

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

Oded Regev *

September 8, 2007

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.

We also present a (classical) public-key cryptosystem whose security is based on the hardness of the learning problem. By the main result, its security is also 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 based only 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 a factor of $\tilde{O}(n)$ (in previous cryptosystems these values are $\tilde{O}(n^4)$ and $\tilde{O}(n^2)$, respectively). In fact, under the assumption that all parties share a random bit string of length $\tilde{O}(n^2)$, the size of the public key can be reduced to $\tilde{O}(n)$.

1 Introduction

Main theorem. For an integer $n \geq 1$ and a real number $\varepsilon \geq 0$, consider the ‘learning from parity with error’ problem, defined as follows: the goal is to find an unknown $\mathbf{s} \in \mathbb{Z}_2^n$ 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 , $\langle \mathbf{s}, \mathbf{a}_i \rangle = \sum_j s_j (a_i)_j$ is the inner product modulo 2 of \mathbf{s} and \mathbf{a}_i , and each equation is correct independently with probability $1 - \varepsilon$. More precisely, the input to the problem consists of pairs (\mathbf{a}_i, b_i) where each \mathbf{a}_i is chosen independently and

*Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Supported by an Alon Fellowship, by the Binational Science Foundation, by the Israel Science Foundation, by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, and by the Army Research Office grant DAAD19-03-1-0082.

uniformly from \mathbb{Z}_2^n and each b_i is independently chosen to be equal to $\langle \mathbf{s}, \mathbf{a}_i \rangle$ with probability $1 - \varepsilon$. The goal is to find \mathbf{s} . Notice that the case $\varepsilon = 0$ can be solved efficiently by, say, Gaussian elimination. This requires $O(n)$ equations and $\text{poly}(n)$ 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 that we are interested in recovering only the first 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 $(1, 0, \dots, 0)$. 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 [10] 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, $(1, 0, \dots, 0)$. 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 [19, 5].

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 $p = p(n) \leq \text{poly}(n)$ 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 $\chi : \mathbb{Z}_p \rightarrow \mathbb{R}^+$ 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 $\text{LWE}_{p,\chi}$ (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 $\chi(0) > 1/p + 1/\text{poly}(n)$), the maximum likelihood algorithm described above solves $\text{LWE}_{p,\chi}$ for $p \leq \text{poly}(n)$ using $\text{poly}(n)$ equations and $2^{O(n \log n)}$ time. Under a similar assumption, an algorithm resembling the one by Blum et al. [10] 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 $\text{LWE}_{p,\chi}$ implies a quantum solution to worst-case lattice problems.

Theorem 1.1 (Informal) *Let n, p be integers and $\alpha \in (0, 1)$ be such that $\alpha p > 2\sqrt{n}$. If there exists an efficient algorithm that solves $\text{LWE}_{p,\Psi_{\alpha}}$ then there exists an efficient 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 $\alpha = 1/(\sqrt{n} \log^2 n)$ (in fact, these are the parameters that we use in our cryptographic application).

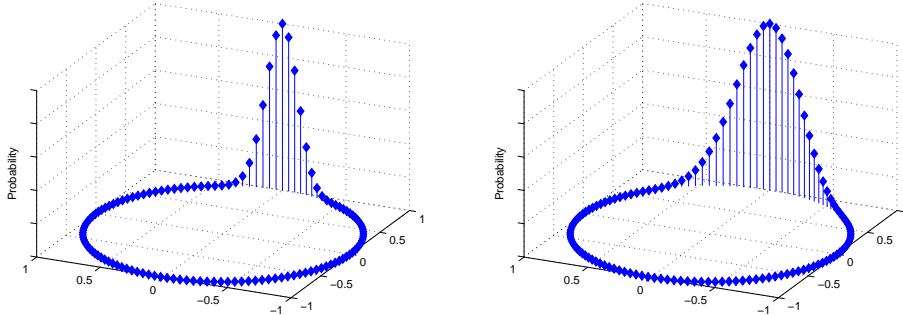


Figure 1: $\bar{\Psi}_\alpha$ for $p = 127$ with $\alpha = 0.05$ (left) and $\alpha = 0.1$ (right). The elements of \mathbb{Z}_p are arranged on a circle.

SVP and SIVP are two of the main computational problems on lattices. The best known polynomial time algorithms for them yield only mildly subexponential approximation factors [20, 32, 5]. 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 even conjecture that there is no *quantum* polynomial time algorithm that approximates SVP (or SIVP) to within any polynomial factor. One can then 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 known quantum algorithms for lattice problems that outperform classical algorithms, even though this is probably one of the most important open questions in the field of quantum computing.¹

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 note that it is possible that our main theorem will one day 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 $m = \text{poly}(n)$ 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.

¹If forced to make a guess, the author would say that the conjecture is true.

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 in Section 4 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 s . The latter formulation is the one we use in our cryptographic applications.

Cryptosystem. In Section 5 we present a public key cryptosystem whose security is based on the LWE problem. The cryptosystem and its security proof are entirely classical. In fact, the cryptosystem itself is quite simple; the reader is encouraged to glimpse at the beginning of Section 5.

By using our main theorem, we obtain that the security of the system is based also on the worst-case 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 $\tilde{O}(n^{1.5})$. Previous cryptosystems, such as the Ajtai-Dwork cryptosystem [4] and the one by Regev [30], were based only 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 completely 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. Moreover, using an idea of Ajtai [3], we can reduce the size of the public key to $\tilde{O}(n)$. This requires all users of the cryptosystem to share some (trusted) random bit string of length $\tilde{O}(n^2)$. This can be achieved by, say, distributing such a bit string as part of the encryption and decryption software.

We mention that learning problems similar to ours were already suggested as possible sources of cryptographic hardness in, e.g., [9, 6], although this was done without establishing any connection to lattice problems. In another related work [3], Ajtai suggested an efficient cryptosystem that has several properties in common with ours, although its security is not based on worst-case lattice problems.

Recent work. After the original publication of our results in [31], several improvements of our cryptosystem were suggested. The first was given by Kawachi et al. [17], who proposed a modification to our cryptosystem that slightly improves the encryption blowup to $O(n)$, essentially getting rid of a log factor. A much more significant improvement is described by Peikert et al. in [27]. By a relatively simple modification to the cryptosystem, they managed to bring the encryption blowup down to only $O(1)$, in addition to several equally significant improvements in running time.

In another recent line of work, the LWE problem was used as the security assumption in the construction of several important cryptographic primitives. This includes the paper by Peikert and Waters [28] who constructed, among other things, CCA-secure cryptosystems. All previous lattice-based cryptosystems (including the one in this paper) are not CCA-secure. In another recent paper, Peikert et al. [27] show how to construct oblivious transfer protocols. In both cases, the security of the construction is based on the LWE problem and hence, by our main theorem, also on the worst-case quantum hardness of lattice problems.

Finally, we mention that Peikert extended the worst case assumptions in our main theorem to ℓ_p norms for all $2 \leq p \leq \infty$ [29].

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. Unlike previous constructions of lattice-based public-key cryptosystems, the proof of our main theorem uses an ‘iterative construction’. 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. So far, such iterative techniques have been used only in the construction of lattice-based one-way functions [2, 11, 22, 24]. Another novel aspect of our main theorem is its crucial use of quantum computation. Our cryptosystem is the first *classical* cryptosystem whose security is based on a *quantum* hardness assumption (see [25] for a somewhat related recent work).

Our proof is based on the Fourier transform of Gaussian measures, a technique that was developed in previous papers [30, 24, 1]. More specifically, we use a parameter known as the smoothing parameter, as introduced in [24]. We also use the discrete Gaussian distribution and approximations to its Fourier transform, ideas that were developed in [1].

Open questions. One 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 over the last few years, 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 [13], Feige presents a conjecture 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 [6] 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 per-

haps using our main theorem, some inapproximability results can be derived based on the quantum-hardness of lattices. Some interesting progress in this direction was recently made by Klivans and Sherstov [18].

1.1 Overview

In this subsection, we give a brief informal overview of the proof of our main theorem in Section 3. We do not include a description of the reductions in Section 4 and the cryptosystem in 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 $\exp(-\pi\|\mathbf{x}/r\|^2)$ (see Eq. (4) and Figure 2). Its support is the lattice L and for a ‘not too small’ r (which is the case in all our applications), vectors chosen from it have length roughly $r\sqrt{n}$ with high probability.

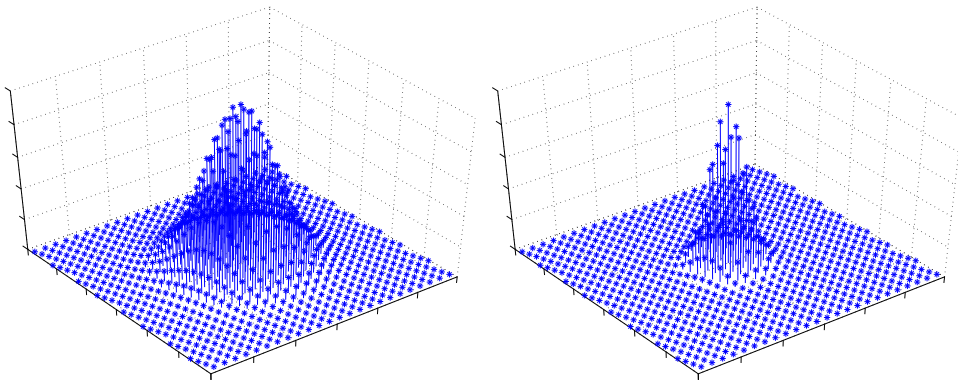


Figure 2: $D_{L,2}$ (left) and $D_{L,1}$ (right) for a two-dimensional lattice L . The z -axis represents probability.

The proof of the main theorem shows a quantum algorithm that solves the worst-case lattice problems SVP and SIVP using an oracle for the $\text{LWE}_{p,\bar{\Psi}_\alpha}$ problem. As we prove in Subsection 3.3, in order to solve these worst-case lattice problems, it suffices to be able to generate samples from $D_{L,r}$ for any given lattice L and a ‘small enough’ r . Roughly speaking, the reason for this is that samples from $D_{L,r}$ are short lattice vectors (of length roughly $\sqrt{n}r$) and therefore allow us to solve SVP and SIVP. In the rest of this subsection we describe our algorithm for sampling from $D_{L,r}$.

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 (such samples can be computed efficiently by a simple procedure described in Lemma 3.2). 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 . When r is small enough, the algorithm simply outputs the samples found.

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 $D_{L,r'}$. We do this in two steps (see Figure 3). In the first step, we use these samples to construct an algorithm that solves the closest vector problem on the dual lattice L^* for points that are within distance $\alpha p/r$ of L^* , a problem that we denote by $\text{CVP}_{L^*,\alpha p/r}$ (in fact, we only solve $\text{CVP}_{L^*,\alpha p/(\sqrt{2}r)}$ but for simplicity we ignore the factor $\sqrt{2}$ here). 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.

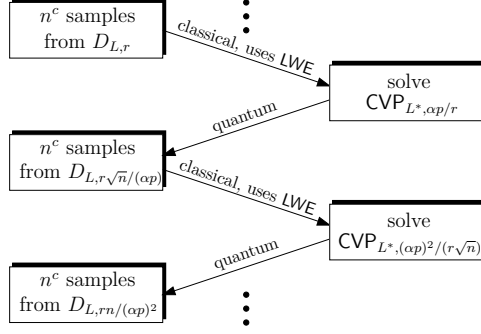


Figure 3: Two iterations of the algorithm

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 $f : \mathbb{R}^n \rightarrow \mathbb{C}$, defined as

$$f(\mathbf{x}) = \sum_{\mathbf{y} \in L} D(\mathbf{y}) \exp(2\pi i \langle \mathbf{x}, \mathbf{y} \rangle) = \mathbb{E}_{\mathbf{y} \sim D} [\exp(2\pi i \langle \mathbf{x}, \mathbf{y} \rangle)]$$

where in the second equality we simply rewrite the sum as an expectation. By definition, f 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 of f to within $\pm 1/\text{poly}(n)$. To see this, note that by the Chernoff-Hoeffding bound, if $\mathbf{y}_1, \dots, \mathbf{y}_N$ are $N = \text{poly}(n)$ independent samples from D , then

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

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

If we take our samples from the distribution $D_{L,r}$, we obtain a good approximation of its Fourier transform, which we denote by $f_{1/r}$. We omit the exact expression for $f_{1/r}$. It suffices to know that since r is not too small, $f_{1/r}$ is closely approximately by

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

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 the range of r that we are interested in, the distance between any two vectors in L^* is much larger than $1/r$ and hence these Gaussians are well-separated.

Although not needed in this paper, let us briefly outline how one can solve $\text{CVP}_{L^*, 1/r}$ using samples from $D_{L,r}$ (recall that our final goal is to solve the more difficult $\text{CVP}_{L^*, \alpha p/r}$). Assume that 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} . Actually, the procedure as described here does not quite work: due to the error in our approximation

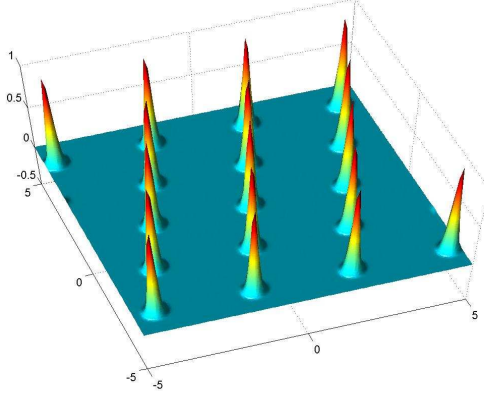


Figure 4: $f_{1/r}$ for a two-dimensional lattice

of $f_{1/r}$, we cannot find the closest lattice point exactly. It is possible to overcome this difficulty; see [21] for the details. Notice that this procedure, unlike our solution to $\text{CVP}_{L^*, \alpha p/r}$ below, 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 $\text{CVP}_{L^*, \alpha p/r}$ 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 $\{pL + L\mathbf{a} \mid \mathbf{a} \in \mathbb{Z}_p^n\}$ 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 not too small, and so each of the translated lattices $pL + L\mathbf{a}$ 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 $\text{CVP}_{L^*, p/r}$ which is even better than what we need (since $\alpha < 1$). However, the probability that **SAMPLEPAIR** produces a pair with $\mathbf{a} = \mathbf{0}$ 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

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

Notice that the absolute value of (2) is $f_{p/r}(\mathbf{x})$. If we 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 of (2). By taking its absolute value, we would obtain a good approximation of $f_{p/r}$ from which we can proceed as before to solve $\text{CVP}_{L^*,p/r}$. 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.

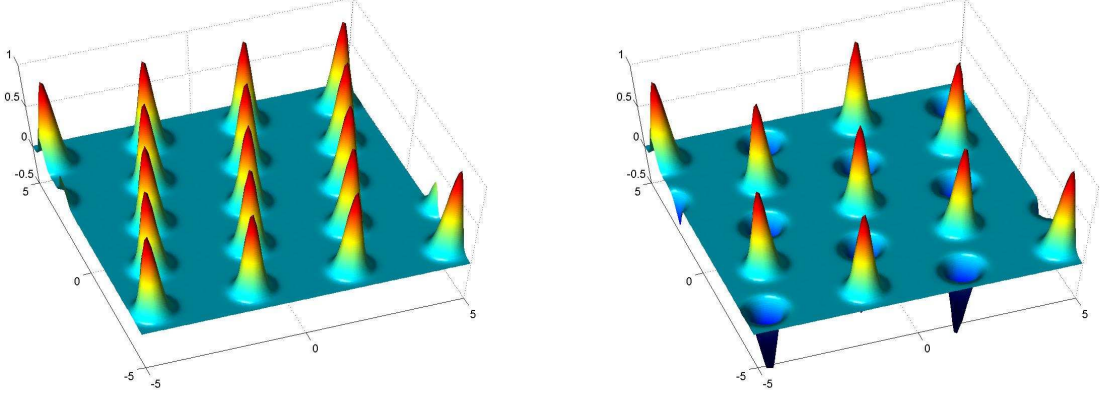


Figure 5: The Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ with $n = 2, p = 2, \mathbf{a} = (0, 0)$ (left), $\mathbf{a} = (1, 1)$ (right).

To explain our approach, let us step back and examine the expression $\exp(2\pi i \langle \mathbf{x}, \mathbf{y} \rangle)$. 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 $\exp(2\pi i \langle \mathbf{a}, \tau(\mathbf{x}) \rangle / p)$. 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 seen directly).

The situation becomes more interesting once we take a vector \mathbf{x} whose distance to L^* is $\beta p / r$ for some $0 \leq \beta \leq \alpha$. 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, $\exp(-\pi\beta^2)$). 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 $0 \leq \beta \leq \alpha$. 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 $\lfloor p\langle \mathbf{x}, \mathbf{y} \rangle \rfloor \bmod p$ is distributed like $\tilde{\Psi}_\beta$ around $\langle \mathbf{a}, \tau(\mathbf{x}) \rangle \bmod p$. This can be written as an ‘equation with error’

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

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

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

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) $\beta \leq \alpha$.

Step 2: In this step, we describe a quantum algorithm that, using a $\text{CVP}_{L^*, \alpha p/r}$ oracle, generates one sample from $D_{L, r\sqrt{n}/(\alpha p)}$ (one can generate more samples by simply repeating the algorithm). Equivalently, we show how to produce a sample from $D_{L, r}$ given a $\text{CVP}_{L^*, \sqrt{n}/r}$ 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

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

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 present discussion, 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 $1/r$,

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

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

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

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

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

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

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

However, ‘erasing’ a register is in general not a reversible operation. In order for it to be reversible, we need to be able to compute \mathbf{x} from the remaining register $\mathbf{x} + \mathbf{z}$. This is precisely why we need the $\text{CVP}_{L^*, \sqrt{n}/r}$ oracle. It can be shown that almost all the mass of $\exp(-\pi\|r\mathbf{z}\|^2)$ is on \mathbf{z} such that $\|\mathbf{z}\| \leq \sqrt{n}/r$. 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,

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

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. 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 $\lfloor x \rfloor$ 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 $\{0, 1, \dots, p-1\}$ 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 $\lim_{n \rightarrow \infty} n^c f(n) = 0$ 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 have an exponentially small error probability, and we sometimes do not state this explicitly.

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 can be given for discrete random variables. The statistical distance satisfies the triangle inequality, i.e., for any ϕ_1, ϕ_2, ϕ_3 ,

$$\Delta(\phi_1, \phi_3) \leq \Delta(\phi_1, \phi_2) + \Delta(\phi_2, \phi_3).$$

Another important fact which we often 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., [23]. 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 $\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right)$ 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 a sample from the Gaussian distribution ν_s can be obtained by taking n independent samples from the 1-dimensional Gaussian distribution. Hence, sampling from ν_s to within arbitrarily good accuracy can be performed efficiently by using standard techniques. For simplicity, in this paper we assume that we can sample from ν_s exactly.² When s is not specified, we assume that it is 1. Functions are extended to sets in the usual way; i.e., $\rho_s(A) = \sum_{\mathbf{x} \in A} \rho_s(\mathbf{x})$ 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 . The following simple claim bounds the amount by which $\rho_s(\mathbf{x})$ can shrink by a small change in \mathbf{x} .

Claim 2.1 For all $s, t, l > 0$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ with $\|\mathbf{x}\| \leq t$ and $\|\mathbf{x} - \mathbf{y}\| \leq l$,

$$\rho_s(\mathbf{y}) \geq (1 - \pi(2lt + l^2)/s^2)\rho_s(\mathbf{x}).$$

Proof: Using the inequality $e^{-z} \geq 1 - z$,

$$\rho_s(\mathbf{y}) = e^{-\pi\|\mathbf{y}/s\|^2} \geq e^{-\pi(\|\mathbf{x}\|/s + l/s)^2} = e^{-\pi(2l\|\mathbf{x}\|/s^2 + (l/s)^2)} \rho_s(\mathbf{x}) \geq (1 - \pi(2lt + l^2)/s^2)\rho_s(\mathbf{x}).$$

■

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

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

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),

$$\forall r \in [0, 1), \Psi_\beta(r) := \sum_{k=-\infty}^{\infty} \frac{1}{\beta} \cdot \exp\left(-\pi\left(\frac{r-k}{\beta}\right)^2\right).$$

²In 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.

Clearly, one can efficiently sample from Ψ_β . The following technical claim shows that a small change in the parameter β does not change the distribution Ψ_β by much.

Claim 2.2 For any $0 < \alpha < \beta \leq 2\alpha$,

$$\Delta(\Psi_\alpha, \Psi_\beta) \leq 9\left(\frac{\beta}{\alpha} - 1\right).$$

Proof: We will show that the statistical distance between a normal variable with standard deviation $\alpha/\sqrt{2\pi}$ and one with standard deviation $\beta/\sqrt{2\pi}$ is at most $9(\frac{\beta}{\alpha} - 1)$. This implies the claim since applying a function (modulo 1 in this case) cannot increase the statistical distance. By scaling, we can assume without loss of generality that $\alpha = 1$ and $\beta = 1 + \varepsilon$ for some $0 < \varepsilon \leq 1$. Then the statistical distance that we wish to bound is given by

$$\begin{aligned} \int_{\mathbb{R}} \left| e^{-\pi x^2} - \frac{1}{1+\varepsilon} e^{-\pi x^2/(1+\varepsilon)^2} \right| dx &\leq \int_{\mathbb{R}} |e^{-\pi x^2} - e^{-\pi x^2/(1+\varepsilon)^2}| dx + \int_{\mathbb{R}} \left| \left(1 - \frac{1}{1+\varepsilon}\right) e^{-\pi x^2/(1+\varepsilon)^2} \right| dx \\ &= \int_{\mathbb{R}} |e^{-\pi x^2} - e^{-\pi x^2/(1+\varepsilon)^2}| dx + \varepsilon \\ &= \int_{\mathbb{R}} |e^{-\pi(1-1/(1+\varepsilon)^2)x^2} - 1| e^{-\pi x^2/(1+\varepsilon)^2} dx + \varepsilon. \end{aligned}$$

Now, since $1 - z \leq e^{-z} \leq 1$ for all $z \geq 0$,

$$|e^{-\pi(1-1/(1+\varepsilon)^2)x^2} - 1| \leq \pi(1 - 1/(1+\varepsilon)^2)x^2 \leq 2\pi\varepsilon x^2.$$

Hence we can bound the statistical distance above by

$$\varepsilon + 2\pi\varepsilon \int_{\mathbb{R}} x^2 e^{-\pi x^2/(1+\varepsilon)^2} dx = \varepsilon + \varepsilon(1+\varepsilon)^3 \leq 9\varepsilon.$$

■

For an arbitrary probability distribution with density function $\phi : \mathbb{T} \rightarrow \mathbb{R}^+$ and some integer $p \geq 1$ we define its discretization $\bar{\phi} : \mathbb{Z}_p \rightarrow \mathbb{R}^+$ 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 $\chi : \mathbb{Z}_p \rightarrow \mathbb{R}^+$ 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 $\text{LWE}_{p,\chi}$ if, for any $\mathbf{s} \in \mathbb{Z}_p^n$, given samples from $A_{\mathbf{s},\chi}$ 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 $\text{LWE}_{p,\phi}$ if, for any $\mathbf{s} \in \mathbb{Z}_p^n$, given samples from $A_{\mathbf{s},\phi}$ 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 an arbitrary integer.

Lattices. We briefly review some basic definitions; for a good introduction to lattices, see [23]. 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 $(\mathbf{v}_1, \dots, \mathbf{v}_n)$ of a lattice L , the *fundamental parallelepiped* generated by this basis is defined as

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

When the choice of basis is clear, we write $\mathcal{P}(L)$ instead of $\mathcal{P}(\mathbf{v}_1, \dots, \mathbf{v}_n)$. For a point $\mathbf{x} \in \mathbb{R}^n$ we define $\mathbf{x} \bmod \mathcal{P}(L)$ as the unique point $\mathbf{y} \in \mathcal{P}(L)$ such that $\mathbf{y} - \mathbf{x} \in L$. 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 lattice given by the set of all vectors $\mathbf{y} \in \mathbb{R}^n$ such that $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$ for all vectors $\mathbf{x} \in L$. Similarly, given a basis $(\mathbf{v}_1, \dots, \mathbf{v}_n)$ of a lattice, we define the dual basis as the set of vectors $(\mathbf{v}_1^*, \dots, \mathbf{v}_n^*)$ 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 $\mathbf{v}_1, \dots, \mathbf{v}_n$. 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 integer coefficient vector of \mathbf{v} .

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 \mathbf{v} whose distance from L is less than $\lambda_1(L)/2$ we define $\kappa_L(\mathbf{v})$ as the (unique) closest point to \mathbf{v} in L . The following useful fact, due to Banaszczyk, is known as a ‘transference theorem’. We remark that the lower bound is easy to prove.

Lemma 2.3 ([8], Theorem 2.1) *For any lattice n -dimensional L , $1 \leq \lambda_1(L) \cdot \lambda_n(L^*) \leq n$.*

Two other useful facts by Banaszczyk are the following. The first bounds the amount by which the Gaussian measure of a lattice changes by scaling; the second shows that for any lattice L , the discrete Gaussian measure $D_{L,r}$ is concentrated on points of norm at most \sqrt{nr} (the analogous statement for the continuous Gaussian ν_r is easy to establish).

Lemma 2.4 ([8], Lemma 1.4(i)) *For any lattice L and $a \geq 1$, $\rho_a(L) \leq a^n \rho(L)$.*

Lemma 2.5 ([8], Lemma 1.5(i)) *Let B_n denote the Euclidean unit ball. Then, for any lattice L and any $r > 0$, $\rho_r(L \setminus \sqrt{nr}B_n) < 2^{-2n} \cdot \rho_r(L)$.*

In this paper we consider the following lattice problems. The first two, the shortest vector problem (SVP) and the shortest independent vectors problem (SIVP), are among the most well-known lattice problems and are concerned with λ_1 and λ_n , respectively. In the definitions below, $\gamma = \gamma(n) \geq 1$ is the approximation factor, and the input lattice is given in the form of some arbitrary basis. For SVP we define the decision variant, whereas for SIVP we define the search variant.

Definition 2.6 *An instance of GAPSVP_γ is given by an n -dimensional lattice L and a number $d > 0$. In YES instances, $\lambda_1(L) \leq d$ whereas in NO instances $\lambda_1(L) > \gamma(n) \cdot d$.*

Definition 2.7 *An instance of SIVP_γ is given by an n -dimensional lattice L . The goal is to output a set of n linearly independent lattice vectors of length at most $\gamma(n) \cdot \lambda_n(L)$.*

A useful generalization of SIVP is the following somewhat less standard lattice problem, known as the generalized independent vectors problem (GIVP). Here, φ denotes an arbitrary real-valued function on lattices. Choosing $\varphi = \lambda_n$ results in SIVP.

Definition 2.8 *An instance of GIVP_φ is given by an n -dimensional lattice L . The goal is to output a set of n linearly independent lattice vectors of length at most $\gamma(n) \cdot \varphi(L)$.*

Another useful (and even less standard) lattice problem is the following. We call it the discrete Gaussian sampling problem (DGS). As before, φ denotes an arbitrary real-valued function on lattices.

Definition 2.9 *An instance of DGS_φ is given by an n -dimensional lattice L and a number $r > \varphi(L)$. The goal is to output a sample from $D_{L,r}$.*

We also consider a variant of the closest vector problem (which is essentially what is known as the bounded distance decoding problem [21]): For an n -dimensional lattice L , and some $d > 0$, we say that an algorithm solves $\text{CVP}_{L,d}$ 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} . In this paper d will always be smaller than $\lambda_1(L)/2$ and hence the closest vector is unique.

The smoothing parameter. We make heavy use of a lattice parameter known as the *smoothing parameter* [24]. 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 2.10 *For an n -dimensional lattice L and positive real $\varepsilon > 0$, we define the smoothing parameter $\eta_\varepsilon(L)$ to be the smallest s such that $\rho_{1/s}(L^* \setminus \{\mathbf{0}\}) \leq \varepsilon$.*

In other words, $\eta_\varepsilon(L)$ is the smallest s such that a Gaussian measure scaled by $1/s$ on the dual lattice L^* 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 n . Notice that $\rho_{1/s}(L^* \setminus \{\mathbf{0}\})$ is a continuous and strictly decreasing function of s . Moreover, it can be shown that $\lim_{s \rightarrow 0} \rho_{1/s}(L^* \setminus \{\mathbf{0}\}) = \infty$ and $\lim_{s \rightarrow \infty} \rho_{1/s}(L^* \setminus \{\mathbf{0}\}) = 0$. 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 [24]. 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 ’. The formal statement is given by considering distributions on $\mathcal{P}(L)$; see [24] 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 obtains when summing two independent samples from ν_s . Hence, intuitively, the noise ν_s is enough to hide the discrete structure of $D_{L,s}$.

The following two upper bounds on the smoothing parameter appear in [24].

Lemma 2.11 *For any n -dimensional lattice L , $\eta_\varepsilon(L) \leq \sqrt{n}/\lambda_1(L^*)$ where $\varepsilon = 2^{-n}$.*

Lemma 2.12 *For any n -dimensional lattice L and $\varepsilon > 0$,*

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

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

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

Claim 2.13 *For any lattice L and any $\varepsilon > 0$,*

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

In particular, for any $\varepsilon(n) = o(1)$ and any constant $c > 0$, $\eta_{\varepsilon(n)}(L) > c/\lambda_1(L^) \geq c\lambda_n(L)/n$ 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,

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

The first inequality follows by solving for s . The second inequality is by Lemma 2.3. ■

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., [12]. The Fourier transform of a function $h : \mathbb{R}^n \rightarrow \mathbb{C}$ 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

$$\hat{h}(\mathbf{w}) = e^{2\pi i \langle \mathbf{v}, \mathbf{w} \rangle} \hat{g}(\mathbf{w}). \quad (5)$$

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

$$\hat{h}(\mathbf{w}) = \hat{g}(\mathbf{w} - \mathbf{v}). \quad (6)$$

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 $\hat{\rho}_s = s^n \rho_{1/s}$. Finally, we will use the following formulation of the Poisson summation formula.

Lemma 2.14 *For any lattice L and any function $f : \mathbb{R}^n \rightarrow \mathbb{C}$, $f(L) = \det(L^*) \hat{f}(L^*)$.*

3 Main Theorem

Our main theorem is the following. The connection to the standard lattice problems GAPSVP and SIVP will be established in Subsection 3.3 by polynomial time reductions to DGS.

Theorem 3.1 *Let $\varepsilon = \varepsilon(n)$ be some negligible function of n . Also, let $p = p(n)$ be some integer and $\alpha = \alpha(n) \in (0, 1)$ be such that $\alpha p > 2\sqrt{n}$. Assume that we have access to an oracle W that solves $\text{LWE}_{p, \Psi_\alpha}$ given a polynomial number of samples. Then there exists an efficient quantum algorithm for $\text{DGS}_{\sqrt{2n} \cdot \eta_\varepsilon(L)/\alpha}$.*

Proof: The input to our algorithm is an n -dimensional lattice L and a number $r > \sqrt{2n} \cdot \eta_\varepsilon(L)/\alpha$. Our goal is to output a sample from $D_{L,r}$. Let r_i denote $r \cdot (\alpha p/\sqrt{n})^i$. The algorithm starts by producing n^c samples from $D_{L,r_{3n}}$ where c is the constant from Lemma 3.3. By Claim 2.13, $r_{3n} > 2^{3n}r > 2^{2n}\lambda_n(L)$, and hence we can produce these samples efficiently by the procedure described in Lemma 3.2. Next, for $i = 3n, 3n-1, \dots, 1$ we use our n^c samples from D_{L,r_i} to produce n^c samples from $D_{L,r_{i-1}}$. The procedure that does this is the core of the algorithm and is described in Lemma 3.3. Notice that the condition in Lemma 3.3 is satisfied since for all $i \geq 1$, $r_i \geq r_1 = r\alpha p/\sqrt{n} > \sqrt{2}p\eta_\varepsilon(L)$. At the end of the loop, we end up with n^c samples from $D_{L,r_0} = D_{L,r}$ and we complete the algorithm by simply outputting the first of those.

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 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 *There exists an efficient algorithm that, given any n -dimensional lattice L and $r > 2^{2n}\lambda_n(L)$, outputs a sample from a distribution that is within statistical distance $2^{-\Omega(n)}$ of $D_{L,r}$.*

Proof: By using the LLL basis reduction algorithm [20], we obtain a basis for L of length at most $2^n\lambda_n(L)$ and let $\mathcal{P}(L)$ be the parallelepiped generated by this basis. The sampling procedure samples a vector \mathbf{y} from ν_r and then outputs $\mathbf{y} - (\mathbf{y} \bmod \mathcal{P}(L)) \in L$. Notice that $\|\mathbf{y} \bmod \mathcal{P}(L)\| \leq \text{diam}(\mathcal{P}(L)) \leq n2^n\lambda_n(L)$.

Our goal is to show that the resulting distribution is exponentially close to $D_{L,r}$. By Lemma 2.5, all but an exponentially small part of $D_{L,r}$ is concentrated on points of norm at most $\sqrt{n}r$. So consider any $\mathbf{x} \in L$ with $\|\mathbf{x}\| \leq \sqrt{n}r$. By definition, the probability given to it by $D_{L,r}$ is $\rho_r(\mathbf{x})/\rho_r(L)$. By Lemma 2.14, the denominator is $\rho_r(L) = \det(L^*) \cdot r^n \rho_{1/r}(L^*) \geq \det(L^*) \cdot r^n$ and hence the probability is at most $\rho_r(\mathbf{x})/(\det(L^*) \cdot r^n) = \det(L)\nu_r(\mathbf{x})$. On the other hand, by Claim 2.1, the probability given to $\mathbf{x} \in L$ by our procedure is

$$\int_{\mathbf{x} + \mathcal{P}(L)} \nu_r(\mathbf{y}) d\mathbf{y} \geq (1 - 2^{-\Omega(n)}) \det(L)\nu_r(\mathbf{x}).$$

Together, these facts imply that our output distribution is within statistical distance $2^{-\Omega(n)}$ of $D_{L,r}$. ■

3.2 The iterative step

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

Note that the output distribution is taken with respect to the randomness (and quantum measurements) used in the algorithm, and not with respect to the input samples. In particular, this means that from the same set of n^c samples from $D_{L,r}$ we can produce 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 $\text{CVP}_{L^*, \alpha p/(\sqrt{2}r)}$. The second part is shown in Lemma 3.12. There, we describe a quantum algorithm that, given an oracle that solves $\text{CVP}_{L^*, \alpha p/(\sqrt{2}r)}$, outputs a sample from $D_{L,r\sqrt{n}/(\alpha p)}$. This is the only quantum component in this paper. The condition in Lemma 3.12 is satisfied since by Claim 2.13, $\alpha p/(\sqrt{2}r) \leq 1/\eta_\varepsilon(L) \leq \lambda_1(L^*)/2$. ■

3.2.1 From samples to CVP

Our goal in this subsection is to prove the following.

Lemma 3.4 *Let $\varepsilon = \varepsilon(n)$ be a negligible function, $p = p(n) \geq 2$ be an integer, and $\alpha = \alpha(n) \in (0, 1)$ be a real number. Assume that we have access to an oracle W that solves $\text{LWE}_{p, \Psi_\alpha}$ given a polynomial number of samples. Then, there exists an efficient algorithm that, given any n -dimensional lattice L , a number $r > \sqrt{2p}\eta_\varepsilon(L)$, and n^c samples from $D_{L,r}$ for some constant $c > 0$, solves $\text{CVP}_{L^*, \alpha p/(\sqrt{2}r)}$.*

For an n -dimensional lattice L , some $0 < d < \lambda_1(L)/2$, and an integer $p \geq 2$, we say that an algorithm solves $\text{CVP}_{L,d}^{(p)}$ if, given any point $\mathbf{x} \in \mathbb{R}^n$ within distance d of L , it outputs $L^{-1}\kappa_L(\mathbf{x}) \bmod p \in \mathbb{Z}_p^n$, the coefficient vector of the closest vector to \mathbf{x} reduced modulo p . We start with the following lemma, which shows a reduction from $\text{CVP}_{L,d}$ to $\text{CVP}_{L,d}^{(p)}$.

Lemma 3.5 *There exists an efficient algorithm that given a lattice L , a number $d < \lambda_1(L)/2$ and an integer $p \geq 2$, solves $\text{CVP}_{L,d}$ given access to an oracle for $\text{CVP}_{L,d}^{(p)}$.*

Proof: Our input is a point \mathbf{x} within distance d of L . We define a sequence of points $\mathbf{x}_1 = \mathbf{x}, \mathbf{x}_2, \mathbf{x}_3, \dots$ 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 [7]. 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 $\mathbf{a}_{n-1}, \mathbf{a}_{n-2}, \dots, \mathbf{a}_1$. This completes the algorithm since $L\mathbf{a}_1$ is the closest point to $\mathbf{x}_1 = \mathbf{x}$. ■

As we noted in the proof of Lemma 3.3, for our choice of r , $\alpha p/(\sqrt{2}r) \leq \lambda_1(L^*)/2$. Hence, in order to prove Lemma 3.4, it suffices to present an efficient algorithm for $\text{CVP}_{L^*, \alpha p/(\sqrt{2}r)}^{(p)}$. We do this by combining two lemmas. The first, Lemma 3.7, shows an algorithm W' that, given samples from $A_{\mathbf{s}, \Psi_\beta}$ for some (unknown) $\beta \leq \alpha$, outputs \mathbf{s} with probability exponentially close to 1 by using W as an oracle. Its proof is based on Lemma 3.6. The second, Lemma 3.8, is the main lemma of this subsection, and shows how to use W' and the given samples from $D_{L,r}$ in order to solve $\text{CVP}_{L^*, \alpha p/(\sqrt{2}r)}^{(p)}$.

Lemma 3.6 *Let $p = p(n) \geq 1$ be some integer. There exists an efficient algorithm that, given \mathbf{s}' and samples from $A_{\mathbf{s}, \Psi_\alpha}$ for some (unknown) $\mathbf{s} \in \mathbb{Z}_p^n$ and $\alpha < 1$, outputs whether $\mathbf{s} = \mathbf{s}'$ and is correct with probability exponentially close to 1.*

We remark that the lemma holds also for all $\alpha \leq O(\sqrt{\log n})$ with essentially the same proof.

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 $x - \langle \mathbf{a}, \mathbf{s}' \rangle / p \in \mathbb{T}$. The algorithm takes n samples y_1, \dots, y_n from ξ . It then computes $z := \frac{1}{n} \sum_{i=1}^n \cos(2\pi y_i)$. If $z > 0.02$, 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 $k \geq 2$. 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 $\gcd(p, s_j - s'_j) < p$. This clearly implies that the distribution of $a_j(s_j - s'_j)/p \bmod 1$ is periodic with period $1/k$ for some $k \geq 2$. 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 expectation³

$$\tilde{z} := \mathbb{E}_{y \sim \xi} [\cos(2\pi y)] = \int_0^1 \cos(2\pi y) \xi(y) dy = \text{Re} \left[\int_0^1 \exp(2\pi i y) \xi(y) dy \right].$$

First, a routine calculation shows that for $\xi = \Psi_\alpha$, $\tilde{z} = \exp(-\pi\alpha^2)$, which is at least 0.04 for $\alpha < 1$. Moreover, if ξ has a period of $1/k$, then

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

which implies that if $k \geq 2$ then $\tilde{z} = 0$. We complete the proof by noting that by the Chernoff bound, $|z - \tilde{z}| \leq 0.01$ with probability exponentially close to 1. \blacksquare

Lemma 3.7 *Let $p = p(n) \geq 2$ be some integer and $\alpha = \alpha(n) \in (0, 1)$. Assume that we have access to an oracle W that solves $\text{LWE}_{p, \Psi_\alpha}$ by using a polynomial number of samples. Then, there exists an efficient algorithm W' that, given samples from $A_{\mathbf{s}, \Psi_\beta}$ for some (unknown) $\beta \leq \alpha$, outputs \mathbf{s} with probability exponentially close to 1.*

³We remark that this expectation is essentially the Fourier series of ξ at point 1 and that the following arguments can be explained in terms of properties of the Fourier series.

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 $n^{-2c}\alpha^2$ between 0 and α^2 . 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 element of each sample a noise sampled independently from $\Psi_{\sqrt{\gamma}}$. This creates n^c samples taken from the distribution $A_{\mathbf{s}, \Psi_{\sqrt{\beta^2 + \gamma}}}$. It then applies W and obtains some candidate \mathbf{s}' . Using Lemma 3.6, it checks whether $\mathbf{s}' = \mathbf{s}$. If the answer is yes, it outputs \mathbf{s}' ; otherwise, it continues.

We now show that W' finds \mathbf{s} with probability exponentially close to 1. By Lemma 3.6, 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 $\gamma \geq \alpha^2 - \beta^2$. Clearly, $\gamma \leq \alpha^2 - \beta^2 + n^{-2c}\alpha^2$. Define $\alpha' = \sqrt{\beta^2 + \gamma}$. Then,

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

By Claim 2.2, the statistical distance between Ψ_α and $\Psi_{\alpha'}$ is at most $9n^{-2c}$. Hence, the statistical distance between n^c samples from Ψ_α and n^c samples from $\Psi_{\alpha'}$ is at most $9n^{-c}$. Therefore, for our choice of γ , W outputs \mathbf{s} with probability at least $1 - 9n^{-c}/2 - 2^{-\Omega(n)} \geq \frac{1}{2}$. The probability that \mathbf{s} is not found in any of the n calls to W is at most 2^{-n} . ■

Lemma 3.8 *Let $\varepsilon = \varepsilon(n)$ be a negligible function, $p = p(n) \geq 2$ be an integer, and $\alpha = \alpha(n) \in (0, 1)$ be a real number. Assume that we have access to an oracle W that for all $\beta \leq \alpha$, finds \mathbf{s} given a polynomial number of samples from $A_{\mathbf{s}, \Psi_\beta}$ (without knowing β). Then, there exists an efficient algorithm that given an n -dimensional lattice L , a number $r > \sqrt{2}p\eta_\varepsilon(L)$, and a polynomial number of samples from $D_{L,r}$, solves $\text{CVP}_{L^*, \alpha p / (\sqrt{2}r)}^{(p)}$.*

Proof: By scaling the lattice L appropriately, we can assume without loss of generality that $r = 1$. We describe a procedure that given \mathbf{x} within distance $\alpha p / \sqrt{2}$ of L^* , outputs samples from the distribution $A_{\mathbf{s}, \Psi_\beta}$ for some $\beta \leq \alpha$ 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 sample a vector $\mathbf{v} \in L$ from $D_{L,1}$, and let $\mathbf{a} = L^{-1}\mathbf{v} \bmod p$. We then output $(\mathbf{a}, \langle \mathbf{x}, \mathbf{v} \rangle / p + e \bmod 1)$ where $e \in \mathbb{R}$ is chosen according to a normal distribution with standard deviation $\alpha / (2\sqrt{\pi})$. We claim that the distribution given by this procedure is within negligible statistical distance of $A_{\mathbf{s}, \Psi_\beta}$ for some $\beta \leq \alpha$.

We 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

$$\begin{aligned} \sum_{\mathbf{x} \in pL + L\mathbf{a}} \rho(\mathbf{x}) &= \sum_{\mathbf{x} \in pL} \rho_{-L\mathbf{a}}(\mathbf{x}) \\ &= p^{-n} \det(L^*) \sum_{\mathbf{y} \in L^*/p} \widehat{\rho_{-L\mathbf{a}}}(\mathbf{y}) \quad (\text{by Lemma 2.14}) \\ &= p^{-n} \det(L^*) \sum_{\mathbf{y} \in L^*/p} \exp(2\pi i \langle L\mathbf{a}, \mathbf{y} \rangle) \rho(\mathbf{y}) \\ &= p^{-n} \det(L^*) (1 \pm \varepsilon) \quad (\text{since } \rho(L^*/p \setminus \{\mathbf{0}\}) = \rho_p(L^* \setminus \{\mathbf{0}\}) \leq \varepsilon). \end{aligned}$$

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

Next, we consider the distribution of the second element in the procedure's output. Define $\mathbf{x}' = \mathbf{x} - \kappa_{L^*}(\mathbf{x})$ and note that $\|\mathbf{x}'\| \leq \alpha p / \sqrt{2}$. 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,

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

since $L^{-1} = (L^*)^T$. 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 any 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 suffices to show that there exists some $\beta \leq \alpha$ such that conditioned on any fixed value of \mathbf{a} , the statistical distance between the distribution of $\langle \mathbf{x}', \mathbf{v} \rangle / p + e \bmod 1$ and Ψ_β is negligible. In fact, we will show 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 (this implies the previous statement since the statistical distance cannot increase by applying a function, mod 1 in this case).

So let us consider the distribution of $\langle \mathbf{x}', \mathbf{v} \rangle / p + e$. Since we are conditioning on the value \mathbf{a} , the distribution of \mathbf{v} is $D_{pL+L\mathbf{a},1}$. 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 $D_{pL+L\mathbf{a},1}$ and a noise vector \mathbf{h} from $\nu_{p\alpha/(\sqrt{2}\|\mathbf{x}'\|)}$ 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 $\alpha / (2\sqrt{\pi})$.

Lemma 3.9 tells us that the statistical distance between $\mathbf{v} + \mathbf{h}$ and $\nu_{\sqrt{1+(p\alpha/(\sqrt{2}\|\mathbf{x}'\|))^2}}$ is negligible. The condition in the lemma holds since by our assumption,

$$\rho_{\sqrt{(\sqrt{2}\|\mathbf{x}'\|/(p\alpha))^2+1}}(L^*/p \setminus \{\mathbf{0}\}) \leq \rho_{\sqrt{2}}(L^*/p \setminus \{\mathbf{0}\}) = \rho_{\sqrt{2}p}(L^* \setminus \{\mathbf{0}\}) \leq \varepsilon.$$

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

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

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

$$\begin{aligned}
Y(\mathbf{x}) &= \frac{1}{s^n \rho_t(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \rho_t(\mathbf{y}) \rho_s(\mathbf{x} - \mathbf{y}) \\
&= \frac{1}{s^n \rho_t(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \exp(-\pi(\|\mathbf{y}/t\|^2 + \|(\mathbf{x} - \mathbf{y})/s\|^2)) \\
&= \frac{1}{s^n \rho_t(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \exp\left(-\pi\left(\frac{s^2 + t^2}{s^2 \cdot t^2} \cdot \left\|\mathbf{y} - \frac{t^2}{s^2 + t^2} \mathbf{x}\right\|^2 + \frac{1}{s^2 + t^2} \|\mathbf{x}\|^2\right)\right) \\
&= \exp\left(-\frac{\pi}{s^2 + t^2} \|\mathbf{x}\|^2\right) \frac{1}{s^n \rho_t(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \exp\left(-\pi\left(\frac{s^2 + t^2}{s^2 \cdot t^2} \cdot \left\|\mathbf{y} - \frac{t^2}{s^2 + t^2} \mathbf{x}\right\|^2\right)\right) \\
&= \frac{1}{s^n} \rho_r(\mathbf{x}) \cdot \frac{\rho_{st/r, (t/r)^2 \mathbf{x} - \mathbf{u}}(L)}{\rho_{t, -\mathbf{u}}(L)} \\
&= \frac{1}{s^n} \rho_r(\mathbf{x}) \cdot \frac{\widehat{\rho_{st/r, (t/r)^2 \mathbf{x} - \mathbf{u}}}(L^*)}{\widehat{\rho_{t, -\mathbf{u}}}(L^*)} \\
&= \rho_r(\mathbf{x}) / r^n \cdot \frac{(r/st)^n \widehat{\rho_{st/r, (t/r)^2 \mathbf{x} - \mathbf{u}}}(L^*)}{(1/t)^n \widehat{\rho_{t, -\mathbf{u}}}(L^*)} \tag{7}
\end{aligned}$$

where in the second to last equality we used Lemma 2.14. Using Eq. (5),

$$\begin{aligned}
\widehat{\rho_{st/r, (t/r)^2 \mathbf{x} - \mathbf{u}}}(\mathbf{w}) &= \exp(-2\pi i \langle (t/r)^2 \mathbf{x} - \mathbf{u}, \mathbf{w} \rangle) \cdot (st/r)^n \rho_{r/st}(\mathbf{w}), \\
\widehat{\rho_{t, -\mathbf{u}}}(\mathbf{w}) &= \exp(2\pi i \langle \mathbf{u}, \mathbf{w} \rangle) \cdot t^n \rho_{1/t}(\mathbf{w}).
\end{aligned}$$

Hence,

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

where the last inequality follows from $1/t \leq r/st$. Hence, the quotient in (7) is between $(1 - \varepsilon)/(1 + \varepsilon) \geq 1 - 2\varepsilon$ and $(1 + \varepsilon)/(1 - \varepsilon) \leq 1 + 4\varepsilon$. This implies that,

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

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

3.2.2 From CVP to samples

In this subsection we describe a quantum procedure that uses a CVP oracle in order to create samples from the discrete Gaussian distribution. We assume familiarity with some basic notions of quantum computation, such as (pure) states, measurements, and the quantum Fourier transform. See, e.g., [26] for a good introduction. For clarity, we often omit the normalization factors from quantum states.

The following lemma shows that we can efficiently create a ‘discrete quantum Gaussian state’ of radius r as long as r is large enough compared with $\lambda_n(L)$. It can be seen as the quantum analogue of Lemma 3.2. The assumption that $L \subseteq \mathbb{Z}^n$ is essentially without loss of generality since a lattice with rational coordinates can always be rescaled so that $L \subseteq \mathbb{Z}^n$.

Lemma 3.10 *There exists an efficient quantum algorithm that, given an n -dimensional lattice $L \subseteq \mathbb{Z}^n$ and $r > 2^{2n} \lambda_n(L)$, outputs a state that is within ℓ_2 distance $2^{-\Omega(n)}$ of the normalized state corresponding to*

$$\sum_{\mathbf{x} \in L} \sqrt{\rho_r(\mathbf{x})} |\mathbf{x}\rangle = \sum_{\mathbf{x} \in L} \rho_{\sqrt{2r}}(\mathbf{x}) |\mathbf{x}\rangle. \quad (8)$$

Proof: We start by creating the ‘one-dimensional Gaussian state’

$$\sum_{x=-\sqrt{nr}}^{\sqrt{nr}} e^{-\pi(x/(\sqrt{2r}))^2} |x\rangle. \quad (9)$$

This state can be created efficiently using a technique by Grover and Rudolph [15] who show that in order to create such a state, it suffices to be able to compute for any $a, b \in \{-\sqrt{nr}, \dots, \sqrt{nr}\}$ the sum $\sum_{x=a}^b e^{-\pi(x/r)^2}$ to within good precision. This can be done using the same standard techniques used in sampling from the normal distribution.

Repeating the procedure described above n times, creates a system whose state is the n -fold tensor product of the state in Eq. (9), which can be written as

$$\sum_{\mathbf{x} \in \{-\sqrt{nr}, \dots, \sqrt{nr}\}^n} \rho_{\sqrt{2r}}(\mathbf{x}) |\mathbf{x}\rangle.$$

Since $\mathbb{Z}^n \cap \sqrt{nr} B_n \subseteq \{-\sqrt{nr}, \dots, \sqrt{nr}\}^n$, Lemma 2.5 implies that this state is within ℓ_2 distance $2^{-\Omega(n)}$ of

$$\sum_{\mathbf{x} \in \mathbb{Z}^n} \rho_{\sqrt{2r}}(\mathbf{x}) |\mathbf{x}\rangle \quad (10)$$

and hence for our purposes we can assume that we have generated the state in Eq. (10).

Next, using the LLL basis reduction algorithm [20], we obtain a basis for L of length at most $2^n \lambda_n(L)$ and let $\mathcal{P}(L)$ be the parallelepiped generated by this basis. We now compute in a new register $\mathbf{x} \bmod \mathcal{P}(L)$ and measure it. Let $\mathbf{y} \in \mathcal{P}(L)$ denote the result and note that $\|\mathbf{y}\| \leq \text{diam}(\mathcal{P}(L)) \leq n2^n \lambda_n(L)$. The state we obtain after the measurement is

$$\sum_{\mathbf{x} \in L + \mathbf{y}} \rho_{\sqrt{2r}}(\mathbf{x}) |\mathbf{x}\rangle.$$

Finally, we subtract \mathbf{y} from our register, and obtain

$$\sum_{\mathbf{x} \in L} \rho_{\sqrt{2r}}(\mathbf{x} + \mathbf{y}) |\mathbf{x}\rangle.$$

Our goal is to show that this state is within ℓ_2 distance $2^{-\Omega(n)}$ of the one in Eq. (8). First, by Lemma 2.5, all but an exponentially small part of the ℓ_2 norm of the state in Eq. (8) is concentrated on points of norm at most $\sqrt{n} \cdot r$. So consider any $\mathbf{x} \in L$ with $\|\mathbf{x}\| \leq \sqrt{n} \cdot r$. The amplitude squared given to it in Eq. (8) is $\rho_r(\mathbf{x})/\rho_r(L)$. By Lemma 2.14, the denominator is $\rho_r(L) = \det(L^*) \cdot r^n \rho_{1/r}(L^*) \geq \det(L^*) \cdot r^n$ and hence the amplitude squared is at most $\rho_r(\mathbf{x})/(\det(L^*) \cdot r^n) = \det(L) \nu_r(\mathbf{x})$.

On the other hand, the amplitude squared given to \mathbf{x} by our procedure is $\rho_r(\mathbf{x} + \mathbf{y})/\rho_r(L + \mathbf{y})$. By Lemma 2.14, the denominator is

$$\rho_r(L + \mathbf{y}) = \det(L^*) \cdot r^n \sum_{\mathbf{z} \in L^*} e^{2\pi i \langle \mathbf{z}, \mathbf{y} \rangle} \rho_{1/r}(\mathbf{z}) \leq (1 + 2^{-\Omega(n)}) \det(L^*) \cdot r^n.$$

To obtain this inequality, first note that by the easy part of Lemma 2.3, $\lambda_1(L^*) \geq 1/\lambda_n(L) > \sqrt{n}/r$, and then apply Lemma 2.5. Moreover, by Claim 2.1, the numerator is at least $(1 - 2^{-\Omega(n)})\rho_r(\mathbf{x})$. Hence, the amplitude squared given to \mathbf{x} is at least $(1 - 2^{-\Omega(n)}) \det(L)\nu_r(\mathbf{x})$, as required. ■

For a lattice L and a positive integer R , we denote by L/R the lattice obtained by scaling down L by a factor of R . The following technical claim follows from the fact that almost all the mass of ρ is on points of norm at most \sqrt{n} .

Claim 3.11 *Let $R \geq 1$ be an integer and L be an n -dimensional lattice satisfying $\lambda_1(L) > 2\sqrt{n}$. Let $\mathcal{P}(L)$ be some basic parallelepiped of L . Then, the ℓ_2 distance between the normalized quantum states corresponding to*

$$\begin{aligned} |\vartheta_1\rangle &= \sum_{\mathbf{x} \in L/R, \|\mathbf{x}\| < \sqrt{n}} \rho(\mathbf{x}) |\mathbf{x} \bmod \mathcal{P}(L)\rangle, \quad \text{and} \\ |\vartheta_2\rangle &= \sum_{\mathbf{x} \in L/R} \rho(\mathbf{x}) |\mathbf{x} \bmod \mathcal{P}(L)\rangle = \sum_{\mathbf{x} \in L/R \cap \mathcal{P}(L)} \sum_{\mathbf{y} \in L} \rho(\mathbf{x} - \mathbf{y}) |\mathbf{x}\rangle \end{aligned}$$

is $2^{-\Omega(n)}$.

Proof: We think of $|\vartheta_1\rangle$ and $|\vartheta_2\rangle$ as vectors in R^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 distance 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 . Since $\lambda_1(L) > 2\sqrt{n}$, each ‘ket’ in the definition of $|\vartheta_1\rangle$ appears in the sum only once, and so

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

By applying Lemma 2.5 to the lattice $\sqrt{2}L/R$, we obtain that

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

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

$$\begin{aligned} \||\vartheta_1\rangle - |\vartheta_2\rangle\|_2 &\leq \||\vartheta_1\rangle - |\vartheta_2\rangle\|_1 \\ &= \sum_{\mathbf{x} \in L/R, \|\mathbf{x}\| \geq \sqrt{n}} \rho(\mathbf{x}) \\ &\leq 2^{-2n}\rho(L/R) \quad (\text{by Lemma 2.5}) \\ &\leq 2^{-2n}2^{n/2}\rho(\sqrt{2}L/R) \quad (\text{by Lemma 2.4}) \\ &\leq 2^{-n}\rho(\sqrt{2}L/R). \end{aligned}$$

■

We now prove the main lemma of this subsection.

Lemma 3.12 *There exists an efficient quantum algorithm that, given any n -dimensional lattice L , a number $d < \lambda_1(L^*)/2$, and an oracle that solves $\text{CVP}_{L^*,d}$, outputs a sample from $D_{L, \sqrt{n}/(\sqrt{2}d)}$.*

Proof: By scaling, we can assume without loss of generality that $d = \sqrt{n}$. Let $R \geq 2^{3n} \lambda_n(L^*)$ be a large enough integer. We can assume that $\log R$ is polynomial in the input size (since such an R can be computed in polynomial time given the lattice L). Our first step is to create a state exponentially close to

$$\sum_{\mathbf{x} \in L^*/R \cap \mathcal{P}(L^*)} \sum_{\mathbf{y} \in L^*} \rho(\mathbf{x} - \mathbf{y}) |\mathbf{x}\rangle. \quad (11)$$

This is a state on $n \log R$ qubits, a number that is polynomial in the input size. To do so, we first use Lemma 3.10 with $r = 1/\sqrt{2}$ and the lattice L^*/R to create the state

$$\sum_{\mathbf{x} \in L^*/R} \rho(\mathbf{x}) |\mathbf{x}\rangle.$$

By Lemma 2.5, this is exponentially close to

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

Next, we compute $\mathbf{x} \bmod \mathcal{P}(L^*)$ in a new register and obtain

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

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

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

Using Claim 3.11, this state is exponentially close to the required state (11).

In the second step, we apply the quantum Fourier transform. First, using the natural mapping between $L^*/R \cap \mathcal{P}(L^*)$ and \mathbb{Z}_R^n , we can rewrite (11) as

$$\sum_{\mathbf{s} \in \mathbb{Z}_R^n} \sum_{\mathbf{r} \in \mathbb{Z}^n} \rho(L^* \mathbf{s}/R - L^* \mathbf{r}) |\mathbf{s}\rangle.$$

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

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

where the last equality follows from Lemma 2.14 and Eq. (6). Hence, the resulting state can be equivalently written as

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

Notice that $\lambda_1(RL) = R\lambda_1(L) \geq R/\lambda_n(L^*) \geq 2^{3n}$. Hence, we can apply Claim 3.11 to the lattice RL , and obtain that this state is exponentially close to

$$\sum_{\mathbf{x} \in L, \|\mathbf{x}\| < \sqrt{n}} \rho(\mathbf{x} | \mathbf{x} \bmod \mathcal{P}(RL)).$$

We measure this state and obtain $\mathbf{x} \bmod \mathcal{P}(RL)$ for some vector \mathbf{x} with $\|\mathbf{x}\| < \sqrt{n}$. Since $\mathbf{x} \bmod \mathcal{P}(RL)$ is within \sqrt{n} of the lattice RL , and $\lambda_1(RL) \geq 2^{3n}$, we can recover \mathbf{x} by using, say, Babai's nearest plane algorithm [7]. 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 2.5, 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 Standard lattice problems

We now complete the proof of the main theorem by reducing the standard lattice problems GAPSVP and SIVP to DGS. We start with SIVP. The basic idea of the reduction is simple: we call the DGS oracle enough times. We show that with high probability, there are n short linearly independent vectors among the returned vectors. We prove this by using the following lemma, which appeared in the preliminary version of [24]. We include the proof since only a proof sketch was given there.

Lemma 3.13 *Let L be an n -dimensional lattice and let r be such that $r \geq \sqrt{2}\eta_\varepsilon(L)$ where $\varepsilon \leq \frac{1}{10}$. Then for any subspace H of dimension at most $n - 1$ the probability that $\mathbf{x} \notin H$ where \mathbf{x} is chosen from $D_{L,r}$ is at least $\frac{1}{10}$.*

Proof: Assume without loss of generality that the vector $(1, 0, \dots, 0)$ is orthogonal to H . Using Lemma 2.14,

$$\begin{aligned} \mathbb{E}_{\mathbf{x} \sim D_{L,r}} [\exp(-\pi(x_1/r)^2)] &= \frac{1}{\rho_r(L)} \sum_{\mathbf{x} \in L} \exp\left(-\pi(\sqrt{2}x_1/r)^2\right) \exp(-\pi(x_2/r)^2) \cdots \exp(-\pi(x_n/r)^2) \\ &= \frac{\det(L^*) r^n}{\sqrt{2}\rho_r(L)} \sum_{\mathbf{y} \in L^*} \exp\left(-\pi(ry_1/\sqrt{2})^2\right) \exp(-\pi(ry_2)^2) \cdots \exp(-\pi(ry_n)^2) \\ &\leq \frac{\det(L^*) r^n}{\sqrt{2}\rho_r(L)} \rho_{\sqrt{2}/r}(L^*) \\ &\leq \frac{\det(L^*) r^n}{\sqrt{2}\rho_r(L)} (1 + \varepsilon). \end{aligned}$$

By using Lemma 2.14 again, we see that $\rho_r(L) = \det(L^*) r^n \rho_{1/r}(L^*) \geq \det(L^*) r^n$. Therefore, the expectation above is at most $\frac{1}{\sqrt{2}}(1 + \varepsilon) < 0.9$ and the lemma follows. ■

Corollary 3.14 *Let L be an n -dimensional lattice and let r be such that $r \geq \sqrt{2}\eta_\varepsilon(L)$ where $\varepsilon \leq \frac{1}{10}$. Then, the probability that a set of n^2 vectors chosen independently from $D_{L,r}$ contains no n linearly independent vectors is exponentially small.*

Proof: Let $\mathbf{x}_1, \dots, \mathbf{x}_{n^2}$ be n^2 vectors chosen independently from $D_{L,r}$. For $i = 1, \dots, n$, let B_i be the event that

$$\dim \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_{(i-1)n}) = \dim \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_{in}) < n.$$

Clearly, if none of the B_i 's happens, then $\dim \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_{n^2}) = n$. Hence, in order to complete the proof it suffices to show that for all i , $\Pr[B_i] \leq 2^{-\Omega(n)}$. So fix some i , and let us condition on some fixed choice of $\mathbf{x}_1, \dots, \mathbf{x}_{(i-1)n}$ such that $\dim \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_{(i-1)n}) < n$. By Lemma 3.13, the probability that

$$\mathbf{x}_{(i-1)n+1}, \dots, \mathbf{x}_{in} \in \dim \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_{(i-1)n})$$

is at most $(9/10)^n = 2^{-\Omega(n)}$. This implies that $\Pr[B_i] \leq 2^{-\Omega(n)}$, as required. \blacksquare

In the following lemma we give the reduction from SIVP (in fact, GIVP) to DGS. It shows that under the assumptions of Theorem 3.1, there exists an efficient quantum algorithm for $\text{GIVP}_{2\sqrt{2n}\eta_\varepsilon(L)/\alpha}$. By Lemma 2.12, this algorithm also solves $\text{SIVP}_{\tilde{O}(n/\alpha)}$.

Lemma 3.15 *For any $\varepsilon = \varepsilon(n) \leq \frac{1}{10}$ and any $\varphi(L) \geq \sqrt{2}\eta_\varepsilon(L)$, there is a polynomial time reduction from $\text{GIVP}_{2\sqrt{n}\varphi}$ to DGS_φ .*

Proof: As mentioned above, the idea of the reduction is to simply call the DGS oracle in an attempt to find n short linearly independent vectors. One technical complication is that the function φ is not necessarily efficiently computable, and hence we do not know which radius r to give the DGS oracle. The solution is easy: we just try many values of r and take the shortest set of n linearly independent vectors found.

We now present the reduction in detail. The input to the reduction is a lattice L . We first apply the LLL algorithm [20] to obtain n linearly independent vectors of length at most $2^n \lambda_n(L)$. Let S denote the resulting set, and let $\tilde{\lambda}_n$ be the length of the longest vector in S . By construction we have $\lambda_n(L) \leq \tilde{\lambda}_n \leq 2^n \lambda_n(L)$. For each $i \in \{0, \dots, 2n\}$ call the DGS oracle n^2 times with the pair (L, r_i) where $r_i = \tilde{\lambda}_n 2^{-i}$, and let S_i be the resulting set of vectors. At the end, look for a set of n linearly independent vectors in each of $S, S_0, S_1, \dots, S_{2n}$, and output the shortest set found.

We now prove correctness. If $\varphi(L) \geq \tilde{\lambda}_n$ then S is already shorter than $2\sqrt{n}\varphi(L)$ and so we are done. Otherwise, let $i \in \{0, \dots, 2n\}$ be such that $\varphi(L) < r_i \leq 2\varphi(L)$. Such an i must exist by Claim 2.13. By Corollary 3.14, S_i contains n linearly independent vectors with probability exponentially close to 1. Moreover, by Lemma 2.5, all vectors in S_i are of length at most $r_i \sqrt{n} \leq 2\sqrt{n}\varphi(L)$ with probability exponentially close to 1. Hence, our reduction outputs a set of n linearly independent vectors of length at most $2\sqrt{n}\varphi(L)$, as required. \blacksquare

We now present the reduction from GAPSVP to DGS. We first define the decision version of the closest vector problem (GAPCVP) and a slight variant of it.

Definition 3.16 *An instance of GAPCVP_γ is given by an n -dimensional lattice L , a vector \mathbf{t} , and a number $d > 0$. In YES instances, $\text{dist}(\mathbf{t}, L) \leq d$, whereas in NO instances, $\text{dist}(\mathbf{t}, L) > \gamma(n) \cdot d$.*

Definition 3.17 *An instance of GAPCVP'_γ is given by an n -dimensional lattice L , a vector \mathbf{t} , and a number $d > 0$. In YES instances, $\text{dist}(\mathbf{t}, L) \leq d$. In NO instances, $\lambda_1(L) > \gamma(n) \cdot d$ and $\text{dist}(\mathbf{t}, L) > \gamma(n) \cdot d$.*

In [14] it is shown that for any $\gamma = \gamma(n) \geq 1$, there is a polynomial time reduction from GAPSV_γ to GAPCVP'_γ (see also Lemma 5.22 in [24]). Hence, it suffices to show a reduction from GAPCVP' to DGS. This reduction is given in the following lemma. By using Lemma 2.11, we obtain that under the assumptions of Theorem 3.1 there exists an efficient quantum algorithm for $\text{GAPCVP}'_{O(n/\alpha)}$ (and hence also for $\text{GAPSV}_{O(n/\alpha)}$).

Lemma 3.18 *For any $\gamma = \gamma(n) \geq 1$, there is a polynomial time reduction from $\text{GAPCVP}'_{100\sqrt{n}\cdot\gamma(n)}$ to $\text{DGS}_{\sqrt{n}\gamma(n)/\lambda_1(L^*)}$.*

Proof: The main component in our reduction is the NP verifier for COGAPCVP shown in [1]. In more detail, [1] present an efficient algorithm, call it \mathcal{V} , whose input consists of an n -dimensional lattice L , a vector \mathbf{t} , a number $d > 0$, and a sequence of vectors $\mathbf{w}_1, \dots, \mathbf{w}_N$ in L^* for some $N = \text{poly}(n)$. When $\text{dist}(\mathbf{t}, L) \leq d$, the algorithm is guaranteed to reject. When $\text{dist}(\mathbf{t}, L) > 100\sqrt{n}d$, and $\mathbf{w}_1, \dots, \mathbf{w}_N$ are chosen from the distribution $D_{L^*, 1/(100d)}$, then the algorithm accepts with probability exponentially close to 1.

The input to the reduction is an n -dimensional lattice L , a vector \mathbf{t} , and a number $d > 0$. We call the DGS oracle N times with the lattice L^* and the value $\frac{1}{100d}$ to obtain vectors $\mathbf{w}_1, \dots, \mathbf{w}_N \in L^*$. We then apply \mathcal{V} with L , \mathbf{t} , d , and the vectors $\mathbf{w}_1, \dots, \mathbf{w}_N$. We accept if and only if \mathcal{V} rejects.

To prove correctness, notice first that in the case of a YES instance, $\text{dist}(\mathbf{t}, L) \leq d$, and hence \mathcal{V} must reject (irrespective of the \mathbf{w} 's). In the case of a NO instance we have that $\frac{1}{100d} > \sqrt{n}\gamma(n)/\lambda_1(L)$, and hence $\mathbf{w}_1, \dots, \mathbf{w}_N$ are guaranteed to be valid samples from $D_{L^*, 1/(100d)}$. Moreover, $\text{dist}(\mathbf{t}, L) > 100\sqrt{n}\gamma(n)d \geq 100\sqrt{n}d$, and hence \mathcal{V} accepts with probability exponentially close to 1. ■

4 Variants of the LWE problem

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

Proof: The proof is based on the following transformation. For any $\mathbf{t} \in \mathbb{Z}_p^n$ consider the function $f_{\mathbf{t}} : \mathbb{Z}_p^n \times \mathbb{Z}_p \rightarrow \mathbb{Z}_p^n \times \mathbb{Z}_p$ 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 $O(n^{2c_2+1})$ times on each of

the input distributions. By the Chernoff bound, 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 $A_{s,\chi}$ from U for all s . First, we claim that when R is U , the acceptance probability of W' is exponentially close to 0. Indeed, in this case, $f_t(U) = U$ 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 $A_{s,\chi}$ for some s . In each of the n^{c_1+1} iterations, we are considering the distribution $f_t(A_{s,\chi}) = A_{s+t,\chi}$ for some uniformly chosen t . Notice that the distribution of $s+t$ is uniform on \mathbb{Z}_p^n . Hence, with probability exponentially close to 1, in one of the n^{c_1+1} iterations, t is such that the acceptance probability of W on inputs from $A_{s+t,\chi}$ 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 close to 1. ■

Lemma 4.2 (Decision to Search) *Let $n \geq 1$ be some integer, $2 \leq p \leq \text{poly}(n)$ be a prime, and χ be some distribution on \mathbb{Z}_p . Assume that we have access to procedure W that for all s accepts with probability exponentially close to 1 on inputs from $A_{s,\chi}$ and rejects with probability exponentially close to 1 on inputs from U . Then, there exists an efficient algorithm W' that, given samples from $A_{s,\chi}$ for some s , outputs s with probability exponentially close to 1.*

Proof: Let us show how W' finds $s_1 \in \mathbb{Z}_p$, the first coordinate of 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 $(\mathbf{a} + (l, 0, \dots, 0), b + l \cdot k)$ 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 $A_{s,\chi}$ to itself. Finally, if $k \neq s_1$ then it takes $A_{s,\chi}$ 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 $p < \text{poly}(n)$ possibilities for s_1 we can try all of them. ■

Lemma 4.3 (Discrete to Continuous) *Let $n, p \geq 1$ be some integers, let ϕ be some probability density function on \mathbb{T} , and let $\bar{\phi}$ be its discretization to \mathbb{Z}_p . Assume that we have access to an algorithm W that solves $\text{LWE}_{p,\bar{\phi}}$. Then, there exists an efficient algorithm W' that solves $\text{LWE}_{p,\phi}$.*

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

By combining the three lemmas above, we obtain

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

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 $p \geq 2$ to be some prime number between n^2 and $2n^2$ and let $m = (1 + \varepsilon)(n + 1) \log p$ for some arbitrary constant $\varepsilon > 0$. 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 $\lim_{n \rightarrow \infty} \alpha(n) \cdot \sqrt{n} \log n = 0$. For example, we can choose $\alpha(n) = 1/(\sqrt{n} \log^2 n)$. 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 $i = 1, \dots, m$, choose m vectors $\mathbf{a}_1, \dots, \mathbf{a}_m \in \mathbb{Z}_p^n$ independently from the uniform distribution. Also choose elements $e_1, \dots, e_m \in \mathbb{Z}_p$ 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 set S uniformly among all 2^m subsets of $[m]$. The encryption is $(\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i)$ if the bit is 0 and $(\sum_{i \in S} \mathbf{a}_i, \lfloor \frac{p}{2} \rfloor + \sum_{i \in S} b_i)$ 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 the encryption process increases the size of a message by a factor of $O(n \log p) = \tilde{O}(n)$. In fact, it is possible to reduce the size of the public key to $O(m \log p) = \tilde{O}(n)$ by the following idea of Ajtai [3]. Assume all users of the cryptosystem share some fixed (and trusted) random choice of $\mathbf{a}_1, \dots, \mathbf{a}_m$. This can be achieved by, say, distributing these vectors as part of the encryption and decryption software. Then the public key need only consist of b_1, \dots, b_m . This modification does not affect the security of the cryptosystem.

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 χ^{*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 χ^{*0} as the distribution that is constantly 0). For a probability distribution ϕ on \mathbb{T} we define ϕ^{*k} similarly. For an element $a \in \mathbb{Z}_p$ we define $|a|$ as the integer a if $a \in \{0, 1, \dots, \lfloor \frac{p}{2} \rfloor\}$ 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 $k \in \{0, 1, \dots, m\}$, χ^{*k} satisfies that*

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

Then, the probability of decryption error is at most δ . That is, for any bit $c \in \{0, 1\}$, if we use the protocol above to choose private and public keys, encrypt c , and then decrypt the result, then the outcome is c with probability at least $1 - \delta$.

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 $\chi^{*|S|}$. 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 $k \in \{0, 1, \dots, m\}$,

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

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

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

In order to 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 be some integer. For any l elements $g_1, \dots, g_l \in G$ consider the statistical distance between the uniform distribution on G and the distribution given by the sum of a random subset of g_1, \dots, g_l . Then the expectation of this statistical distance over a uniform choice of $g_1, \dots, g_l \in G$ is at most $\sqrt{|G|/2^l}$. In particular, the probability that this statistical distance is more than $\sqrt[4]{|G|/2^l}$ is at most $\sqrt[4]{|G|/2^l}$.

Proof: For a choice $\mathbf{g} = (g_1, \dots, g_l)$ of l elements from G , let $P_{\mathbf{g}}$ be the distribution of the sum of a random subsets of g_1, \dots, g_l , i.e.,

$$P_{\mathbf{g}}(h) = \frac{1}{2^l} \left| \left\{ \mathbf{b} \in \{0, 1\}^l \mid \sum_i b_i g_i = h \right\} \right|.$$

In order to show that this distribution is close to uniform, we compute its ℓ_2 norm, and note that it is very close to $1/|G|$. From this it will follow that the distribution must be close to the uniform distribution (this idea was used in [16]). The ℓ_2 norm of $P_{\mathbf{g}}$ is given by

$$\begin{aligned} \sum_{h \in G} P_{\mathbf{g}}(h)^2 &= \Pr_{\mathbf{b}, \mathbf{b}'} \left[\sum b_i g_i = \sum b'_i g_i \right] \\ &\leq \frac{1}{2^l} + \Pr_{\mathbf{b}, \mathbf{b}'} \left[\sum b_i g_i = \sum b'_i g_i \mid \mathbf{b} \neq \mathbf{b}' \right]. \end{aligned}$$

Taking expectation over \mathbf{g} , and using the fact that for any $\mathbf{b} \neq \mathbf{b}'$, $\Pr_{\mathbf{g}}[\sum b_i g_i = \sum b'_i g_i] = 1/|G|$, we obtain that

$$\mathbb{E}_{\mathbf{g}} \left[\sum_h P_{\mathbf{g}}(h)^2 \right] \leq \frac{1}{2^l} + \frac{1}{|G|}.$$

Finally, the expected distance from the uniform distribution is

$$\begin{aligned} \mathbb{E}_{\mathbf{g}} \left[\sum_h |P_{\mathbf{g}}(h) - 1/|G|| \right] &\leq \mathbb{E}_{\mathbf{g}} \left[|G|^{1/2} \left(\sum_h (P_{\mathbf{g}}(h) - 1/|G|)^2 \right)^{1/2} \right] \\ &= \sqrt{|G|} \mathbb{E}_{\mathbf{g}} \left[\left(\sum_h P_{\mathbf{g}}(h)^2 - 1/|G| \right)^{1/2} \right] \\ &\leq \sqrt{|G|} \left(\mathbb{E}_{\mathbf{g}} \left[\sum_h P_{\mathbf{g}}(h)^2 \right] - 1/|G| \right)^{1/2} \\ &\leq \sqrt{\frac{|G|}{2^l}}. \end{aligned}$$

■

We now prove that our cryptosystem is semantically secure, i.e., that it is hard to distinguish between encryptions of 0 and encryptions of 1. More precisely, we show that if such a distinguisher exists, then there exists a distinguisher that distinguishes between $A_{s,\chi}$ and U for a non-negligible fraction of all s . If $\chi = \bar{\Psi}_\alpha$ and $p \leq \text{poly}(n)$ is a prime, then by Lemma 4.4, this also implies an efficient (classical) algorithm that solves $\text{LWE}_{p,\Psi_\alpha}$. This in turn implies, by Theorem 3.1, an efficient quantum algorithm for $\text{DGS}_{\sqrt{2n}\cdot\eta_\varepsilon(L)/\alpha}$. Finally, by Lemma 3.15 we also obtain an efficient quantum algorithm for $\text{SIVP}_{\tilde{O}(n/\alpha)}$ and by Lemma 3.18 we obtain an efficient quantum algorithm for $\text{GAPSV}_{O(n/\alpha)}$.

Lemma 5.4 (Security) *For any $\varepsilon > 0$ and $m \geq (1 + \varepsilon)(n + 1) \log p$, 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 $A_{s,\chi}$ and U for a non-negligible fraction of all possible 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.

For $\mathbf{s} \in \mathbb{Z}_p^n$, 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 $\mathbb{Z}_p^n \times \mathbb{Z}_p$. Our assumption on W' says that $|\text{Exp}_{\mathbf{s}}[p_0(\mathbf{s})] - \text{Exp}_{\mathbf{s}}[p_u(\mathbf{s})]| \geq \frac{1}{2n^c}$. Define

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

By an averaging argument we get that a fraction of at least $\frac{1}{4n^c}$ of the \mathbf{s} are in Y . 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 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 uniform element of $\mathbb{Z}_p^n \times \mathbb{Z}_p$. By applying W' a polynomial number of times, 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,

$$|p_0((\mathbf{a}_i, b_i)_{i=1}^m) - p_u((\mathbf{a}_i, b_i)_{i=1}^m)| \leq 2^{-\Omega(n)}.$$

Hence, our two estimates differ by at most $\frac{1}{32n^c} + 2^{-\Omega(n)}$, and Z rejects.

Next, we show that if R is $A_{s,\chi}$ for $s \in Y$ then Z accepts with probability $1/\text{poly}(n)$. Notice that $p_0(s)$ (respectively, $p_u(s)$) 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 $A_{s,\chi}$. From $|p_0(s) - p_u(s)| \geq \frac{1}{4n^c}$ we obtain by an averaging argument that

$$|p_0((\mathbf{a}_i, b_i)_{i=1}^m) - p_u((\mathbf{a}_i, b_i)_{i=1}^m)| \geq \frac{1}{8n^c}$$

with probability at least $\frac{1}{8n^c}$ over the choice of $(\mathbf{a}_i, b_i)_{i=1}^m$ from $A_{s,\chi}$. Hence, with probability at least $\frac{1}{8n^c}$, 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. ■

Acknowledgments

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

References

- [1] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. *Journal of the ACM*, 52(5):749–765, 2005. Preliminary version in FOCS’04.
- [2] M. Ajtai. Generating hard instances of lattice problems. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 1–32. Dept. Math., Seconda Univ. Napoli, Caserta, 2004. Preliminary version in STOC 1996.
- [3] M. Ajtai. Representing hard lattices with $O(n \log n)$ bits. In *Proc. 37th Annual ACM Symp. on Theory of Computing (STOC)*, pages 94–103, 2005.
- [4] 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.
- [5] 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.
- [6] M. Alekhovich. More on average case vs approximation complexity. In *Proc. 44th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 298–307, 2003.
- [7] L. Babai. On Lovasz’ lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
- [8] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.
- [15] L. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. In *quant-ph/0208112*, <http://xxx.lanl.gov>, 2002.
- [16] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proc. 30th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 248–253, 1989.
- [17] A. Kawachi, K. Tanaka, and K. Xagawa. Multi-bit cryptosystems based on lattice problems. In *Public Key Cryptography – PKC 2007*, volume 4450 of *Lecture Notes in Comput. Sci.*, pages 315–329, Berlin, 2007. Springer.
- [18] A. Klivans and A. Sherstov. Cryptographic hardness results for learning intersections of halfspaces. In *Proc. 47th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, 2006.
- [19] 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.
- [20] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.
- [21] Y.-K. Liu, V. Lyubashevsky, and D. Micciancio. On bounded distance decoding for general lattices. In *International Workshop on Randomization and Computation - Proceedings of RANDOM 2006*, volume 4110 of *Lecture Notes in Comput. Sci.*, pages 450–461, Barcelona, Spain, Aug. 2006. Springer.
- [22] D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai’s connection factor. *SIAM Journal on Computing*, 34(1):118–169, 2004. Preliminary version in STOC 2002.
- [23] 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.

- [24] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.
- [25] C. Moore, A. Russell, and U. Vazirani. A classical one-way function to confound quantum adversaries. In *quant-ph/0701115*, <http://xxx.lanl.gov>, 2007.
- [26] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [27] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for oblivious transfer, 2007. In preparation.
- [28] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. Cryptology ePrint Archive, Report 2007/279, 2007.
- [29] C. J. Peikert. Limits on the hardness of lattice problems in ℓ_p norms. In *Proc. of 22nd IEEE Annual Conference on Computational Complexity (CCC)*, pages 333–346, 2007.
- [30] O. Regev. New lattice based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, 2004. Preliminary version in STOC’03.
- [31] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. 37th ACM Symp. on Theory of Computing (STOC)*, pages 84–93, 2005.
- [32] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoret. Comput. Sci.*, 53(2-3):201–224, 1987.