mathbf{y})$. We use the latter formulation in the following analysis of SAMPLEPAIR.

If we could somehow obtain samples $(\mathbf{a}, \mathbf{y})$ with $\mathbf{a} = \mathbf{0}$, then the distribution of $\mathbf{y}$ would be $D_{L,r/p}$. As described above, using a polynomial number of such samples, we can solve $\mathsf{CVP}_{L^*,p/r}$ which is even better than what we need (since $\alpha < 1$). However, the probability that SAMPLEPAIR produces a pair with $\mathbf{a} = 0$ is exponentially small, namely, p^{-n}.

So let us now examine the Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ for nonzero $\mathbf{a}$ (see Figure 5). Intuitively, since $D_{L+L\mathbf{a}/p,r/p}$ looks like a translation of the distribution $D_{L,r/p}$, we expect its Fourier transform to be the same as that of $D_{L,r/p}$ up to phase. Indeed, a standard calculation shows that the Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ is given by

$$\exp\left(2\pi i \langle \mathbf{a}, \tau(\mathbf{x})\rangle / p\right) \cdot f_{p/r}(\mathbf{x}). \tag{2}$$

Notice that the absolute value of (2) is $f_{p/r}(\mathbf{x})$. If could somehow obtain a polynomial number of samples from $D_{L+L\mathbf{a}/p,r/p}$ *for the same* $\mathbf{a}$, then we could also obtain a good approximation to (2). By taking its absolute value, we would obtain a good approximation to $f_{p/r}$ from which we can proceed as before to solve $\mathsf{CVP}_{L^*,p/r}$. The problem with this solution is that SAMPLEPAIR produces pairs $(\mathbf{a}, \mathbf{y})$ for some uniform value $\mathbf{a} \in \mathbb{Z}_p^n$. With very high probability, we never obtain any value of $\mathbf{a}$ more than once. So it seems that we cannot hope to approximate (2) directly; a new approach is needed.

To explain our approach, let us step back and examine the expression $\exp\left(2\pi i \langle \mathbf{x}, \mathbf{y}\rangle\right)$. We know that its average over $\mathbf{y} \sim D_{L+L\mathbf{a}/p,r/p}$ is given by (2). For $\mathbf{x} \in L^*$, (2) is equal to $\exp\left(2\pi i \langle \mathbf{a}, \tau(\mathbf{x})\rangle / p\right)$. Since the absolute value of this expression is 1, we see that for such $\mathbf{x}$, the random variable $\langle \mathbf{x}, \mathbf{y}\rangle \bmod 1$ (where $\mathbf{y} \sim D_{L+L\mathbf{a}/p,r/p}$) is deterministically equal to $\langle \mathbf{a}, \tau(\mathbf{x})\rangle / p \bmod 1$ (this fact can also be shown more directly).

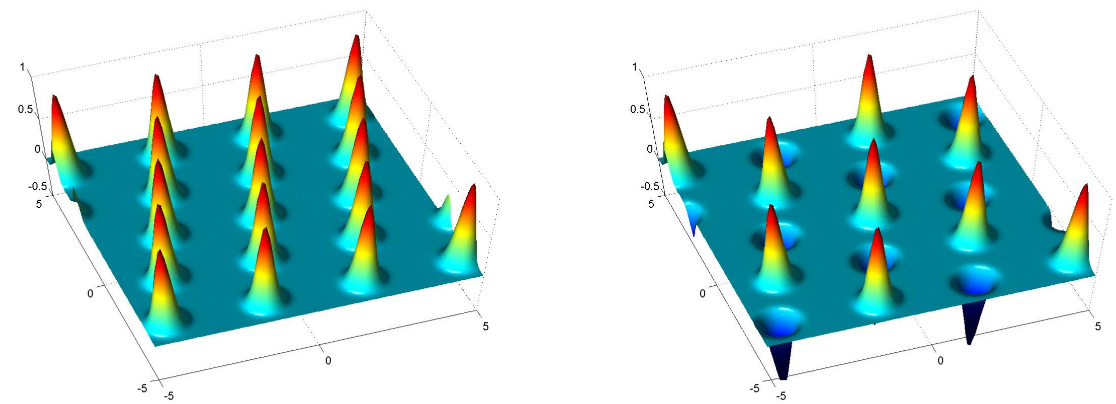

Figure 5: The Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ with $n=2$, $p=2$, $\mathbf{a}=(0,0)$ (left), $\mathbf{a}=(1,1)$ (right).

The situation becomes more interesting once we take a vector $\mathbf{x}$ whose distance to L^* is $\beta p/r$ for some $0 \le \beta \le \alpha$. Then the phase component of (2) remains the same (i.e., $\exp(2\pi i \langle \mathbf{a}, \tau(\mathbf{x})\rangle/p)$) but its magnitude decreases to some positive constant (namely, $\exp\left(-\pi\beta^2\right)$). One of our main technical contributions is to show that for such $\mathbf{x}$, the distribution of $\langle \mathbf{x}, \mathbf{y}\rangle \bmod 1$ (where $\mathbf{y} \sim D_{L+L\mathbf{a}/p,r/p}$) is close to a Gaussian centered around $\langle \mathbf{a}, \tau(\mathbf{x})\rangle/p \bmod 1$ with standard deviation β. Notice that for the case $\beta = 0$ we obtain the situation described before. Also note that as β increases, $\langle \mathbf{x}, \mathbf{y}\rangle \bmod 1$ becomes less concentrated. This causes the absolute value of (2) to decrease. Finally, let us remark that the distribution $\langle \mathbf{x}, \mathbf{y}\rangle \bmod 1$ is discrete; when we say that it is close to a Gaussian, we actually mean that by adding a small amount of noise, the distribution becomes a continuous distribution that is very close to a Gaussian.

Fix some $\mathbf{x}$ whose distance to L^* is $\beta p/r$ for some $0 \le \beta \le \alpha$. Then by the above discussion we know that if $\mathbf{y}$ is sampled from $D_{L+L\mathbf{a}/p,r/p}$ then $p\langle \mathbf{x}, \mathbf{y}\rangle \bmod p$ is distributed like a Gaussian around $\langle \mathbf{a}, \tau(\mathbf{x})\rangle \bmod p$ with standard deviation βp. By rounding, we obtain that $\lfloor p\langle \mathbf{x}, \mathbf{y}\rangle \rceil \bmod p$ is distributed like $\bar{\Psi}_\beta$ around $\langle \mathbf{a}, \tau(\mathbf{x})\rangle \bmod p$. This can be written as an 'equation with error'

$$\langle \mathbf{a}, \tau(\mathbf{x})\rangle \approx_{\bar{\Psi}_\beta} \lfloor p\langle \mathbf{x}, \mathbf{y}\rangle \rceil \pmod{p}.$$

We now create a polynomial number of equations and use the LWE oracle to find $\tau(\mathbf{x})$.

To summarize, our algorithm for solving $\mathsf{CVP}_{L^*,\alpha p/r}$ is the following. Let $\mathbf{x}$ be the input vector. Using SAMPLEPAIR, obtain a polynomial number of pairs $(\mathbf{a}_i, \mathbf{y}_i)$. We call the LWE oracle with the pairs $(\mathbf{a}_i, \lfloor p\langle \mathbf{x}, \mathbf{y}_i\rangle \rceil)$. By the arguments above, the LWE oracle outputs $\tau(\mathbf{x})$. As mentioned before, this is enough to solve $\mathsf{CVP}_{L^*,\alpha p/r}$.

The above description is not entirely accurate as the LWE oracle is only supposed to work for the noise distribution $\bar{\Psi}_\alpha$ and might not work for noise distribution $\bar{\Psi}_\beta$ for $\beta < \alpha$. We address this issue in the paper by showing how to transform an oracle that works with $\bar{\Psi}_\alpha$ to an oracle that works with $\bar{\Psi}_\beta$ for any (unknown) $\beta < \alpha$.

Step 2: In this step, we describe a quantum algorithm that, using a $\mathsf{CVP}_{L^*,\alpha p/r}$ oracle, generates one sample from $D_{L,r\sqrt{n}/(\alpha p)}$ (one can generate more samples by simply repeating the algorithm). Equivalently, let us show how to produce a sample from $D_{L,r}$ given a $\mathsf{CVP}_{L^*,\sqrt{n}/r}$ oracle. The procedure is essentially the following: first, by using the CVP oracle, create a quantum state corresponding to $f_{1/r}$. Then, apply the quantum Fourier transform and obtain a quantum state corresponding to $D_{L,r}$. By measuring this state we obtain a sample from $D_{L,r}$.

In the following, we describe this procedure in more detail. Our first goal is to create a quantum state corresponding to $f_{1/r}$. Informally, this can be written as

$$\sum_{\mathbf{x}\in\mathbb{R}^n} f_{1/r}|\mathbf{x}\rangle. \tag{3}$$

This state is clearly not well-defined. In the actual procedure, $\mathbb{R}^n$ is replaced with some finite set (namely, all points inside the basic parallelepiped of L^* that belong to some fine grid). This introduces several technical complications and makes the computations rather tedious. Therefore, in the rest of this subsection, we opt to continue with informal expressions as in (3).

Let us now continue our description of the procedure. In order to prepare the state in (3), we first create the uniform superposition on L^*,

$$\sum_{\mathbf{x}\in L^*} |\mathbf{x}\rangle.$$

On a separate register, we create a 'Gaussian state' of width $1/r$,

$$\sum_{\mathbf{z}\in\mathbb{R}^n} \exp\left(-\pi\|r\mathbf{z}\|^2\right)|\mathbf{z}\rangle.$$

This can be done using known techniques. The combined state of the system can be written as

$$\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} \exp\left(-\pi\|r\mathbf{z}\|^2\right)|\mathbf{x},\mathbf{z}\rangle.$$

We now add the first register to the second (a reversible operation), and obtain

$$\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} \exp\left(-\pi\|r\mathbf{z}\|^2\right)|\mathbf{x},\mathbf{x}+\mathbf{z}\rangle.$$

Finally, we would like to *erase*, or *uncompute*, the first register to obtain

$$\sum_{\mathbf{x}\in L^*,\mathbf{z}\in\mathbb{R}^n} \exp\left(-\pi\|r\mathbf{z}\|^2\right)|\mathbf{x}+\mathbf{z}\rangle \approx \sum_{\mathbf{z}\in\mathbb{R}^n} f_{1/r}|\mathbf{z}\rangle.$$

However, 'erasing' a register is in general not a reversible operation. In order for it to be reversible, we need to able to compute $\mathbf{x}$ from the remaining register $\mathbf{x}+\mathbf{z}$. This is precisely why we need the $\mathsf{CVP}_{L^*,\sqrt{n}/r}$ oracle. It can be shown that almost all the mass of $\exp\left(-\pi\|r\mathbf{z}\|^2\right)$ is on $\mathbf{z}$ such that $\|\mathbf{z}\| \le \sqrt{n}/r$. Hence, $\mathbf{x}+\mathbf{z}$ is within distance $\sqrt{n}/r$ of the lattice and the oracle finds the closest lattice point, namely, $\mathbf{x}$. This allows us to erase the first register in a reversible way.

In the final part of the procedure, we apply the quantum Fourier transform. This yields the quantum state corresponding to $D_{L,r}$, namely,

$$\sum_{\mathbf{y}\in L} D_{L,r}|\mathbf{y}\rangle.$$

By measuring this state, we obtain a sample from the distribution $D_{L,r}$ (in fact, the probability distribution is $D_{L,r}^2 = D_{L,r/\sqrt{2}}$ but this is a minor issue).

2 Preliminaries

In this section we include some notation that will be used throughout the paper. Most of the notation is standard. Less standard notation includes: the Gaussian function ρ, the Gaussian distribution ν, the periodic normal distribution Ψ, the discretization of a distribution on $\mathbb{T}$, the discrete Gaussian distribution D, the unique closest lattice vector κ, and the smoothing parameter η.

General. All logarithms are of base 2 unless otherwise specified. For two real numbers x and $y > 0$ we define $x \bmod y$ as $x - \lfloor x/y \rfloor y$. For $x \in \mathbb{R}$ we define $\lfloor x \rceil$ as the integer closest to x or, in case two such integers exist, the smaller of the two. For any integer $p \geq 2$, we write $\mathbb{Z}_p$ for the cyclic group $\{0, 1, \ldots, p-1\}$ with addition modulo p. We also write $\mathbb{T}$ for $\mathbb{R}/\mathbb{Z}$, i.e., the segment $[0, 1)$ with addition modulo 1.

We define a negligible amount in n as an amount that is asymptotically smaller than n^{-c} for any constant $c > 0$. More precisely, $f(n)$ is a negligible function in n if $\lim_{n\to\infty} n^c f(n) = 0$ for any $c > 0$. Similarly, a non-negligible amount is one which is at least n^{-c} for some $c > 0$. Also, when we say that an expression is exponentially small in n we mean that it is at most $2^{-\Omega(n)}$. Finally, when we say that an expression (most often, some probability) is exponentially close to 1, we mean that it is $1 - 2^{-\Omega(n)}$.

We say that an algorithm $\mathcal{A}$ with oracle access is a distinguisher between two distributions if its acceptance probability when the oracle outputs samples of the first distribution and its acceptance probability when the oracle outputs samples of the second distribution differ by a non-negligible amount.

Essentially all algorithms and reductions in this paper are probabilistic and work with probability exponentially close to 1.

For clarity, we present some of our reductions in a model that allows operations on real numbers. It is possible to modify them in a straightforward way so that they operate in a model that approximates real numbers up to an error of 2^{-n^c} for arbitrary large constant c in time polynomial in n.

Given two probability density functions ϕ_1, ϕ_2 on $\mathbb{R}^n$, we define the statistical distance between them as

$$\Delta(\phi_1, \phi_2) := \int_{\mathbb{R}^n} |\phi_1(\mathbf{x}) - \phi_2(\mathbf{x})| d\mathbf{x}$$

(notice that with this definition, the statistical distance ranges in $[0, 2]$). A similar definition holds for discrete random variables. We sometimes abuse notation, and use the same notation for two arbitrary functions. We note that the triangle inequality

$$\Delta(f_1, f_3) \leq \Delta(f_1, f_2) + \Delta(f_2, f_3)$$

holds for any functions f_1, f_2, f_3. One important fact that we use is that the statistical distance cannot increase by applying a (possibly randomized) function f, i.e.,

$$\Delta(f(X), f(Y)) \leq \Delta(X, Y),$$

see, e.g., [17]. In particular, this implies that the acceptance probability of any algorithm on inputs from X differs from its acceptance probability on inputs from Y by at most $\frac{1}{2}\Delta(X, Y)$ (the factor half coming from the choice of normalization in our definition of Δ).

Gaussians and other distributions. Recall that the *normal distribution* with mean 0 and variance σ^2 is the distribution on $\mathbb{R}$ given by the density function $\frac{1}{\sqrt{2\pi}\cdot\sigma} \exp\left(-\frac{1}{2}(\frac{x}{\sigma})^2\right)$ where $\exp(y)$ denotes e^y. Also recall that the sum of two independent normal variables with mean 0 and variances σ_1^2 and σ_2^2 is a normal variable with mean 0 and variance $\sigma_1^2 + \sigma_2^2$. For a vector $\mathbf{x}$ and any $s > 0$, let

$$\rho_s(\mathbf{x}) = \exp\left(-\pi\|\mathbf{x}/s\|^2\right)$$

be a Gaussian function scaled by a factor of s. Note that $\int_{\mathbf{x}\in\mathbb{R}^n} \rho_s(\mathbf{x})d\mathbf{x} = s^n$. So, $\nu_s := \rho_s/s^n$ is an n-dimensional probability density function. The dimension n is implicit. Notice that the Gaussian distribution ν_s can be expressed as the sum of n orthogonal 1-dimensional Gaussian distributions, and each of them can be efficiently approximated with arbitrary precision using standard techniques. So, the distribution ν_s can be efficiently approximated. For simplicity, in this paper we assume we can sample from ν_s exactly.[1] When s is not specified, we assume that it is 1. Functions are extended to sets in the usual way; e.g., $\rho_s(A) = \sum_{\mathbf{x}\in A} \rho_s(\mathbf{x})$ for any countable set A. For any vector $\mathbf{c} \in \mathbb{R}^n$, we define $\rho_{s,\mathbf{c}}(\mathbf{x}) := \rho_s(\mathbf{x} - \mathbf{c})$ to be a shifted version of ρ_s.

For any countable set A and a parameter $s > 0$, we define the discrete Gaussian probability distribution $D_{A,s}$ as

$$\forall \mathbf{x} \in A,\ D_{A,s}(\mathbf{x}) = \frac{\rho_s(\mathbf{x})}{\rho_s(A)}. \tag{4}$$

See Figure 2 for an illustration.

For $\beta \in \mathbb{R}^+$ the distribution Ψ_β is the distribution on $\mathbb{T}$ obtained by sampling from a normal variable with mean 0 and standard deviation $\frac{\beta}{\sqrt{2\pi}}$ and reducing the result modulo 1 (i.e., a periodization of the normal distribution):

$$\Psi_\beta(r) := \sum_{k=-\infty}^{\infty} \frac{1}{\beta} \cdot \exp\left(-\pi\left(\frac{r-k}{\beta}\right)^2\right).$$

Clearly, one can efficiently sample from Ψ_β. For an arbitrary probability distribution with density function $\phi : \mathbb{T} \to \mathbb{R}^+$ and some integer $p > 0$ we define its discretization $\bar{\phi} : \mathbb{Z}_p \to \mathbb{R}^+$ as the discrete probability distribution obtained by sampling from ϕ, multiplying by p, and rounding to the closest integer modulo p. More formally,

$$\bar{\phi}(i) := \int_{(i-1/2)/p}^{(i+1/2)/p} \phi(x)dx.$$

As an example, $\bar{\Psi}_\beta$ is shown in Figure 1.

Let $p \geq 2$ be some integer, and let $\chi : \mathbb{Z}_p \to \mathbb{R}^+$ be some probability distribution on $\mathbb{Z}_p$. Let n be an integer and let $\mathbf{s} \in \mathbb{Z}_p^n$ be a vector. We define $A_{\mathbf{s},\chi}$ as the distribution on $\mathbb{Z}_p^n \times \mathbb{Z}_p$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_p^n$ uniformly at random, choosing $e \in \mathbb{Z}_p$ according to χ, and outputting $(\mathbf{a}, \langle\mathbf{a},\mathbf{s}\rangle + e)$, where additions are performed in $\mathbb{Z}_p$, i.e., modulo p. We also define U as the uniform distribution on $\mathbb{Z}_p^n \times \mathbb{Z}_p$.

For a probability density function ϕ on $\mathbb{T}$, we define $A_{\mathbf{s},\phi}$ as the distribution on $\mathbb{Z}_p^n \times \mathbb{T}$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_p^n$ uniformly at random, choosing $e \in \mathbb{T}$ according to ϕ, and outputting $(\mathbf{a}, \langle\mathbf{a},\mathbf{s}\rangle/p + e)$, where the addition is performed in $\mathbb{T}$, i.e., modulo 1.

Learning with errors. For an integer $p = p(n)$ and a distribution χ on $\mathbb{Z}_p$, we say that an algorithm solves $\mathsf{LWE}_{p,\chi}$ if, given samples from $A_{\mathbf{s},\chi}$ for some arbitrary $\mathbf{s}$, it outputs $\mathbf{s}$ with probability exponentially close to 1. Similarly, for a probability density function ϕ on $\mathbb{T}$, we say that an algorithm solves $\mathsf{LWE}_{p,\phi}$ if, given

[1] In practice, when only finite precision is available, ν_s can be approximated by picking a fine grid, and picking points from the grid with probability approximately proportional to ν_s. All our arguments can be made rigorous by selecting a sufficiently fine grid.

samples from $A_{\mathbf{s},\phi}$ for some arbitrary $\mathbf{s}$, it outputs $\mathbf{s}$ with probability exponentially close to 1. In both cases, we say that the algorithm is efficient if it runs in polynomial time in n. Finally, we note that p is assumed to be prime only in Lemma 4.2. In the rest of the paper, including the main theorem, p can be any arbitrary integer.

Lattices. We briefly review some basic definitions; for a good introduction to lattices, see [17]. 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. Given a basis $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ of a lattice L, the fundamental parallelepiped is defined as

$$\mathcal{P}(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{v}_i \;\middle|\; x_i \in [0, 1) \right\}.$$

When the choice of basis is clear, we write $\mathcal{P}(L)$ instead of $\mathcal{P}(\mathbf{v}_1, \ldots, \mathbf{v}_n)$. Note that a lattice has a different fundamental parallelepiped for every possible basis. We denote by $\det(L)$ the volume of the fundamental parallelepiped of L or equivalently, the determinant of the matrix whose columns are the basis vectors of the lattice ($\det(L)$ is a lattice invariant, i.e., it is independent of the choice of basis). The dual of a lattice L in $\mathbb{R}^n$, denoted L^*, is the set of all vectors $y \in \mathbb{R}^n$ such that $\langle x, y \rangle \in \mathbb{Z}$ for all vectors $x \in L$. Similarly, given a basis $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ of a lattice, we define the dual basis as the set of vectors $(\mathbf{v}_1^*, \ldots, \mathbf{v}_n^*)$ such that $\langle \mathbf{v}_i, \mathbf{v}_j^* \rangle = \delta_{ij}$ for all $i, j \in [n]$ where δ_{ij} denotes the Kronecker delta, i.e., 1 if $i = j$ and 0 otherwise. With a slight abuse of notation, we sometimes write L for the $n \times n$ matrix whose columns are $\mathbf{v}_1, \ldots, \mathbf{v}_n$. With this notation, we notice that $L^* = (L^T)^{-1}$. From this it follows that $\det(L^*) = 1/\det(L)$. As another example of this notation, for a point $\mathbf{v} \in L$ we write $L^{-1}\mathbf{v}$ to indicate the coefficient vector of $\mathbf{v}$.

The point $\mathbf{x} \in \mathbb{R}^n$ reduced modulo the parallelepiped $\mathcal{P}(L)$, denoted $x \bmod \mathcal{P}(L)$, is the unique point $\mathbf{y} \in \mathcal{P}(L)$ such that $\mathbf{y} - \mathbf{x} \in L$.

Let $\lambda_1(L)$ denote the length of the shortest nonzero vector in the lattice L. We denote by $\lambda_n(L)$ the minimum length of a set of n linearly independent vectors from L, where the length of a set is defined as the length of longest vector in it. For a lattice L and a point v whose distance from L is less than $\lambda_1(L)/2$ we define $\kappa_L(v)$ as the (unique) closest point to v in L.

Two main lattice problem are the SVP and the SIVP. The former is concerned with finding (or approximating) $\lambda_1(L)$, while the latter is concerned with $\lambda_n(L)$. We also define a non-standard variant of the closest vector. For an n-dimensional lattice L, and some $d > 0$, we say that an algorithm solves $\mathsf{CVP}_{L,d}$ if, given a point $\mathbf{x} \in \mathbb{R}^n$ whose distance to L is at most d, the algorithm finds the closest lattice point to $\mathbf{x}$. We note that in our applications, d will always be smaller than $\lambda_1(L)/2$. Hence, the closest vector is unique.

The smoothing parameter. In our analysis, we use a relatively new lattice parameter known as the *smoothing parameter* [18]. This makes the analysis tighter and allows us to obtain better approximation factors. Intuitively, this parameter quantifies the amount of noise that one has to add to a lattice in order for it to lose its discrete structure. The precise definition is the following:

Definition 2.1 *For an n-dimensional lattice L, and positive real $\varepsilon > 0$, we define its smoothing parameter $\eta_\varepsilon(L)$ to be the smallest s such that $\rho_{1/s}(L^* \setminus \{\mathbf{0}\}) \le \varepsilon$.*

In other words, $\eta_\varepsilon(L)$ is the smallest s such that a Gaussian measure on the dual lattice L^* scaled by $1/s$ gives all but $\varepsilon/(1+\varepsilon)$ of its weight to the origin. We usually take ε to be some negligible function of the lattice dimension. Notice that $\rho_{1/s}(L^* \setminus \{\mathbf{0}\})$ is a continuous and strictly decreasing function of s such that $\lim_{s\to 0} \rho_{1/s}(L^* \setminus \{\mathbf{0}\}) = \infty$ and $\lim_{s\to\infty} \rho_{1/s}(L^* \setminus \{\mathbf{0}\}) = 0$. So, the parameter $\eta_\varepsilon(L)$ is well defined for any $\varepsilon > 0$, and $\varepsilon \mapsto \eta_\varepsilon(L)$ is the inverse function of $s \mapsto \rho_{1/s}(L^* \setminus \{\mathbf{0}\})$. In particular, $\eta_\varepsilon(L)$ is also a continuous and strictly decreasing function of ε.

The motivation for this definition (and the name 'smoothing parameter') comes from the following result, shown in [18]. Informally, it says that if we choose a 'random' lattice point from an n-dimensional lattice L and add noise ν_s for some $s > \eta_\varepsilon(L)$ then the resulting distribution is within statistical distance ε of the 'uniform distribution on $\mathbb{R}^n$' (think of ε as some negligible function of n). The formal statement is given by considering distributions on $\mathcal{P}(L)$, see [18] for further details. In this paper, we show another important property of this parameter: for $s > \sqrt{2}\eta_\varepsilon(L)$, if we sample a point from $D_{L,s}$ and add Gaussian noise ν_s, we obtain a distribution whose statistical distance to a continuous Gaussian $\nu_{\sqrt{2}s}$ is at most 4ε. Notice that $\nu_{\sqrt{2}s}$ is the distribution one expects to obtain when summing two independent samples from ν_s. Hence, intuitively, the noise ν_s is enough to hide the discrete the discrete structure of $D_{L,s}$.

The following two upper bounds on the smoothing parameter appear in [18].

Lemma 2.2 *For any n-dimensional lattice L, $\eta_\varepsilon(L) \le \sqrt{n}/\lambda_1(L^*)$ where $\varepsilon = 2^{-n}$.*

Lemma 2.3 *For any n-dimensional lattice L and positive real $\varepsilon > 0$,*

$$\eta_\varepsilon(L) \le \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}} \cdot \lambda_n(L).$$

In particular, for any superlogarithmic function $\omega(\log n)$, $\eta_{\varepsilon(n)}(L) \le \sqrt{\omega(\log n)} \cdot \lambda_n$ for some negligible function $\varepsilon(n)$.

We also need the following simple lower bound on the smoothing parameter:

Claim 2.4 *For any lattice L and any $\varepsilon > 0$,*

$$\eta_\varepsilon(L) \ge \sqrt{\frac{\ln 1/\varepsilon}{\pi}} \cdot \frac{1}{\lambda_1(L^*)}.$$

In particular, for any negligible function $\varepsilon(n)$ and any constant $c > 0$, $\eta_{\varepsilon(n)}(L) > c/\lambda_1(L^*)$ for large enough n.

Proof: Let $\mathbf{v} \in L^*$ be a vector of length $\lambda_1(L^*)$ and let $s = \eta_\varepsilon(L)$. Then,

$$\varepsilon = \rho_{1/s}(L^* \setminus \{\mathbf{0}\}) \ge \rho_{1/s}(\mathbf{v}) = \exp\left(-\pi(s\lambda_1(L^*))^2\right).$$

We obtain the required bound by solving for s. ■

The Fourier transform. We briefly review some of the important properties of the Fourier transform. In the following, we omit certain technical conditions as these will always be satisfied in our applications. For a more precise and in-depth treatment, see, e.g., [11]. The Fourier transform of a function $h : \mathbb{R}^n \mapsto \mathbb{R}$ is defined to be

$$\hat{h}(\mathbf{w}) = \int_{\mathbb{R}^n} h(\mathbf{x}) e^{-2\pi i \langle \mathbf{x}, \mathbf{w} \rangle} d\mathbf{x}.$$

From the definition we can obtain two useful formulas; first, if h is defined by $h(\mathbf{x}) = g(\mathbf{x}+\mathbf{v})$ for some function g and vector $\mathbf{v}$ then
$$\hat{h}(\mathbf{w}) = e^{2\pi i\langle \mathbf{v},\mathbf{w}\rangle}\hat{g}(\mathbf{w}). \tag{5}$$
Similarly, if h is defined by $h(\mathbf{x}) = e^{2\pi i\langle \mathbf{x},\mathbf{v}\rangle} g(\mathbf{x})$ for some function g and vector $\mathbf{v}$ then
$$\hat{h}(\mathbf{w}) = \hat{g}(\mathbf{w}-\mathbf{v}). \tag{6}$$

Another important fact is that the Gaussian is its own Fourier transform, i.e., $\hat{\rho} = \rho$. More generally, for any $s > 0$ it holds that $\widehat{\rho_s} = s^n \rho_{1/s}$. We will use the following formulation of the Poisson summation formula:

Lemma 2.5 *For any lattice L and any function $f : \mathbb{R}^n \to \mathbb{C}$, $f(L) = \det(L^*)\hat{f}(L^*)$ where $\hat{f}$ denotes the Fourier transform of f.*

3 Main Theorem

Theorem 3.1 *Let $\varepsilon = \varepsilon(n)$ be some negligible function of n. Also, let $\alpha = \alpha(n) \in (0,1)$ be some real and let $p = p(n)$ be some integer such that $\alpha p > 2\sqrt{n}$. Assume there exists an efficient (possibly quantum) algorithm W that solves $\mathsf{LWE}_{p,\Psi_\alpha}$. Then there exists an efficient quantum algorithm for solving the following worst-case lattice problems:*

- *Find a set of n linearly independent lattice vectors of length at most $2n \cdot \eta_\varepsilon(L)/\alpha \le \tilde{O}(\lambda_n(L)\cdot n/\alpha)$.*
- *Approximate $\lambda_1(L)$ to within $\tilde{O}(n/\alpha)$.*

Proof: We only present an algorithm for finding short linearly independent lattice vectors. The other result will be included in the full version. Our algorithm is the following.

1 Compute $\tilde{\lambda_n}$ such that $1 \le \tilde{\lambda_n}/\lambda_n(L) \le 2^n$
2 For each $k \in \{1, 2, \ldots, \lceil \log p \rceil\}$ do
3    $r \leftarrow 2^{4n}\tilde{\lambda}_n \cdot 2^k$
4   Let $\mathcal{S}$ be n^c samples taken from $D_{L,r}$
5   While $\mathcal{S}$ contains n lin. indep. vectors from L of length $\le r\sqrt{n}$ do
6       $\mathcal{S}_k \leftarrow n$ lin. indep. vectors of length $\le r\sqrt{n}$ from $\mathcal{S}$
7       $r' \leftarrow r\sqrt{n}/(\alpha p)$
8   Let $\mathcal{S}'$ be n^c samples taken from $D_{L,r'}$
9   Set $r \leftarrow r'$ and $\mathcal{S} \leftarrow \mathcal{S}'$
10 Output the shortest set among $\mathcal{S}_1, \ldots, \mathcal{S}_{\lceil \log p \rceil}$

We now explain the structure and implementation of this algorithm. The rough estimate on $\lambda_n(L)$ in Step 1 can be obtained by using the LLL algorithm [14]. In the outer loop, we try several different values for k. Each iteration yields a candidate set $\mathcal{S}_k$ of n short linearly independent vectors. In the last step we output the shortest of these sets, where the length of a set is defined as the length of the longest vector in it. We remark that the only purpose of this outer loop is to achieve the $\tilde{O}(n/\alpha)$ approximation factor. Without it, the algorithm would obtain somewhat worse polynomial factors.

The implementation of Step 4 is described in Lemma 3.2. We choose $c = \max\{c_1, 2\}$ where c_1 is the constant from Lemma 3.3 and the constant 2 comes from Lemma 3.15.

The iterative step, Step 8, is the heart of the algorithm and is described in Lemma 3.3. It allows us to create samples from $D_{L,r\sqrt{n}/(\alpha p)}$ by using samples from $D_{L,r}$. This distribution is narrower by a constant factor since $r\sqrt{n}/(\alpha p) \leq r/2$. We remark that the description of this step is a little misleading. Indeed, in order to obtain samples from $D_{L,r\sqrt{n}/(\alpha p)}$, the condition $r \geq p\eta_\varepsilon(L)$ in Lemma 3.3 must hold. Since the algorithm does not check this condition, Step 8 might run for smaller than r. In such cases, we do not assume anything about the set $\mathcal{S}'$: it might contain arbitrary vectors that do not correspond to the distribution $D_{L,r\sqrt{n}/(\alpha p)}$. As we shall see later, this poses no difficulty.

The analysis of the algorithm is completed in Subsection 3.3. There, we show that its running time is polynomial and that it indeed outputs n linearly independent vectors of length at most $2n \cdot \eta_\varepsilon(L)/\alpha$.

Finally, we note the following crucial fact: using n^c samples from $D_{L,r}$, the iterative step generates the same number of samples n^c from $D_{L,r\sqrt{n}/(\alpha p)}$ (in fact, we could even generate more than n^c samples). The algorithm would not work if we could only generate, say, $n^c/2$ samples. This is because the number of iterations of Step 8 is polynomial and if the number of samples is halved in each iteration, we would have to start with an exponential number of samples.

■

3.1 Bootstrapping

Lemma 3.2 *Given any lattice L, we can efficiently sample from $D_{L,r}$ for $r > 2^{4n}\lambda_n(L)$.*

Proof: By using the LLL basis reduction algorithm [14], we can obtain a basis for L of length at most $2^n\lambda_n(L)$. For any $\mathbf{y} \in \mathbb{R}^n$, let $\xi(\mathbf{y})$ denote the unique lattice vector $\mathbf{v} \in L$ such that $\mathbf{y} \in \mathbf{v} + \mathcal{P}(L)$. In other words, for any $\mathbf{v} \in L$, ξ maps $\mathbf{v} + \mathcal{P}(L)$ to $\mathbf{v}$. Notice that for any vector $\mathbf{y}$, $\|\xi(\mathbf{y}) - \mathbf{y}\| \leq \text{diam}(\mathcal{P}(L)) \leq n2^n\lambda_n(L)$. We can compute $\xi(\mathbf{y})$ efficiently by writing $\mathbf{y} = L\mathbf{a}$ and outputting $L(\lfloor a_1 \rfloor, \ldots, \lfloor a_n \rfloor)$. The sampling procedure samples a vector $\mathbf{y}$ from ν_r and then outputs $\xi(\mathbf{y})$.

Consider the n-dimensional distribution whose density function μ is given by $\mu(\mathbf{v}) := \nu_r(\xi(\mathbf{v}))/Z$ where Z is a normalizing factor. By Claim A.3,

$$\Delta(\mu, \nu_r) \leq 100n\sqrt{n2^n\lambda_n(L)/(2^{4n}\lambda_n(L))} \leq 2^{-n}.$$

It remains to notice that the distribution obtained by sampling a point $\mathbf{v}$ from μ and outputting $\xi(\mathbf{v})$ is exactly $D_{L,r}$. Hence, the distribution given by the above procedure is exponentially close to $D_{L,r}$.

■

3.2 The iterative step

Lemma 3.3 *Let L be an n-dimensional lattice, $\varepsilon = \varepsilon(n)$ be some negligible function, $\alpha = \alpha(n) \in (0,1)$ be some real, and $p = p(n) \geq 2$ be some integer. Assume we are given n^{c_1} samples from $D_{L,r}$ for some $r > p\eta_\varepsilon(L)$ and a large enough c_1. Also assume that we have an efficient algorithm W that solves $\mathsf{LWE}_{p,\Psi_\alpha}$. Then, there exists an efficient quantum algorithm that produces any polynomial number of samples from $D_{L,r\sqrt{n}/(\alpha p)}$.*

Proof: The algorithm consists of two main parts. The first part is shown in Lemma 3.4. There, we describe a (classical) algorithm that using W and the samples from $D_{L,r}$, solves $\mathsf{CVP}_{L^*,\alpha p/r}$. The second part is described in Lemma 3.13. There, we describe a quantum algorithm that, given an oracle that solves $\mathsf{CVP}_{L^*,\alpha p/r}$, outputs samples from $D_{L,r\sqrt{n}/(\alpha p)}$. This is the only quantum part in this paper.

Note that by Claim 2.4,

$$\frac{\alpha p}{r} \le \frac{\alpha}{\eta_\varepsilon(L)} \le \frac{1}{\eta_\varepsilon(L)} \le \frac{\lambda_1(L^*)}{2}$$

and hence the $\mathsf{CVP}_{L^*,\alpha p/r}$ is well-defined. This fact is implicitly used in the next two subsections. ∎

3.2.1 From samples to CVP

The main result of this subsection is the following:

Lemma 3.4 *Let L be an n-dimensional lattice, $\varepsilon = \varepsilon(n)$ be some negligible function, $\alpha = \alpha(n) \in (0,1)$ be some real, and $p \ge 2$ be some integer. Assume we can sample from $D_{L,r}$ for some $r > p\eta_\varepsilon(L)$. Also assume that we have an efficient algorithm W that solves $\mathsf{LWE}_{p,\Psi_\alpha}$. Then, there exists an efficient algorithm that solves $\mathsf{CVP}_{L^*,\alpha p/r}$.*

Proof: By scaling, we can assume without loss of generality that $r = 1$. The proof follows by combining several lemmas. First, using Lemma 3.6, we obtain an algorithm W' that, given samples from $A_{\mathbf{s},\psi_\beta}$ for some (unknown) $\beta < \alpha$, outputs $\mathbf{s}$ with probability exponentially close to 1. Using Lemma 3.8 with W' we obtain an algorithm that finds, given any point $\mathbf{x}$ within distance αp of L^*, $(L^*)^{-1}\kappa_{L^*}(\mathbf{x}) \bmod p \in \mathbb{Z}_p^n$, the coefficient vector of the closest vector to $\mathbf{x}$ reduced modulo p. By the remark following the statement of Lemma 3.8, we know that $\alpha p < \lambda_1(L^*)/2$. Hence, using Lemma 3.7, we obtain a solution for $\mathsf{CVP}_{L^*,\alpha p}$. ∎

Lemma 3.5 *There exists an efficient algorithm that, given $\mathbf{s}'$ and samples from $A_{\mathbf{s},\psi_\alpha}$ for some (unknown) $\mathbf{s}$ and $\alpha < 1$, outputs whether $\mathbf{s} = \mathbf{s}'$ and is correct with probability exponentially close to 1.*

Proof: The idea is to perform a statistical test on samples from $A_{\mathbf{s},\psi_\alpha}$ that checks whether $\mathbf{s} = \mathbf{s}'$. Let ξ be the distribution on $\mathbb{T}$ obtained by sampling $(\mathbf{a}, x)$ from $A_{\mathbf{s},\psi_\alpha}$ and outputting $x - \langle \mathbf{a}, \mathbf{s}'\rangle/p \in \mathbb{T}$. For some large enough $N = \mathrm{poly}(n)$, the algorithm takes N samples $y_1, \ldots, y_N$ from ξ. It then computes $z := \frac{1}{N}\sum_{i=1}^N \cos(2\pi y_i)$. If $z > 0.01$, it decides that $\mathbf{s} = \mathbf{s}'$, otherwise it decides that $\mathbf{s} \neq \mathbf{s}'$.

We now analyze this algorithm. Consider the distribution ξ. Notice that it be obtained by sampling e from Ψ_α, sampling $\mathbf{a}$ uniformly from $\mathbb{Z}_p^n$ and outputting $e + \langle \mathbf{a}, \mathbf{s} - \mathbf{s}'\rangle/p \in \mathbb{T}$. From this it easily follows that if $\mathbf{s} = \mathbf{s}'$, ξ is exactly Ψ_α. Otherwise, if $\mathbf{s} \neq \mathbf{s}'$, we claim that ξ has a period of $1/k$ for some integer $k \ge 2$. Indeed, let j be an index on which $s_j \neq s'_j$. Then the distribution of $a_j(s_j - s'_j) \bmod p$ is periodic with period $\mathrm{gcd(p, s_j - s'_j)} < \mathrm{p}$. This clearly implies that the distribution of $a_j(s_j - s'_j)/p \bmod 1$ is periodic with period $1/k$ for some $k \ge 2$. Since a sample from ξ can be obtained by adding a sample from $a_j(s_j - s'_j)/p \bmod 1$ and an independent sample from some other distribution, we obtain that ξ also has the same period of $1/k$.

Consider the expectation

$$\mathop{\mathrm{Exp}}_{y\sim\xi}[\cos(2\pi y)] = \int_0^1 \cos(2\pi y)\xi(y)dy = \mathrm{Re}\Big[\int_0^1 \exp(2\pi i y)\xi(y)dy\Big]. \tag{7}$$

We now compute this integral for two cases; we remark that this integral is no other than the Fourier series of ξ at point 1 and that the following arguments can be explained in terms of properties of the Fourier series.

First, a routine calculation shows that for $\xi = \Psi_\alpha$, the value of this expectation is exactly $\exp\left(-\pi\alpha^2\right)$ and that this value is at least 0.04 for $\alpha < 1$. Moreover, if ξ has a period of $1/k$ for some $k \geq 2$, then

$$\int_0^1 \exp\left(2\pi i y\right)\xi(y)dy = \int_0^1 \exp\left(2\pi i(y+\tfrac{1}{k})\right)\xi(y)dy = \exp\left(2\pi i/k\right)\int_0^1 \exp\left(2\pi i y\right)\xi(y)dy$$

from which it follows that

$$\int_0^1 \exp\left(2\pi i y\right)\xi(y)dy = 0.$$

By taking the real part, we see that (7) is 0.

We complete the proof by noting that for a large enough N, the Chernoff bound shows that z is within ± 0.001 of (7) with probability exponentially close to 1. ∎

The following lemma shows that being able to extract $\mathbf{s}$ from samples of $A_{\mathbf{s},\psi_\alpha}$ implies being able to extract $\mathbf{s}$ from samples of $A_{\mathbf{s},\psi_\beta}$ for any $\beta < \alpha$, even without knowing β.

Lemma 3.6 *Assume there exists an efficient algorithm W that solves $\mathsf{LWE}_{p,\psi_\alpha}$ for some $\alpha < 1$. Then, there exists an algorithm W' that, given samples from $A_{\mathbf{s},\psi_\beta}$ for some (unknown) $\beta < \alpha$, outputs $\mathbf{s}$ with probability exponentially close to 1.*

Proof: The proof is based on the following idea: by adding the right amount of noise, we can transform samples from $A_{\mathbf{s},\psi_\beta}$ to samples from $A_{\mathbf{s},\psi_\alpha}$. Assume that the number of samples required by W is at most n^c for some $c > 0$. Let Z be the set of all integer multiples of $n^{-2c}\alpha$ between 0 and α. For each $\gamma \in Z$, Algorithm W' does the following n times. It takes n^c samples from $A_{\mathbf{s},\psi_\beta}$ and adds to the second register of each sample a noise sampled independently from ψ_γ. This creates n^c samples taken from the distribution $A_{\mathbf{s},\psi_{\sqrt{\beta^2+\gamma^2}}}$. It then applies W and obtains some candidate $\mathbf{s}'$. Using Lemma 3.5, it checks whether $\mathbf{s}' = \mathbf{s}$ with confidence exponentially close to 1. If $\mathbf{s}' = \mathbf{s}$, it outputs the answer $\mathbf{s}'$. Otherwise, it continues.

We now show that W' finds $\mathbf{s}$ with probability exponentially close to 1. By Lemma 3.5, if W' outputs some value, then this value is correct with probability exponentially close to 1. Hence, it is enough to show that in one of the iterations, W' outputs some value. Consider the smallest $\gamma \in Z$ such that $\gamma \geq \sqrt{\alpha^2-\beta^2}$. Clearly, $\gamma \leq \sqrt{\alpha^2-\beta^2} + n^{-2c}\alpha$. Define $\alpha' = \sqrt{\beta^2+\gamma^2}$. Then,

$$\alpha \leq \alpha' \leq \sqrt{\alpha^2 + 2n^{-2c}\alpha\sqrt{\alpha^2-\beta^2} + n^{-4c}\alpha^2} \leq \alpha\sqrt{1+3n^{-2c}} \leq \alpha(1+3n^{-2c}).$$

By Claim A.2, the statistical distance between Ψ_α and $\Psi_{\alpha'}$ is at most $18n^{-2c}$. Hence, the statistical distance between n^c samples from Ψ_α and n^c samples from $\Psi_{\alpha'}$ is at most $18n^{-c}$. Therefore, for our choice of γ, W outputs $\mathbf{s}$ with probability at least $1 - 9n^{-c} - 2^{-\Omega(n)} \geq \frac{1}{2}$. The probability that $\mathbf{s}$ is not found in any of the n calls to W is at most 2^{-n}. ∎

The following lemma shows that in order to solve CVP, it is enough to be able to find the coefficients modulo p of the closest lattice point.

Lemma 3.7 *Let L be an n-dimensional lattice, let d be a positive real number smaller than $\lambda_1(L)/2$, and let $p \geq 2$ be some integer. Assume we are given an oracle that given a point $\mathbf{x} \in \mathbb{R}^n$ within distance d of L, finds $L^{-1}\kappa_L(\mathbf{x}) \bmod p \in \mathbb{Z}_p^n$. Then, there exists an algorithm that solves $\mathsf{CVP}_{L,d}$ using this oracle.*

Proof: We define a sequence of points $\mathbf{x}_1 = \mathbf{x}, \mathbf{x}_2, \mathbf{x}_3, \ldots$ as follows. Let $\mathbf{a}_i = L^{-1}\kappa_L(\mathbf{x}_i) \in \mathbb{Z}^n$ be the coefficient vector of the closest lattice point to $\mathbf{x}_i$. We define $\mathbf{x}_{i+1} = (\mathbf{x}_i - L(\mathbf{a}_i \bmod p))/p$. Notice that the closest lattice point to $\mathbf{x}_{i+1}$ is $L(\mathbf{a}_i - (\mathbf{a}_i \bmod p))/p \in L$ and hence $\mathbf{a}_{i+1} = (\mathbf{a}_i - (\mathbf{a}_i \bmod p))/p$. Moreover, the distance of $\mathbf{x}_{i+1}$ from L is at most d/p^i. Also note that this sequence can be computed by using the oracle.

After n steps, we have a point $\mathbf{x}_{n+1}$ whose distance to the lattice is at most d/p^n. We now apply an algorithm for approximately solving the closest vector problem, such as Babai's nearest plane algorithm [6]. This yields a lattice point $L\mathbf{a}$ within distance $2^n \cdot d/p^n \leq d < \lambda_1(L)/2$ of $\mathbf{x}_{n+1}$. Hence, $L\mathbf{a}$ is the lattice point closest to $\mathbf{x}_{n+1}$ and we managed to recover $\mathbf{a}_{n+1} = \mathbf{a}$. Knowing $\mathbf{a}_{n+1}$ and $\mathbf{a}_n \bmod p$ (by using the oracle), we can now recover $\mathbf{a}_n = p\mathbf{a}_{n+1} + (\mathbf{a}_n \bmod p)$. Continuing this process, we can recover $\mathbf{a}_{n-1}, \mathbf{a}_{n-2}, \ldots, \mathbf{a}_1$. This completes the algorithm since $L\mathbf{a}_1$ is the closest point to $\mathbf{x}_1 = \mathbf{x}$. ■

Lemma 3.8 *Let L be an n-dimensional lattice, let $\alpha = \alpha(n) < 1$ be some positive real, and let $p \geq 2$ be some integer. Assume we can sample from $D_{L,1}$. Assume, moreover, that $\eta_\varepsilon(L) \leq 1/p$ for some negligible function $\varepsilon = \varepsilon(n)$. Also assume that we have an oracle W that finds $\mathbf{s}$ given samples from $A_{\mathbf{s},\psi_\beta}$ for all $\beta < \sqrt{2}\alpha$. Then, given any $\mathbf{x}$ point within distance αp of L^*, we can find $(L^*)^{-1}\kappa_{L^*}(\mathbf{x}) \bmod p \in \mathbb{Z}_p^n$, the coefficient vector of the closest vector to $\mathbf{x}$ reduced modulo p.*

Proof: We describe a procedure that given $\mathbf{x}$ as above outputs samples from the distribution $A_{\mathbf{s},\psi_\beta}$ for some $\beta < \sqrt{2}\alpha$ where $\mathbf{s} = (L^*)^{-1}\kappa_{L^*}(\mathbf{x}) \bmod p$. By performing this procedure a polynomial number of times and then using W, we can find $\mathbf{s}$.

The procedure works as follows. We first sample a vector $\mathbf{v} \in L$ from $D_{L,1}$. Let $\mathbf{a} = L^{-1}\mathbf{v} \bmod p$. We then compute $r = \langle \mathbf{x}, \mathbf{v}\rangle/p + e \bmod 1$ where $e \in \mathbb{R}$ is chosen according to a normal distribution with standard deviation $\alpha/\sqrt{2\pi}$. The output of the procedure is $(\mathbf{a}, r)$. We claim that the distribution given by this procedure is exponentially close to $A_{\mathbf{s},\psi_\beta}$ for some $\beta < \sqrt{2}\alpha$.

Let us first show that the distribution of $\mathbf{a}$ is very close to uniform. Fix some $\mathbf{a} \in \mathbb{Z}_p^n$. Then, the probability of obtaining this $\mathbf{a}$ is proportional to

$$\begin{aligned}
\sum_{\mathbf{x} \in pL + L\mathbf{a}} \rho(\mathbf{x}) &= \sum_{\mathbf{x} \in pL} \rho_{-L\mathbf{a}}(x) \\
&= p^{-n}\det(L^*) \sum_{\mathbf{y} \in L^*/p} \widehat{\rho_{-L\mathbf{a}}}(\mathbf{y}) \qquad \text{(by Lemma 2.5)} \\
&= p^{-n}\det(L^*) \sum_{\mathbf{y} \in L^*/p} \exp\left(-2\pi i \langle L\mathbf{a}, \mathbf{y}\rangle\right)\widehat{\rho}(\mathbf{y}) \\
&= p^{-n}\det(L^*)(1 \pm \varepsilon) \qquad \text{(since } \eta_\varepsilon(pL) = p\eta_\varepsilon(L) \leq 1\text{)}\,.
\end{aligned}$$

This implies that statistical distance between the distribution on $\mathbf{a}$ and the uniform distribution is negligible.

Next, we consider the distribution of r conditioned on some fixed value of $\mathbf{a}$. Define $\mathbf{x}' = \mathbf{x} - \kappa_{L^*}(\mathbf{x})$. Then,

$$\langle \mathbf{x}, \mathbf{v}\rangle/p + e \bmod 1 = \langle \mathbf{x}', \mathbf{v}\rangle/p + e + \langle \kappa_{L^*}(\mathbf{x}), \mathbf{v}\rangle/p \bmod 1.$$

Now,

$$\langle \kappa_{L^*}(\mathbf{x}), \mathbf{v}\rangle = \langle L^* \cdot (L^*)^{-1} \cdot \kappa_{L^*}(\mathbf{x}), L \cdot L^{-1} \cdot \mathbf{v}\rangle = \langle (L^*)^{-1}\kappa_{L^*}(\mathbf{x}), L^{-1}\mathbf{v}\rangle$$

where the last equality follows since $(L^*)^T = L^{-1}$. In words, this says that the inner product between $\kappa_{L^*}(\mathbf{x})$ and $\mathbf{v}$ (and in fact, between any vector in L and a vector in L^*) is the same as the inner product between the corresponding coefficient vectors. Since the coefficient vectors are integer,

$$\langle \kappa_{L^*}(\mathbf{x}), \mathbf{v}\rangle \bmod p = \langle \mathbf{s}, \mathbf{a}\rangle \bmod p$$

from which it follows that $\langle \kappa_{L^*}(\mathbf{x}), \mathbf{v}\rangle/p \bmod 1$ is exactly $\langle \mathbf{s}, \mathbf{a}\rangle/p \bmod 1$. Hence, in the following it is enough to show that the statistical distance between the distribution of $\langle \mathbf{x}', \mathbf{v}\rangle/p + e \bmod 1$ and that of ψ_β is negligible for some $\beta < \sqrt{2}\alpha$. In fact, we will show the stronger statement that the statistical distance between the distribution of $\langle \mathbf{x}', \mathbf{v}\rangle/p + e$ and that of a normal variable with mean 0 and standard deviation $\beta/\sqrt{2\pi}$ is negligible for some $\beta < \sqrt{2}\alpha$ (this statement is stronger since statistical distance cannot increase by applying a function, $\bmod 1$ in this case).

So let us consider the distribution of $\langle \mathbf{x}', \mathbf{v}\rangle/p+e$. Notice that conditioned on the value $\mathbf{a}$, the distribution of $\mathbf{v}$ is $D_{pL+L\mathbf{a}}$. It can be seen that the distribution of $\langle \mathbf{x}', \mathbf{v}\rangle/p + e$ is identical to the distribution obtained by sampling $\mathbf{v}$ from $D_{pL+L\mathbf{a}}$ and a noise vector $\mathbf{h}$ from $\rho_{p\alpha/\|\mathbf{x}'\|}/(p\alpha/\|\mathbf{x}'\|)^n$ and outputting $\langle \mathbf{x}', \mathbf{v} + \mathbf{h}\rangle/p$. This holds since $\langle \mathbf{x}', \mathbf{h}\rangle/p$ is distributed like a normal variable with standard deviation $\alpha/\sqrt{2\pi}$.

Lemma 3.9 tells us that the statistical distance between $\mathbf{v} + \mathbf{h}$ and $\nu_{\sqrt{1+(p\alpha/\|\mathbf{x}'\|)^2}}$ is negligible. Since statistical distance cannot increase by applying a function, we obtain that the statistical distance between $\langle \mathbf{x}', \mathbf{v} + \mathbf{h}\rangle/p$ and a normal variable with mean 0 and standard deviation

$$\frac{1}{\sqrt{2\pi}} \cdot \sqrt{1 + (p\alpha/\|\mathbf{x}'\|)^2} \cdot \|\mathbf{x}'\|/p = \frac{1}{\sqrt{2\pi}} \cdot \sqrt{(\|\mathbf{x}'\|/p)^2 + \alpha^2} \le \frac{1}{\sqrt{2\pi}} \cdot \sqrt{2}\alpha$$

is negligible. ■

Lemma 3.9 *Let L be a lattice, let $\mathbf{u} \in \mathbb{R}^n$ be any vector, let $s, t > 0$ be two reals, and let r denote $\sqrt{s^2 + t^2}$. Assume that $\varepsilon := \rho_{r/st}(L^* \setminus \{\mathbf{0}\})$ satisfies $\varepsilon < \frac{1}{2}$. Consider the continuous distribution Y on $\mathbb{R}^n$ obtained by sampling from $D_{L+\mathbf{u},t}$ and then adding a noise vector taken from ν_s. Then, the statistical distance between Y and ν_r is at most 4ε.*

Proof: The probability density function of Y can be written as

$$\begin{aligned}
Y(\mathbf{x}) &= \frac{1}{s^n \rho_t(L+\mathbf{u})} \sum_{\mathbf{y}\in L+\mathbf{u}} \rho_t(\mathbf{y})\rho_s(\mathbf{x}-\mathbf{y}) \\
&= \frac{1}{s^n \rho_t(L+\mathbf{u})} \sum_{\mathbf{y}\in L+\mathbf{u}} \exp\left(-\pi(\|\mathbf{y}/t\|^2 + \|(\mathbf{x}-\mathbf{y})/s\|^2)\right) \\
&= \frac{1}{s^n \rho_t(L+\mathbf{u})} \sum_{\mathbf{y}\in L+\mathbf{u}} \exp\left(-\pi\Big(\frac{s^2+t^2}{s^2\cdot t^2} \cdot \Big\|\mathbf{y} - \frac{t^2}{s^2+t^2}\mathbf{x}\Big\|^2 + \frac{1}{s^2+t^2}\|\mathbf{x}\|^2\Big)\right) \\
&= \exp\left(-\frac{\pi}{s^2+t^2}\|\mathbf{x}\|^2\right) \frac{1}{s^n \rho_t(L+\mathbf{u})} \sum_{\mathbf{y}\in L+\mathbf{u}} \exp\left(-\pi\Big(\frac{s^2+t^2}{s^2\cdot t^2} \cdot \Big\|\mathbf{y} - \frac{t^2}{s^2+t^2}\mathbf{x}\Big\|^2\Big)\right) \\
&= \frac{1}{s^n}\rho_r(\mathbf{x}) \cdot \frac{\rho_{st/r,(t/r)^2\mathbf{x}-\mathbf{u}}(L)}{\rho_{t,-\mathbf{u}}(L)} \\
&= \frac{1}{s^n}\rho_r(\mathbf{x}) \cdot \frac{\widehat{\rho_{st/r,(t/r)^2\mathbf{x}-\mathbf{u}}}(L^*)}{\widehat{\rho_{t,-\mathbf{u}}}(L^*)} \\
&= \rho_r(\mathbf{x})/r^n \cdot \frac{(r/st)^n \widehat{\rho_{st/r,(t/r)^2\mathbf{x}-\mathbf{u}}}(L^*)}{(1/t)^n \widehat{\rho_{t,-\mathbf{u}}}(L^*)} \qquad (8)
\end{aligned}$$

where in the second to last equality we used Lemma 2.5. Using Equation (5),

$$\widehat{\rho_{st/r,(t/r)^2\mathbf{x}-\mathbf{u}}}(\mathbf{w}) = \exp\left(-2\pi i\langle (t/r)^2\mathbf{x} - \mathbf{u}, \mathbf{w}\rangle\right) \cdot (st/r)^n \rho_{r/st}(\mathbf{w})$$
$$\widehat{\rho}_{t,-\mathbf{u}}(\mathbf{w}) = \exp\left(-2\pi i\langle -\mathbf{u}, \mathbf{w}\rangle\right) \cdot t^n \rho_{1/t}(\mathbf{w})$$

By our assumption,

$$\left|1 - (r/st)^n \widehat{\rho_{st/r,(t/r)^2\mathbf{x}-\mathbf{u}}}(L^*)\right| \le \rho_{r/st}(L^* \setminus \{0\}) = \varepsilon$$
$$\left|1 - (1/t)^n \widehat{\rho_{t,-\mathbf{u}}}(L^*)\right| \le \rho_{1/t}(L^* \setminus \{0\}) \le \varepsilon$$

where the last inequality follows from $1/t \le r/st$. Hence, the quotient in (8) is between $(1-\varepsilon)/(1+\varepsilon) \ge 1 - 2\varepsilon$ and $(1+\varepsilon)/(1-\varepsilon) \le 1 + 4\varepsilon$. This implies that,

$$|Y(\mathbf{x}) - \rho_r(\mathbf{x})/r^n| \le \rho_r(\mathbf{x})/r^n \cdot 4\varepsilon$$

We complete the proof by integrating over $\mathbb{R}^n$. ■

3.2.2 From CVP to samples

We start with two lemmas by Banaszczyk followed by a technical claim.

Lemma 3.10 ([7], Lemma 1.4(i)) *For any lattice L and $a \ge 1$, $\rho_a(L) \le a^n \rho(L)$.*

Lemma 3.11 ([7], Lemma 1.5(i)) *Let B_n denote the Euclidean unit ball. Then, for any lattice L, $\rho(L \setminus \sqrt{n}B_n) < 2^{-2n} \cdot \rho(L)$.*

For clarity, we omit the normalization factors from quantum states.

Claim 3.12 *Let L be some n-dimensional lattice such that $\lambda_1(L) > 2\sqrt{n}$ and let $M > 0$ be some integer. Then, the quantum states*

$$|\vartheta_1\rangle = \sum_{\mathbf{x} \in L/M, \|\mathbf{x}\| < \sqrt{n}} \rho(\mathbf{x}) |\mathbf{x} \bmod L\rangle$$

and

$$|\vartheta_2\rangle = \sum_{\mathbf{x} \in \mathcal{P}(L) \cap L/M} \sum_{\mathbf{y} \in L} \rho(\mathbf{x} - \mathbf{y}) |\mathbf{x}\rangle$$

are exponentially close.

Proof: Consider $|\vartheta_1\rangle$ and $|\vartheta_2\rangle$ as unnormalized vectors in M^n-dimensional space. Let Z be the l_2 norm of $|\vartheta_1\rangle$. In the following we show that the l_2 distance between $|\vartheta_1\rangle$ and $|\vartheta_2\rangle$ is at most $2^{-\Omega(n)}Z$. This is enough to establish that the l_2 norm between the normalized quantum states corresponding to $|\vartheta_1\rangle$ and $|\vartheta_2\rangle$ is exponentially small.

We first obtain a good estimate of Z.

$$Z = \sum_{\mathbf{x} \in L/M, \|\mathbf{x}\| < \sqrt{n}} \rho(\mathbf{x})^2 = \sum_{\mathbf{x} \in L/M, \|\mathbf{x}\| < \sqrt{n}} \rho_{1/\sqrt{2}}(\mathbf{x})$$
$$= \rho_{1/\sqrt{2}}(L/M \cap \sqrt{n}B_n) = \rho(\sqrt{2}L/M \cap \sqrt{2n}B_n).$$

In particular, we obtain that

$$\rho(\sqrt{2}L/M \cap \sqrt{n}B_n) \leq Z \leq \rho(\sqrt{2}L/M).$$

Lemma 3.11 with the lattice $\sqrt{2}L/M$ implies that the difference between these two bounds on Z is at most $2^{-2n}\rho(\sqrt{2}L/M)$. Hence,

$$1 - 2^{-2n} \leq Z/\rho(\sqrt{2}L/M) \leq 1.$$

We complete the proof with an upper bound the l_2 distance between the two vectors. Using the monotonicity of norms,

$$\begin{aligned}
\| |\vartheta_1\rangle - |\vartheta_2\rangle \|_2 &\leq \| |\vartheta_1\rangle - |\vartheta_2\rangle \|_1 \\
&= \sum_{\mathbf{x} \in \mathcal{P}(L) \cap L/M} \sum_{\mathbf{y} \in L, \|\mathbf{x}-\mathbf{y}\| \geq \sqrt{n}} \rho(\mathbf{x}-\mathbf{y}) \\
&= \sum_{\mathbf{x} \in L/M, \|\mathbf{x}\| \geq \sqrt{n}} \rho(\mathbf{x}) \\
&\leq 2^{-2n}\rho(L/M) \\
&\leq 2^{-2n}2^{n/2}\rho(\sqrt{2}L/M) \qquad \text{(by Lemma 3.10)} \\
&\leq 2^{-n}\rho(\sqrt{2}L/M).
\end{aligned}$$

■

We can now prove the main lemma of this subsection.

Lemma 3.13 *Let L be an n-dimensional lattice and assume that there exists an oracle that answers $\mathsf{CVP}_{L^*,d}$ queries for some $d < \lambda_1(L^*)/2$. Then, there exists a quantum algorithm for sampling from the distribution $D_{L,\sqrt{n}/d}$.*

Proof: By scaling, we can assume without loss of generality that $d = \sqrt{2n}$. Let M be a large enough integer. More specifically, any $M \geq 2^{n^2}\lambda_n(L^*)$ is large enough. Notice that such an M can be computed in polynomial time from the lattice L, and hence $\log M$ is polynomial. Our first step is to create a state exponentially close to

$$\sum_{\mathbf{x} \in \mathcal{P}(L^*) \cap L^*/M} \sum_{\mathbf{y} \in L^*} \rho(\mathbf{x}-\mathbf{y})|\mathbf{x}\rangle. \tag{9}$$

This is a state on $n \log M$ qubits, a number that is polynomial in the input size. We start by creating the quantum state

$$\sum_{\mathbf{x} \in L^*/M, \|\mathbf{x}\| < \sqrt{n}} \rho(\mathbf{x})|\mathbf{x}\rangle.$$

An exponentially good approximation of this state can be created efficiently. Since this can be done using known techniques, we omit further details. We next compute $\mathbf{x} \bmod \mathcal{P}(L^*)$ in a new register and obtain

$$\sum_{\mathbf{x} \in L^*/M, \|\mathbf{x}\| < \sqrt{n}} \rho(\mathbf{x})|\mathbf{x}, \mathbf{x} \bmod \mathcal{P}(L^*)\rangle.$$

Using the CVP oracle, we can recover $\mathbf{x}$ from $\mathbf{x} \bmod \mathcal{P}(L^*)$. This allows us to uncompute the first register and obtain

$$\sum_{\mathbf{x}\in L^*/M, \|\mathbf{x}\|<\sqrt{n}} \rho(\mathbf{x})|\mathbf{x} \bmod \mathcal{P}(L^*)\rangle.$$

Using Claim 3.12, this state is exponentially close to the required state (9).

In the second step, we apply the quantum Fourier transform. First, using the natural mapping between $\mathcal{P}(L^*) \cap L^*/M$ and $\mathbb{Z}_M^n$, we can rewrite (9) as

$$\sum_{\mathbf{s}\in\mathbb{Z}_M^n} \sum_{\mathbf{y}\in L^*} \rho(L^*\mathbf{s}/M - \mathbf{y})|\mathbf{s}\rangle = \sum_{\mathbf{s}\in\mathbb{Z}_M^n} \sum_{\mathbf{r}\in\mathbb{Z}^n} \rho(L^*\mathbf{s}/M - L^*\mathbf{r})|\mathbf{s}\rangle.$$

We now apply the quantum Fourier transform on $\mathbb{Z}_M^n$. We obtain a state in which the amplitude of $|\mathbf{t}\rangle$ for $\mathbf{t} \in \mathbb{Z}_M^n$ is proportional to

$$\begin{aligned}
&\sum_{\mathbf{s}\in\mathbb{Z}_M^n} \sum_{\mathbf{r}\in\mathbb{Z}^n} \rho(L^*\mathbf{s}/M - L^*\mathbf{r}) \exp(2\pi i\langle \mathbf{s}, \mathbf{t}\rangle/M) \\
&\quad = \sum_{\mathbf{s}\in\mathbb{Z}^n} \rho(L^*\mathbf{s}/M) \exp(2\pi i\langle \mathbf{s}, \mathbf{t}\rangle/M) \\
&\quad = \sum_{\mathbf{x}\in L^*/M} \rho(\mathbf{x}) \exp(2\pi i\langle (L^*)^{-1}\mathbf{x}, \mathbf{t}\rangle) \\
&\quad = \sum_{\mathbf{x}\in L^*/M} \rho(\mathbf{x}) \exp(2\pi i\langle \mathbf{x}, L\mathbf{t}\rangle) \\
&\quad = \det(ML) \sum_{\mathbf{y}\in ML} \rho(\mathbf{y} - L\mathbf{t})
\end{aligned}$$

where the last equality follows from Lemma 2.5 and (6). Hence, the resulting state can be equivalently written as

$$\sum_{\mathbf{x}\in\mathcal{P}(ML)\cap L} \sum_{\mathbf{y}\in ML} \rho(\mathbf{y} - \mathbf{x})|\mathbf{x}\rangle.$$

Notice that $\lambda_1(ML) = M\lambda_1(L) \geq M/\lambda_n(L^*) \geq 2^{n^2}$. Hence, we can apply Claim 3.12 to the lattice ML, and obtain that this state is exponentially close to

$$\sum_{\mathbf{x}\in L, \|\mathbf{x}\|<\sqrt{n}} \rho(x)|\mathbf{x} \bmod ML\rangle.$$

We measure this state and obtain $\mathbf{x} \bmod ML$ for some vector $\mathbf{x}$ with $\|\mathbf{x}\| \leq \sqrt{n}$. Since $\mathbf{x} \bmod ML$ is within $\sqrt{n}$ of the lattice ML, and $\lambda_1(ML) \geq 2^{n^2}$, we can recover $\mathbf{x}$ by using, say, Babai's nearest plane algorithm [6]. The output of the algorithm is $\mathbf{x}$.

We claim that the distribution of $\mathbf{x}$ is exponentially close to $D_{L,1/\sqrt{2}}$. Indeed, the probability of obtaining any $\mathbf{x} \in L, \|\mathbf{x}\| < \sqrt{n}$ is proportional to $\rho(\mathbf{x})^2 = \rho_{1/\sqrt{2}}(\mathbf{x})$. It remains to notice that by Lemma 3.11, all but an exponentially small fraction of the probability distribution $D_{L,1/\sqrt{2}}$ is on points of norm less than $\sqrt{n}$. ∎

3.3 Termination

We first argue that the algorithm terminates in polynomial time. In each of the $\lceil \log p \rceil = O(\log n)$ iterations of the outer loop, the algorithm begins with $r \le 2^{cn}\tilde{\lambda}_n \cdot 2p \le 2^{(c+2)n}\lambda_n(L)$. In each iteration of the inner loop, r gets smaller by a factor of at least 2. Having n linearly independent vectors of length at most $r\sqrt{n}$ implies that $r\sqrt{n} \ge \lambda_n(L)$. Hence, the number of iterations of the inner loop is at most $(c+2)n + \frac{1}{2}\log n = O(n)$.

Next, we show that as long as $r \ge 2\eta_\varepsilon(L)$, n^2 samples from $D_{L,r}$ contain n linearly independent vectors of length at most $r\sqrt{n}$ with probability exponentially close to 1. By Lemma 3.11, with probability exponentially close to 1, a vector sampled from $D_{L,r}$ has length at most $r\sqrt{n}$ (this, in fact, holds for any r). We use the following lemma from [18].

Lemma 3.14 *Let L be any n-dimensional lattice, let r be such that $r \ge 2\eta_\varepsilon(L)$ where $\varepsilon \le \frac{1}{100}$. Then for any $n-1$ dimensional hyperplane H the probability that $\mathbf{x} \notin H$ where $\mathbf{x}$ is chosen from $D_{L,r}$ is at least $\frac{1}{100}$.*

From this we can easily obtain the following.

Lemma 3.15 *Let L be any n-dimensional lattice and let $r \ge 2\eta_\varepsilon(L)$ where $\varepsilon \le \frac{1}{100}$. Then, the probability that a set of n^2 vectors chosen from $D_{L,r}$ contains no n linearly independent vectors is exponentially small.*

Proof: Let each of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ contain n vectors chosen from $D_{L,r}$. Then, we claim that

$$\Pr[\dim \operatorname{span}(\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n) = n] > 1 - 2^{-\Omega(n)}.$$

It is enough to show that for $i = 1, \ldots, n$,

$$\Pr[\dim \operatorname{span}(\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_i) \ge i \mid \mathcal{A}_1, \ldots, \mathcal{A}_{i-1}] > 1 - 2^{-\Omega(n)}$$

where we condition on some arbitrary $\mathcal{A}_1, \ldots, \mathcal{A}_{i-1}$ whose span is of dimension $i-1$. This follows from Lemma 3.14 with $H = \operatorname{span}(\mathcal{A}_1, \ldots, \mathcal{A}_{i-1})$ since $(99/100)^n = 2^{-\Omega(n)}$. ■

To complete the analysis, we now show that the algorithm finds n linearly independent vectors of length at most $2n\eta_\varepsilon(L)/\alpha$. Let $k \in \{1, 2, \ldots, \lceil \log p \rceil\}$ be such that there exists an $i \in \mathbb{Z}$ for which

$$2^{4n}\tilde{\lambda}_n 2^k \left(\frac{\sqrt{n}}{\alpha p}\right)^i \in [p\eta_\varepsilon(L), 2p\eta_\varepsilon(L)).$$

Such a k must exist since $\alpha p/\sqrt{n} \le p \le 2^{\lceil \log p \rceil}$. Consider the kth iteration of the outer loop. Each time we reach Step 6, the set $\mathcal{S}_k$ stores n linearly independent vectors of length at most $r\sqrt{n}$. Hence, in the following, it is enough to show that we reach Step 6 with $r \le 2\sqrt{n}\eta_\varepsilon(L)/\alpha$.

The inner loop starts with $r = 2^{4n}\tilde{\lambda}_n 2^k$ and in each iteration, this value gets multiplied by $\sqrt{n}/(\alpha p)$. We claim that the last iteration of the inner loop must start with $r < p\eta_\varepsilon(L)$. Indeed, consider any iteration that starts with $r \ge p\eta_\varepsilon(L)$. The condition of Lemma 3.3 is satisfied, and Step 8 produces a set of $n^c \ge n^2$ samples from the distribution $D_{L,r'}$ for $r' = r\sqrt{n}/(\alpha p)$. Since $r' \ge 2\eta_\varepsilon(L)$, the arguments above imply that this set contains n linearly independent vectors of length at most $r'\sqrt{n}$. Hence, such an iteration cannot be the last one.

We obtain that in the last iteration of the loop, $r < p\eta_\varepsilon(L)$. By our choice of k, this r must satisfy that $r \le 2p\eta_\varepsilon(L)\sqrt{n}/(\alpha p) = 2\sqrt{n}\eta_\varepsilon(L)/\alpha$. This concludes the proof.

4 Some Reductions

In this section, we consider several variants of the LWE problem. Through a sequence of elementary reductions, we prove that all problems are as hard as LWE. The results of this section are summarized in Lemma 4.4.

Lemma 4.1 (Average-case to Worst-case) *Assume there exists 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 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 $\mathbf{t} \in \mathbb{Z}_p^n$ consider the function $f_{\mathbf{t}} : \mathbb{Z}_p^n \times \mathbb{Z}_p \to \mathbb{Z}_p^n \times \mathbb{Z}_p$ 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.

Assume that for n^{-c_1} of all possible $\mathbf{s}$, the acceptance probability of W on inputs from $A_{\mathbf{s},\chi}$ and on inputs from U differ by at least n^{-c_2}. We construct W' as follows. Let R denote the unknown input distribution. Repeat the following n^{c_1+1} times. Choose a vector $\mathbf{t} \in \mathbb{Z}_p^n$ uniformly at random. Then estimate the acceptance probability of W on U and on $f_{\mathbf{t}}(R)$ by calling W $O(n^{c_2+1})$ times on each of the input distributions. This allows us to obtain an estimate that with probability exponentially close to 1 is within $\pm n^{-c_2}/8$ of the true acceptance probabilities. If the two estimates differ by more than $n^{-c_2}/2$ then we stop and decide to accept. Otherwise we continue. If the procedure ends without accepting, we reject.

We now prove that W' distinguishes $A_{\mathbf{s},\chi}$ from U for all $\mathbf{s}$. First, we claim that when R is U, the acceptance probability of W' is exponentially close to 0. Indeed, in this case, $f_{\mathbf{t}}(U) = U$ and therefore the two estimates that W' performs are of the same distribution. The probability that the estimates differ by more than $n^{-c_2}/2 > 2 \cdot n^{-c_2}/8$ is exponentially small. Next, consider the case that R is $A_{\mathbf{s},\chi}$ for some $\mathbf{s}$. In each of the n^{c_1+1} iterations, we are considering the distribution $f_{\mathbf{t}}(A_{\mathbf{s},\chi}) = A_{\mathbf{s}+\mathbf{t},\chi}$ for some uniformly chosen $\mathbf{t}$. Notice that the distribution of $\mathbf{s}+\mathbf{t}$ is uniform on $\mathbb{Z}_p^n$. Hence, with probability exponentially close to 1, in one of the n^{c_1+1} iterations, $\mathbf{t}$ is such that the acceptance probability of W on inputs from $A_{\mathbf{s}+\mathbf{t},\chi}$ and on inputs from U differ by at least n^{-c_2}. Since our estimates are within $\pm n^{-c_2}/8$, W' accepts with probability exponentially to 1. ■

Lemma 4.2 (Decision to Search) *Let $n \geq 1$ be some integer, $p \geq 2$ be a prime, and χ be some distribution on $\mathbb{Z}_p$. Assume there exists an 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. Then, there exists an algorithm W' that, given samples from $A_{\mathbf{s},\chi}$ for some* $\mathbf{s}$*, outputs* $\mathbf{s}$ *with probability exponentially close to* 1.

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

Hence, using W, we can test whether $k = s_1$. Since there are only $p < \mathrm{poly}(n)$ possibilities for s_1 we can try all of them. ■

Lemma 4.3 (Discrete to Continuous) *Let ϕ be some probability density function on $\mathbb{T}$ and let $\bar{\phi}$ be its discretization to $\mathbb{Z}_p$. Assume there exists an algorithm W that solves $\mathsf{LWE}_{p,\bar{\phi}}$. Then, there exists an algorithm W' that solves $\mathsf{LWE}_{p,\phi}$.*

Proof: Algorithm W' simply takes samples from $A_{\mathbf{s},\phi}$ and discretizes the second element to obtain samples from $A_{\mathbf{s},\bar{\phi}}$. It then applies W with these samples in order to find $\mathbf{s}$. ■

By combining the three lemmas above, we obtain

Lemma 4.4 *Let $n \geq 1$ be some integer and $p \geq 2$ be a prime. Let ϕ be some probability density function on $\mathbb{T}$ and let $\bar{\phi}$ be its discretization to $\mathbb{Z}_p$. Then, assume there exists a distinguisher that distinguishes $A_{\mathbf{s},\bar{\phi}}$ from U for a non-negligible fraction of all possible $\mathbf{s}$. Then, there exists an algorithm that solves $\mathsf{LWE}_{p,\phi}$.*

5 Public Key Cryptosystem

We let n be the security parameter of the cryptosystem. Our cryptosystem is parameterized by two integers m, p and a probability distribution χ on $\mathbb{Z}_p$. A setting of these parameters that guarantees both security and correctness is the following. Choose $m = 5n$ and set $p \geq 2$ to be some prime number between n^2 and $2n^2$. The probability distribution χ is taken to be $\bar{\Psi}_{\alpha(n)}$ for $\alpha(n) = o(1/\sqrt{n \log n})$, i.e., $\alpha(n)$ is such that $\lim_{n\to\infty} \alpha(n)/\sqrt{n \log n} = 0$. For example, we can choose $\alpha(n) = 1/(\sqrt{n} \log n)$. In the following description, all additions are performed in $\mathbb{Z}_p$, i.e., modulo p.

- **Private key:** Choose $\mathbf{s} \in \mathbb{Z}_p^n$ uniformly at random. The private key is $\mathbf{s}$.
- **Public Key:** For $i = 1, \ldots, m$, choose m vectors $\mathbf{a_1}, \ldots, \mathbf{a_m} \in \mathbb{Z}_p^n$ independently from the uniform distribution. Also choose elements $e_1, \ldots, e_m \in \mathbb{Z}_p$ independently according to χ. The public key is given by $(\mathbf{a_i}, b_i)_{i=1}^m$ where $b_i = \langle \mathbf{a_i}, \mathbf{s} \rangle + e_i$.
- **Encryption:** In order to encrypt a bit we choose a random subset S of $[m]$. The encryption is $(\sum_{i\in S} \mathbf{a_i}, \sum_{i\in S} b_i)$ if the bit is 0 and $(\sum_{i\in S} \mathbf{a_i}, \lfloor \frac{p}{2} \rfloor + \sum_{i\in S} b_i)$ 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 $\lfloor \frac{p}{2} \rfloor$ modulo p. Otherwise, the decryption is 1.

Notice that with our choice of parameters, the public key size is $O(mn \log p) = \tilde{O}(n^2)$ and that the encryption process increases the size of a message by a factor of $O(n \log p) = \tilde{O}(n)$.

We next prove that under a certain condition on χ, m, and p, the probability of decryption error is small. We later show that our choice of parameters satisfies this condition. For the following two lemmas we need to introduce some additional notation. For a distribution χ on $\mathbb{Z}_p$ and an integer $k \geq 0$, we define $\chi^{\star k}$ as the distribution obtained by summing together k independent samples from χ, where addition is performed in $\mathbb{Z}_p$ (for $k = 0$ we define $\chi^{\star 0}$ as the distribution that is constantly 0). For a probability distribution ϕ on $\mathbb{T}$ we define $\phi^{\star k}$ similarly. For an element $a \in \mathbb{Z}_p$ we define $|a|$ as the integer a if $a \in \{0, 1, \ldots, \lfloor \frac{p}{2} \rfloor\}$ and as the integer $p - a$ otherwise. In other words, $|a|$ represents the distance of a from 0. Similarly, for $x \in \mathbb{T}$, we define $|x|$ as x for $x \in [0, \frac{1}{2}]$ and as $1 - x$ otherwise.

Lemma 5.1 (Correctness) *Let $\delta > 0$. Assume that for any $k \in \{0, 1, \ldots, m\}$, $\chi^{\star k}$ satisfies that*

$$\Pr_{e \sim \chi^{\star k}} \left[|e| < \left\lfloor \frac{p}{2} \right\rfloor / 2 \right] > 1 - \delta.$$

Then, the probability of decryption error is at most δ.

Proof: Consider first an encryption of 0. It is given by $(\mathbf{a}, b)$ for $\mathbf{a} = \sum_{i \in S} \mathbf{a_i}$ and

$$b = \sum_{i \in S} b_i = \sum_{i \in S} \langle \mathbf{a_i}, \mathbf{s} \rangle + e_i = \langle \mathbf{a}, \mathbf{s} \rangle + \sum_{i \in S} e_i.$$

Hence, $b - \langle \mathbf{a}, \mathbf{s} \rangle$ is exactly $\sum_{i \in S} e_i$. The distribution of the latter is $\chi^{\star |S|}$. According to our assumption, $|\sum_{i \in S} e_i|$ is less than $\lfloor \frac{p}{2} \rfloor / 2$ with probability at least $1 - \delta$. In this case, it is closer to 0 than to $\lfloor \frac{p}{2} \rfloor$ and therefore the decryption is correct. The proof for an encryption of 1 is similar. ■

Claim 5.2 *For our choice of parameters it holds that for any $k \in \{0, 1, \ldots, m\}$,*

$$\Pr_{e \sim \bar{\Psi}_\alpha^{\star k}} \left[|e| < \left\lfloor \frac{p}{2} \right\rfloor / 2 \right] > 1 - \delta(n)$$

for some negligible function $\delta(n)$.

Proof: A sample from $\bar{\Psi}_\alpha^{\star k}$ can be obtained by sampling $x_1, \ldots, x_k$ from Ψ_α and outputting $\sum_{i=1}^k \lfloor p x_i \rceil \bmod p$. Notice that this value is at most $k \leq m < p/32$ away from $\sum_{i=1}^k p x_i \bmod p$. Hence, it is enough to show that $|\sum_{i=1}^k p x_i \bmod p| < p/16$ with high probability. This condition is equivalent to the condition that $|\sum_{i=1}^k x_i| < 1/16$ where the sum is performed modulo 1. Since $\sum_{i=1}^k x_i$ is distributed as $\Psi_{\sqrt{k} \cdot \alpha}$, and $\sqrt{k} \cdot \alpha = o(1/\sqrt{\log n})$, the probability that $|\sum_{i=1}^k x_i| < 1/16$ is $1 - \delta(n)$ for some negligible function $\delta(n)$. ■

Next, we prove the security of the system. We need the following technical claim.

Claim 5.3 *Let G be some finite Abelian group and let $l = c \cdot \log |G|$. For $c \geq 5$, when choosing l elements $g_1, \ldots, g_l$ uniformly from G the probability that the statistical distance between the uniform distribution on G and the distribution given by sums of random subsets of $g_1, \ldots, g_l$ is more than $2/|G|$ is at most $1/|G|$.*

Proof: Let $X_{g,\mathbf{b}}$ for $g \in G, \mathbf{b} \in \{0,1\}^l \setminus \{0^l\}$ denote the event that $\sum_{i=1}^{c \cdot l} b_i g_i = g$ where the probability is taken over the choice of $\{g_1, \ldots, g_l\}$. Then, $E[X_{g,\mathbf{b}}] = 1/|G|$ and $V[X_{g,b}] < 1/|G|$. Hence,

$$E[Y_g] = \frac{2^l - 1}{|G|} = |G|^{c-1} - |G|^{-1}$$

where Y_g denotes $\sum_{\mathbf{b} \in \{0,1\}^l \setminus 0^l} X_{g,\mathbf{b}}$. Moreover, for any $\mathbf{b} \neq \mathbf{b}'$, the events $X_{g,\mathbf{b}}$ and $X_{g,\mathbf{b}'}$ are independent. Therefore,

$$V[Y_g] < \frac{2^l - 1}{|G|} < |G|^{c-1}.$$

Using the Chebyshev inequality,

$$\Pr\left(\left| Y_g - (|G|^{c-1} - |G|^{-1}) \right| \geq |G|^{\frac{c-1}{2} + 1} \right) \leq |G|^{-2}$$

and hence,

$$\Pr\left(\left|Y_g - |G|^{c-1}\right| \geq |G|^{\frac{c-1}{2}+1} + |G|^{-1}\right) \leq |G|^{-2}$$

Using the union bound,

$$\Pr\left(\exists g,\ \left|Y_g - |G|^{c-1}\right| \geq |G|^{\frac{c-1}{2}+1} + |G|^{-1}\right) \leq |G|^{-1}.$$

Therefore, with probability at least $1-|G|^{-1}$ on the choice of $\{g_1, \ldots, g_l\}$, the number of subsets (including the empty subset) mapped to each element g is at most

$$|G|^{\frac{c-1}{2}+1} + |G|^{-1} + 1 < 2|G|^{\frac{c-1}{2}+1}$$

away from $|G|^{c-1}$. This translates to a statistical distance of at most

$$2|G|^{\frac{c-1}{2}+1} \cdot |G|^{-(c-1)} \leq 2/|G|.$$

■

Lemma 5.4 (Security) *For $m \geq 5n$, if there exists a polynomial time algorithm W that distinguishes between encryptions of 0 and 1 then there exists a distinguisher Z that distinguishes between $A_{\mathbf{s},\chi}$ and U for a non-negligible fraction of all possible* $\mathbf{s}$.

Proof: Let $p_0(W)$ be the acceptance probability of W on input $((\mathbf{a_i}, b_i)_{i=1}^m, (\mathbf{a}, b))$ where $(\mathbf{a}, b)$ is an encryption of 0 with the public key $(\mathbf{a_i}, b_i)_{i=1}^m$ and the probability is taken over the randomness in the choice of the private and public keys and over the randomness in the encryption algorithm. We define $p_1(W)$ similarly for encryptions of 1 and let $p_u(W)$ be the acceptance probability of W on inputs $((\mathbf{a_i}, b_i)_{i=1}^m, (\mathbf{a}, b))$ where $(\mathbf{a_i}, b_i)_{i=1}^m$ are again chosen according to the private and public keys distribution but $(\mathbf{a}, b)$ is chosen uniformly from $\mathbb{Z}_p^n \times \mathbb{Z}_p$. With this notation, our hypothesis says that $|p_0(W) - p_1(W)| \geq \frac{1}{n^c}$ for some $c > 0$.

We now construct a W' for which $|p_0(W') - p_u(W')| \geq \frac{1}{2n^c}$. By our hypothesis, either $|p_0(W) - p_u(W)| \geq \frac{1}{2n^c}$ or $|p_1(W) - p_u(W)| \geq \frac{1}{2n^c}$. In the former case we take W' to be the same as W. In the latter case, we construct W' as follows. On input $((\mathbf{a_i}, b_i)_{i=1}^m, (\mathbf{a}, b))$, W' calls W with $((\mathbf{a_i}, b_i)_{i=1}^m, (\mathbf{a}, \frac{p-1}{2} + b))$. Notice that this maps the distribution on encryptions of 0 to the distribution on encryptions of 1 and the uniform distribution to itself. Therefore, W' is the required distinguisher.

Let $p_0(\mathbf{s})$ be the probability that W' accepts on input $((\mathbf{a_i}, b_i)_{i=1}^m, (\mathbf{a}, b))$ where $(\mathbf{a_i}, b_i)_{i=1}^m$ are chosen from $A_{\mathbf{s},\chi}$, and $(\mathbf{a}, b)$ is an encryption of 0 with the public key $(\mathbf{a_i}, b_i)_{i=1}^m$. Similarly, define $p_u(\mathbf{s})$ to be the acceptance probability of W' where $(\mathbf{a_i}, b_i)_{i=1}^m$ are chosen from $A_{\mathbf{s},\chi}$, and $(\mathbf{a}, b)$ is now chosen uniformly at random from $\mathbb{Z}_p^n \times \mathbb{Z}_p$. Define

$$Y = \left\{ \mathbf{s} \;\middle|\; |p_0(\mathbf{s}) - p_u(\mathbf{s})| \geq \frac{1}{4n^c} \right\}.$$

By an averaging argument we get that with probability at least a $\frac{1}{4n^c}$ fraction of the $\mathbf{s}$ are in Y for otherwise W' would have a gap of less than $\frac{1}{2n^c}$. Hence, it is enough to show a distinguisher Z that distinguishes between U and $A_{\mathbf{s},\chi}$ for any $\mathbf{s} \in Y$.

In the following we describe the distinguisher Z. We are given a distribution R that is either U or $A_{\mathbf{s},\chi}$ for some $\mathbf{s} \in Y$. We take m samples $(\mathbf{a_i}, b_i)_{i=1}^m$ from R. Let $p_0((\mathbf{a_i}, b_i)_{i=1}^m)$ be the probability that

W' accepts on input $((\mathbf{a_i}, b_i)_{i=1}^m, (\mathbf{a}, b))$ where the probability is taken only on the choice of $(\mathbf{a}, b)$ as an encryption of the bit 0 with the public key $(\mathbf{a_i}, b_i)_{i=1}^m$. Similarly, let $p_u((\mathbf{a_i}, b_i)_{i=1}^m)$ be the probability that W' accepts on input $((\mathbf{a_i}, b_i)_{i=1}^m, (\mathbf{a}, b))$ where the probability is taken over the choice of $(\mathbf{a}, b)$ as a random element of $\mathbb{Z}_p^n \times \mathbb{Z}_p$. The distinguisher Z estimates both $p_0((\mathbf{a_i}, b_i)_{i=1}^m)$ and $p_u((\mathbf{a_i}, b_i)_{i=1}^m)$ up to an additive error of $\frac{1}{64n^c}$. If the two estimates differ by more than $\frac{1}{16n^c}$, Z accepts. Otherwise, Z rejects.

We first claim that when R is the uniform distribution, Z rejects with high probability. In this case, $(\mathbf{a_i}, b_i)_{i=1}^m$ are chosen uniformly from $\mathbb{Z}_p^n \times \mathbb{Z}_p$. Using Claim 5.3 with the group $G = \mathbb{Z}_p^n \times \mathbb{Z}_p$, we obtain that with probability exponentially close to 1, the distribution on $(\mathbf{a}, b)$ obtained by encryptions of 0 is exponentially close to the uniform distribution on $\mathbb{Z}_p^n \times \mathbb{Z}_p$. Therefore, except with exponentially small probability,

$$|p_0((\mathbf{a_i}, b_i)_{i=1}^m) - p_u((\mathbf{a_i}, b_i)_{i=1}^m)|$$

is exponentially small. Hence, our two estimates differ by at most $\frac{1}{32n^c}$, and Z rejects.

Next, we show that if R is $A_{\mathbf{s},\chi}$ for $\mathbf{s} \in Y$ then Z accepts with probability $1/\mathrm{poly}(n)$. Notice that $p_0(\mathbf{s})$ (respectively, $p_u(\mathbf{s})$) is the average of $p_0((\mathbf{a_i}, b_i)_{i=1}^m)$ (respectively, $p_u((\mathbf{a_i}, b_i)_{i=1}^m)$) taken over the choice of $(\mathbf{a_i}, b_i)_{i=1}^m$ from $A_{\mathbf{s},\chi}$. From $|p_0(\mathbf{s}) - p_u(\mathbf{s})| \geq \frac{1}{4n^c}$ we obtain by an averaging argument that

$$|p_0((\mathbf{a_i}, b_i)_{i=1}^m) - p_u((\mathbf{a_i}, b_i)_{i=1}^m)| \geq \frac{1}{4n^c}$$

with probability at least $\frac{1}{4n^c}$ on the choice of $(\mathbf{a_i}, b_i)_{i=1}^m$ from $A_{\mathbf{s},\chi}$. Hence, with probability at least $\frac{1}{4n^c}$, Z chooses such a $(\mathbf{a_i}, b_i)_{i=1}^m$ and since our estimates are accurate to within $\frac{1}{64n^c}$, the difference between them is more than $\frac{1}{16n^c}$ and Z accepts. ■

6 Acknowledgments

I would like to thank Michael Langberg, Miklos Santha, and Madhu Sudan for useful comments.

References

[1] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. In *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 362–371, 2004.

[2] M. Ajtai. Generating hard instances of lattice problems. In *ECCCTR: Electronic Colloquium on Computational Complexity, technical reports*, 1996.

[3] 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.

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

[5] 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.

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

[7] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

[8] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in cryptology—CRYPTO '93 (Santa Barbara, CA, 1993)*, volume 773 of *Lecture Notes in Comput. Sci.*, pages 278–291. Springer, Berlin, 1994.

[9] 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.

[10] J.-Y. Cai and A. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *Proc. 38th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 468–477, 1997.

[11] W. Ebeling. *Lattices and codes*. Advanced Lectures in Mathematics. Friedr. Vieweg & Sohn, Braunschweig, revised edition, 2002. A course partially based on lectures by F. Hirzebruch.

[12] 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.

[13] R. Kumar and D. Sivakumar. On polynomial approximation to the shortest lattice vector length. In *Proc. 12th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 126–127, 2001.

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

[15] D. Micciancio. Improved cryptographic hash functions with worst-case/average-case connection. In *Proc. 34th Annual ACM Symp. on Theory of Computing (STOC)*, pages 609–618, 2002.

[16] D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM Journal on Computing*, 2004. Accepted for publication. Available from author's web page at URL http://www.cse.ucsd.edu/users/daniele.

[17] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, Mar. 2002.

[18] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, 2004.

[19] O. Regev. New lattice based cryptographic constructions. In *Proc. 35th Annual ACM Symp. on Theory of Computing (STOC)*, pages 407–416, 2003.

A Some Technical Claims

Claim A.1 *For any $0 \le \varepsilon \le 1/2$, if τ is some probability density function and f is some non-negative function such that $\Delta(\tau, f) \le \varepsilon$ then $\Delta(\tau, \phi) \le 3\varepsilon$ where $\phi = f / \int f$ is the probability density function obtained from f by normalizing.*

Proof: By the triangle inequality,

$$1-\varepsilon \le \int \tau - \int |f-\tau| \le \int f \le \int \tau + \int |f-\tau| \le 1+\varepsilon.$$

This implies that,

$$\Delta(f,\phi) = \int |f-\phi| = |1 - 1/\textstyle\int f| \le 2\varepsilon$$

and by another triangle inequality, we obtain

$$\Delta(\tau,\phi) \le \Delta(\tau,f) + \Delta(f,\phi) \le 3\varepsilon.$$

∎

Claim A.2 *For any $0<\varepsilon<1/10$, any integer $n \ge 1$ and any $0<s<t$, the statistical distance between an n-dimensional Gaussian scaled by s and one scaled by t is*

$$\Delta(\nu_s,\nu_t) \le 6n\Big(\frac{t}{s}-1\Big).$$

Proof: By scaling appropriately, we can assume without loss of generality that the two standard deviations are 1 and $1+\varepsilon$ for $\varepsilon := t/s-1$. Moreover, it is enough to prove the claim for $n=1$; the general claim follows since the Gaussian distribution is composed of n independent one-dimensional distributions.

We have

$$\begin{aligned}\Delta(\rho,\rho_{1+\varepsilon}) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \left|\exp\left(-\frac{x^2}{2}\right) - \exp\left(-\frac{x^2}{2(1+\varepsilon)^2}\right)\right| dx \\ &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \exp\left(-\frac{x^2}{2(1+\varepsilon)^2}\right)\left|\exp\left(-\Big(1-\frac{1}{(1+\varepsilon)^2}\Big)\cdot\frac{x^2}{2}\right)-1\right| dx.\end{aligned}$$

Now, using $1-y^2 \le \exp\left(-y^2\right) \le 1$,

$$\left|\exp\left(-\Big(1-\frac{1}{(1+\varepsilon)^2}\Big)\cdot\frac{x^2}{2}\right)-1\right| \le \Big(1-\frac{1}{(1+\varepsilon)^2}\Big)\cdot\frac{x^2}{2} \le \varepsilon x^2.$$

Hence,

$$\Delta(\rho,\rho_{1+\varepsilon}) = \varepsilon\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \exp\left(-\frac{x^2}{2(1+\varepsilon)^2}\right)x^2 dx = \varepsilon(1+\varepsilon)^3 < 2\varepsilon.$$

Since $\nu=\rho$ and $\nu_{1+\varepsilon} = \rho_{1+\varepsilon}/(1+\varepsilon)$, Claim A.1 implies that $\Delta(\nu,\nu_{1+\varepsilon}) \le 6\varepsilon$, as required. ∎

Claim A.3 *Let $\xi : \mathbb{R}^n \to \mathbb{R}^n$ be any function satisfying that for all $\mathbf{x} \in \mathbb{R}^n$, $\|\xi(\mathbf{x})-\mathbf{x}\| \le l$ for some $0<l<s/(4n^2)$. Then, for any $s>0$, the statistical distance between the density functions ν_s and $\nu_s(\xi(\cdot))/Z$, where Z is a normalizing factor, is at most $100n\sqrt{l/s}$.*

Proof: Without loss of generality, we can assume $s=1$. We bound $\rho(\xi(\cdot))$ and ρ from above and below by two functions f_1 and f_2. We obtain the desired result by showing that f_1 is close to f_2.

Let $\varepsilon := \sqrt{l}$ and note that $\varepsilon < 1/(2n)$. Using our assumption on ξ, $(\|y\|+l)^2 \le (1+\varepsilon)\|y\|^2 + (1+1/\varepsilon)l^2$, and $e^{-z} \ge 1-z$ for $z \ge 0$,

$$\begin{aligned}\rho(\xi(\mathbf{y})) &\ge \exp\left(-\pi(\|\mathbf{y}\|+l)^2\right) \\ &\ge (1-\pi(1+1/\varepsilon)l^2)\cdot\exp\left(-\pi((1+\varepsilon)\|\mathbf{y}\|)^2\right).\end{aligned}$$

We have $\pi(1+1/\varepsilon)l^2 \le 2\pi l^2/\varepsilon = 2\pi\varepsilon^3$ and $(1+\varepsilon)^{-n} \ge (1-\varepsilon)^n \ge 1 - n\varepsilon$. Hence,

$$\begin{aligned}\rho(\xi(\mathbf{y})) &\ge (1-2\pi\varepsilon^3)(1-n\varepsilon)\cdot(1+\varepsilon)^n \exp\left(-\pi((1+\varepsilon)\|\mathbf{y}\|)^2\right)\\ &\ge (1-2n\varepsilon)\cdot\nu_{1/(1+\varepsilon)}(\mathbf{y}) =: f_1(\mathbf{y})\end{aligned}$$

Next, we show that

$$\begin{aligned}\rho(\xi(\mathbf{y})) &\le \exp\left(-\pi(\max\{\|\mathbf{y}\|-l,0\})^2\right)\\ &\le (1+2\pi(l/\varepsilon)^2)\cdot\exp\left(-\pi((1-\varepsilon)\|\mathbf{y}\|)^2\right).\end{aligned}$$

The first inequality follows from the assumption on ξ. For the second inequality, notice that if $\|\mathbf{y}\| \ge l/\varepsilon$ then we have $(1-\varepsilon)\|\mathbf{y}\| \le \|\mathbf{y}\| - l$ and hence

$$\exp\left(-\pi((1-\varepsilon)\|\mathbf{y}\|)^2\right) \ge \exp\left(-\pi(\max\{\|\mathbf{y}\|-l,0\})^2\right).$$

If $\|\mathbf{y}\| < l/\varepsilon$, then

$$\exp\left(-\pi((1-\varepsilon)\|\mathbf{y}\|)^2\right) \ge \exp\left(-\pi\|\mathbf{y}\|^2\right) \ge 1-\pi(l/\varepsilon)^2 \ge 1/(1+2\pi(l/\varepsilon)^2).$$

We have $2\pi(l/\varepsilon)^2 = 2\pi\varepsilon^2$ and $(1-\varepsilon)^{-n} \le 1/(1-n\varepsilon) \le 1+2\varepsilon n$ since $n\varepsilon \le 1/2$. Hence,

$$\begin{aligned}\rho(\xi(\mathbf{y})) &\le (1+2\pi\varepsilon^2)(1+2\varepsilon n)\cdot(1-\varepsilon)^n \exp\left(-\pi((1-\varepsilon)\|\mathbf{y}\|)^2\right)\\ &\le (1+3\varepsilon n)\cdot\nu_{1/(1-\varepsilon)}(\mathbf{y}) =: f_2(\mathbf{y})\end{aligned}$$

Now, it is easy to see that for any point $\mathbf{y}$, $f_1(\mathbf{y}) \le \rho(\mathbf{y}) \le f_2(\mathbf{y})$. Hence, $|\rho(\mathbf{y}) - \rho(\xi(\mathbf{y}))| \le |f_2(\mathbf{y}) - f_1(\mathbf{y})|$ and

$$\begin{aligned}\Delta(\rho,\rho(\xi(\cdot))) &\le \Delta(f_1,f_2)\\ &\le \Delta(f_1,\nu_{1/(1+\varepsilon)}) + \Delta(\nu_{1/(1+\varepsilon)},\nu_{1/(1-\varepsilon)}) + \Delta(\nu_{1/(1-\varepsilon)},f_2)\\ &\le 2\varepsilon n + 6n\Big(\frac{1+\varepsilon}{1-\varepsilon} - 1\Big) + 3\varepsilon n\\ &\le 2\varepsilon n + 18\varepsilon n + 3\varepsilon n = 23\varepsilon n\end{aligned}$$

where we used Claim A.2. Using Claim A.1,

$$\Delta(\rho,\rho(\xi(\cdot))/Z) \le 69\varepsilon n < 100\varepsilon n.$$

■