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

Oded Regev *

May 2, 2009

Abstract

Our main result is a reduction from worst-case lattice problems such as GAPSVP 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 GAPSVP and SIVP. A main open question is whether this reduction can be made classical (i.e., non-quantum).

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 GAPSVP and SIVP. The new cryptosystem is much more efficient than previous lattice-based 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

*School 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 Army Research Office grant DAAD19-03-1-0082, by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, and by a European Research Council (ERC) Starting Grant.

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 $\mathrm{poly}(n)$ time.

The problem seems to become significantly harder when we take any positive $\varepsilon > 0$. For example, let us consider again the Gaussian elimination process and assume 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 $(1, 0, \ldots, 0)$. Summing the corresponding values b_i gives us a guess for the first bit of $\mathbf{s}$. However, a standard calculation shows that this guess is correct with probability $\frac{1}{2} + 2^{-\Theta(n)}$. Hence, in order to obtain the first bit with good confidence, we have to repeat the whole procedure $2^{\Theta(n)}$ times. This yields an algorithm that uses $2^{O(n)}$ equations and $2^{O(n)}$ time. In fact, it can be shown that given only $O(n)$ equations, the $\mathbf{s}' \in \mathbb{Z}_2^n$ that maximizes the number of satisfied equations is with high probability $\mathbf{s}$. This yields a simple maximum likelihood algorithm that requires only $O(n)$ equations and runs in time $2^{O(n)}$.

Blum, Kalai, and Wasserman [11] provided the first subexponential algorithm for this problem. Their algorithm requires only $2^{O(n/\log n)}$ equations/time and is currently the best known algorithm for the problem. It is based on a clever idea that allows to find a small set S of equations (say, $O(\sqrt{n})$) among $2^{O(n/\log n)}$ equations, such that $\sum_S \mathbf{a}_i$ is, say, $(1, 0, \ldots, 0)$. This gives us a guess for the first bit of $\mathbf{s}$ that is correct with probability $\frac{1}{2} + 2^{-\Theta(\sqrt{n})}$. We can obtain the correct value with high probability by repeating the whole procedure only $2^{O(\sqrt{n})}$ times. Their idea was later shown to have other important applications, such as the first $2^{O(n)}$-time algorithm for solving the shortest vector problem [23, 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 $p = p(n) \leq \mathrm{poly}(n)$ be some prime integer and consider a list of 'equations with error'

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

where this time $\mathbf{s} \in \mathbb{Z}_p^n$, $\mathbf{a}_i$ are chosen independently and uniformly from $\mathbb{Z}_p^n$, and $b_i \in \mathbb{Z}_p$. The error in the equations is now specified by a probability distribution $\chi : \mathbb{Z}_p \to \mathbb{R}^+$ on $\mathbb{Z}_p$. Namely, for each equation i, $b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i$ where each $e_i \in \mathbb{Z}_p$ is chosen independently according to χ. We denote the problem of recovering $\mathbf{s}$ from such equations by $\mathrm{LWE}_{p,\chi}$ (learning with error). For example, the learning from parity problem with error ε is the special case where $p = 2$, $\chi(0) = 1 - \varepsilon$, and $\chi(1) = \varepsilon$. Under a reasonable assumption on χ (namely, that $\chi(0) > 1/p + 1/\mathrm{poly}(n)$), the maximum likelihood algorithm described above solves $\mathrm{LWE}_{p,\chi}$ for $p \leq \mathrm{poly}(n)$ using $\mathrm{poly}(n)$ equations and $2^{O(n \log n)}$ time. Under a similar assumption, an algorithm resembling the one by Blum et al. [11] 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 $\mathrm{LWE}_{p,\chi}$ 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 $\mathrm{LWE}_{p,\bar{\Psi}_\alpha}$ then there exists an efficient quantum algorithm that approximates the decision version of the shortest vector problem (*GapSVP*) 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).

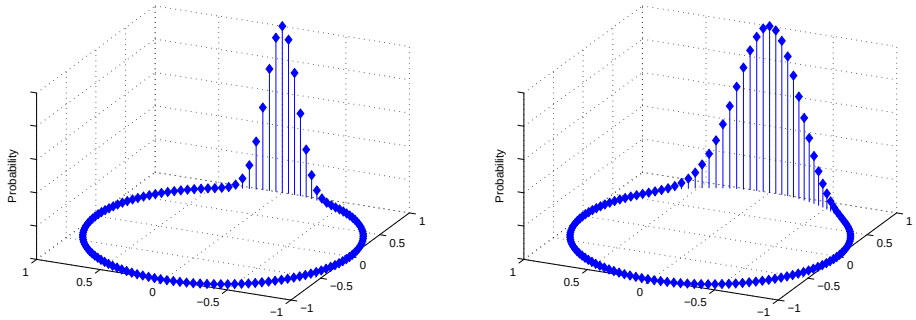

Figure 1: $\bar{\Psi}_\alpha$ for $p = 127$ with $\alpha = 0.05$ (left) and $\alpha = 0.1$ (right). The elements of $\mathbb{Z}_p$ are arranged on a circle.

GAPSVP and SIVP are two of the main computational problems on lattices. In GAPSVP, for instance, the input is a lattice, and the goal is to approximate the length of the shortest nonzero lattice vector. The best known polynomial time algorithms for them yield only mildly subexponential approximation factors [24, 38, 5]. It is conjectured that there is no classical (i.e., non-quantum) 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 GAPSVP (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 $m = \mathrm{poly}(n)$ be arbitrary and let $\mathbf{s} \in \mathbb{Z}_p^n$ be some vector. Then, consider the following problem: given a random matrix $Q \in \mathbb{Z}_p^{m\times n}$ and the vector $\mathbf{t} = Q\mathbf{s} + \mathbf{e} \in \mathbb{Z}_p^m$ where each coordinate of the error vector $\mathbf{e} \in \mathbb{Z}_p^m$ is chosen independently from $\bar{\Psi}_\alpha$, recover $\mathbf{s}$. The Hamming weight of $\mathbf{e}$ is roughly $m(1 - 1/(\alpha p))$ (since a value chosen from $\bar{\Psi}_\alpha$ is 0 with probability roughly $1/(\alpha p)$). Hence, the Hamming distance of $\mathbf{t}$ from $Q\mathbf{s}$ is roughly $m(1 - 1/(\alpha p))$. Moreover, it can be seen that for large enough m, for any other word $\mathbf{s}'$, the Hamming distance of $\mathbf{t}$ from $Q\mathbf{s}'$ is roughly $m(1 - 1/p)$. Hence, we obtain that approximating the nearest codeword problem to within factors smaller than $(1 - 1/p)/(1 - 1/(\alpha p))$ on random codes is as hard as quantumly approximating worst-case lattice problems. This gives a partial

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

answer to the important open question of understanding the hardness of decoding from random linear codes.

It turns out that certain problems, which are seemingly easier than the LWE problem, are in fact equivalent to the LWE problem. We establish these equivalences 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 $\mathbf{s}$. The latter formulation is the one we use in our cryptographic applications.

Cryptosystem. In Section 5 we present a public key cryptosystem and prove that it is secure based on the hardness of the LWE problem. We use the standard security notion of semantic, or IND-CPA, security (see, e.g., [20, Chapter 10]). 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. Essentially, the idea is to provide a list of equations as above as the public key; encryption is performed by summing some of the equations (forming another equation with error) and modifying the right hand side depending on the message to be transmitted. Security follows from the fact that a list of equations with error is computationally indistinguishable from a list of equations in which the b_i's are chosen uniformly.

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 GapSVP to within $\tilde{O}(n^{1.5})$. In other words, breaking our cryptosystem implies an efficient quantum algorithm for approximating SIVP and GapSVP to within $\tilde{O}(n^{1.5})$. Previous cryptosystems, such as the Ajtai-Dwork cryptosystem [4] and the one by Regev [36], are based on the worst-case (classical) hardness of the unique-SVP problem, which can be related to GapSVP (but not SIVP) through the recent result of Lyubashevsky and Micciancio [26].

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., [10, 7], although this was done without establishing any connection to lattice problems. In another related work [3], Ajtai suggested a cryptosystem that has several properties in common with ours (including its efficiency), although its security is not based on worst-case lattice problems.

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

at most d. Clearly, on input $\mathbf{x}$ the oracle outputs $\mathbf{y}$. But this is useless since we already know $\mathbf{y}$!

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

Techniques. 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, 12, 27, 29]. 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 [30] 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 [36, 29, 1]. More specifically, we use a parameter known as the smoothing parameter, as introduced in [29]. We also use the discrete Gaussian distribution and approximations to its Fourier transform, ideas that were developed in [1].

Open questions. The main open question raised by this work is whether Theorem 1.1 can be dequantized: can the hardness of LWE be established based on the classical hardness of SIVP and GAPSVP? 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 can interpret our inability to prove hardness for small p as an indication that the problem might be easier than believed.

Finally, it would be interesting to relate the LWE problem to other average-case problems in the literature, and especially to those considered by Feige in [15]. See Alekhnovich's paper [7] for some related work.

Followup work. We now describe some of the followup work that has appeared since the original publication of our results in 2005 [37].

One line of work focussed on improvements to our cryptosystem. First, Kawachi, Tanaka, and Xagawa [21] 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, Vaikuntanathan, and Waters in [34]. 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. Finally, Akavia, Goldwasser, and Vaikuntanathan [6] show that our cryptosystem remains secure even if almost the entire secret key is leaked.

Another line of work focussed on the design of other cryptographic protocols whose security is based on the hardness of the LWE problem. First, Peikert and Waters [35] constructed, among other things, CCA-

secure cryptosystems (see also [33] for a simpler construction). These are cryptosystems that are secure even if the adversary is allowed access to a decryption oracle (see, e.g., [20, Chapter 10]). All previous lattice-based cryptosystems (including the one in this paper) are not CCA-secure. Second, Peikert, Vaikuntanathan, and Waters [34] showed how to construct oblivious transfer protocols, which are useful, e.g., for performing secure multiparty computation. Third, Gentry, Peikert, and Vaikuntanathan [16] constructed an identity-based encryption (IBE) scheme. This is a public-key encryption scheme in which the public key can be any unique identifier of the user; very few constructions of such schemes are known. Finally, Cash, Peikert, and Sahai [13] constructed a public-key cryptosystem that remains secure even when the encrypted messages may depend upon the secret key. The security of all the above constructions is based on the LWE problem and hence, by our main theorem, also on the worst-case quantum hardness of lattice problems.

The LWE problem has also been used by Klivans and Sherstov to show hardness results related to learning halfspaces [22]. As before, due to our main theorem, this implies hardness of learning halfspaces based on the worst-case quantum hardness of lattice problems.

Finally, we mention two results giving further evidence for the hardness of the LWE problem. In the first, Peikert [32] somewhat strengthens our main theorem by replacing our worst-case lattice problems with their analogues for the ℓ_q norm, where $2 \leq q \leq \infty$ is arbitrary. Our main theorem only deals with the standard ℓ_2 versions.

In another recent result, Peikert [33] shows that the quantum part of our proof can be removed, leading to a *classical* reduction from GAPSVP to the LWE problem. As a result, Peikert is able to show that public-key cryptosystems (including many of the above LWE-based schemes) can be based on the classical hardness of GAPSVP, resolving a long-standing open question (see also [26]). Roughly speaking, the way Peikert circumvents the difficulty we described earlier is by noticing that the *existence* of an oracle that is able to recover $\mathbf{y}$ from $\mathbf{y} + \mathbf{z}$, where $\mathbf{y}$ is a random lattice point and $\mathbf{z}$ is a random perturbation of length at most d, is by itself a useful piece of information as it provides a lower bound on the length of the shortest nonzero vector. By trying to construct such oracles for several different values of d and checking which ones work, Peikert is able to obtain a good approximation of the length of the shortest nonzero vector.

Removing the quantum part, however, comes at a cost: the construction can no longer be iterative, the hardness can no longer be based on SIVP, and even for hardness based on GAPSVP, the modulus p in the LWE problem must be exponentially big unless we assume the hardness of a non-standard variant of GAPSVP. Because of this, we believe that dequantizing our main theorem remains an important open problem.

1.1 Overview

In this subsection, we give a brief informal overview of the proof of our main theorem, Theorem 1.1. The complete proof appears in Section 3. We do not discuss here the reductions in Section 4 and the cryptosystem in Section 5 as these parts of the paper are more similar to previous work.

In addition to some very basic definitions related to lattices, we will make heavy use here of the *discrete Gaussian distribution on L of width r*, denoted $D_{L,r}$. This is the distribution whose support is L (which is typically a lattice), and in which the probability of each $\mathbf{x} \in L$ is proportional to $\exp\left(-\pi\|\mathbf{x}/r\|^2\right)$ (see Eq. (6) and Figure 2). We also mention here the *smoothing parameter* $\eta_\varepsilon(L)$. This is a real positive number associated with any lattice L (ε is an accuracy parameter which we can safely ignore here). Roughly speaking, it gives the smallest r starting from which $D_{L,r}$ 'behaves like' a continuous Gaussian distribution. For instance, for $r \geq \eta_\varepsilon(L)$, vectors chosen from $D_{L,r}$ have norm roughly $r\sqrt{n}$ with high probability. In

contrast, for sufficiently small r, $D_{L,r}$ gives almost all its mass to the origin 0. Although not required for this section, a complete list of definitions can be found in Section 2.

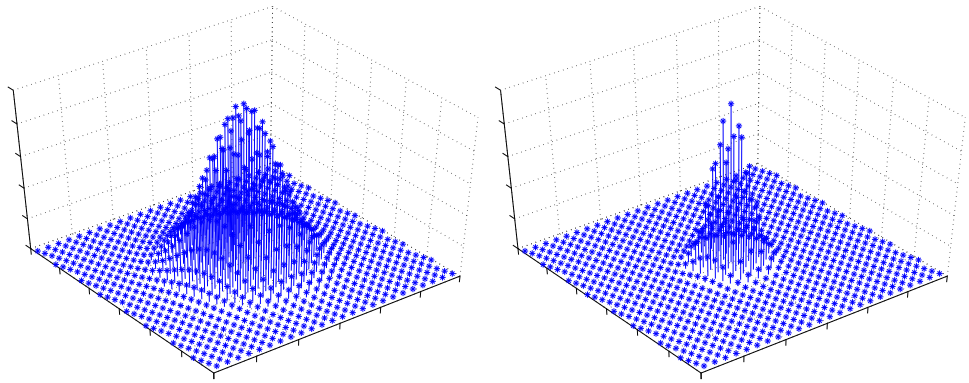

Figure 2: $D_{L,2}$ (left) and $D_{L,1}$ (right) for a two-dimensional lattice L. The z-axis represents probability.

Let α, p, n be such that $\alpha p > 2\sqrt{n}$, as required in Theorem 1.1, and assume we have an oracle that solves $\mathrm{LWE}_{p,\bar{\Psi}_\alpha}$. For concreteness, we can think of $p = n^2$ and $\alpha = 1/n$. Our goal is to show how to solve the two lattice problems mentioned in Theorem 1.1. As we prove in Subsection 3.3 using standard reductions, it suffices to solve the following *discrete Gaussian sampling problem* (DGS): Given an n-dimensional lattice L and a number $r \geq \sqrt{2n} \cdot \eta_\varepsilon(L)/\alpha$, output a sample from $D_{L,r}$. Intuitively, the connection to GAPSVP and SIVP comes from the fact that by taking r close to its lower limit $\sqrt{2n} \cdot \eta_\varepsilon(L)/\alpha$, we can obtain short lattice vectors (of length roughly $\sqrt{n}r$). In the rest of this subsection we describe our algorithm for sampling from $D_{L,r}$. We note that the exact lower bound on r is not that important for purposes of this overview, as it only affects the approximation factor we obtain for GAPSVP and SIVP. It suffices to keep in mind that our goal is to sample from $D_{L,r}$ for r that is rather small, say within a polynomial factor of $\eta_\varepsilon(L)$.

The core of the algorithm is the following procedure, which we call the 'iterative step'. Its input consists of a number r (which is guaranteed to be not too small, namely, greater than $\sqrt{2}p\eta_\varepsilon(L)$), and n^c samples from $D_{L,r}$ where c is some constant. Its output is a sample from the distribution $D_{L,r'}$ for $r' = r\sqrt{n}/(\alpha p)$. Notice that since $\alpha p > 2\sqrt{n}$, $r' < r/2$. In order to perform this 'magic' of converting vectors of norm $\sqrt{n}r$ into shorter vectors of norm $\sqrt{n}r'$, the procedure of course needs to use the LWE oracle.

Given the iterative step, the algorithm for solving DGS works as follows. Let r_i denote $r \cdot (\alpha p/\sqrt{n})^i$. The algorithm starts by producing n^c samples from $D_{L,r_{3n}}$. Because r_{3n} is so large, such samples can be computed efficiently by a simple procedure described in Lemma 3.2. Next comes the core of the algorithm: for $i = 3n, 3n-1, \ldots, 1$ the algorithm uses its n^c samples from D_{L,r_i} to produce n^c samples from $D_{L,r_{i-1}}$ by calling the iterative step n^c times. Eventually, 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. Note the following crucial fact: using n^c samples from D_{L,r_i}, we are able to generate the same number of samples n^c from $D_{L,r_{i-1}}$ (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, as this would require us to start with an exponential number of samples.

We now finally get to describe the iterative step. Recall that as input we have n^c samples from $D_{L,r}$ and we are supposed to generate a sample from $D_{L,r'}$ where $r' = r\sqrt{n}/(\alpha p)$. Moreover, r is known and guaranteed to be at least $\sqrt{2}p\eta_\varepsilon(L)$, which can be shown to imply that $p/r < \lambda_1(L^*)/2$. As mentioned above, the exact lower bound on r does not matter much for this overview; it suffices to keep in mind that r

is sufficiently larger than $\eta_\varepsilon(L)$, and that $1/r$ is sufficiently smaller than $\lambda_1(L^*)$.

The iterative step is obtained by combining two parts (see Figure 3). In the first part, we construct a classical algorithm that uses the given samples and the LWE oracle to solve the following closest vector problem, which we denote by $\mathrm{CVP}_{L^*,\alpha p/r}$: given any point $\mathbf{x} \in \mathbb{R}^n$ within distance $\alpha p/r$ of the dual lattice L^*, output the closest vector in L^* to $\mathbf{x}$.[2] By our assumption on r, the distance between any two points in L^* is greater than $2\alpha p/r$ and hence the closest vector is unique. In the second part, we use this algorithm to generate samples from $D_{L,r'}$. This part is quantum (and in fact, the only quantum part of our proof). The idea here is to use the $\mathrm{CVP}_{L^*,\alpha p/r}$ algorithm to generate a certain quantum superposition which, after applying the quantum Fourier transform and performing a measurement, provides us with a sample from $D_{L,r\sqrt{n}/(\alpha p)}$. In the following, we describe each of the two parts in more detail.

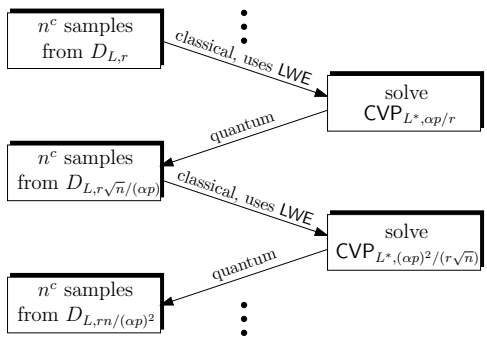

Figure 3: Two iterations of the algorithm

Part 1: We start by recalling the main idea in [1]. Consider some probability distribution D on some lattice L and consider its Fourier transform $f : \mathbb{R}^n \to \mathbb{C}$, defined as

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

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

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

where the approximation is to within $\pm 1/\mathrm{poly}(n)$ and holds with probability exponentially close to 1, assuming that N is a large enough polynomial.

By applying this idea to the samples from $D_{L,r}$ given to us as input, we obtain a good approximation of the Fourier transform of $D_{L,r}$, which we denote by $f_{1/r}$. It can be shown that since $1/r \ll \lambda_1(L^*)$ one has the approximation

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

[2] In fact, we only solve $\mathrm{CVP}_{L^*,\alpha p/(\sqrt{2}r)}$ but for simplicity we ignore the factor $\sqrt{2}$ here.

(see Figure 4). Hence, $f_{1/r}(\mathbf{x}) \approx 1$ for any $\mathbf{x} \in L^*$ (in fact an equality holds) 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 (roughly $\exp(-\pi)$). As the distance from L^* increases, the value of the function soon becomes negligible. Since the distance between any two vectors in L^* is at least $\lambda_1(L^*) \gg 1/r$, the Gaussians around each point of L^* are well-separated.

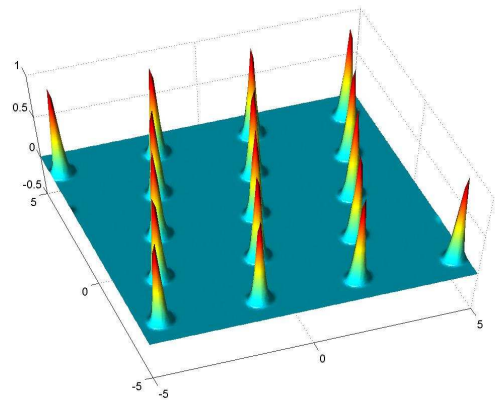

Figure 4: $f_{1/r}$ for a two-dimensional lattice

Although not needed in this paper, let us briefly outline how one can solve $\mathrm{CVP}_{L^*,1/r}$ using samples from $D_{L,r}$. 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 $f_{1/r}$. By repeatedly computing an approximation of $f_{1/r}$ using the samples from $D_{L,r}$ as described above, we 'walk 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 of $f_{1/r}$, we cannot find the closest lattice point exactly. It is possible to overcome this difficulty; see [25] for the details. The same procedure actually works for slightly longer distances, namely $O(\sqrt{\log n}/r)$, but beyond that distance the value of $f_{1/r}$ becomes negligible and no useful information can be extracted from our approximation of it.

Unfortunately, solving $\mathrm{CVP}_{L^*,1/r}$ is not useful for the iterative step as it would lead to samples from $D_{L,r\sqrt{n}}$, which is a wider rather than a narrower distribution than the one we started with. This is not surprising, since our solution to $\mathrm{CVP}_{L^*,1/r}$ did not use the LWE oracle. Using the LWE oracle, we will now show that we can gain an extra αp factor in the radius, and obtain the desired $\mathrm{CVP}_{L^*,\alpha p/r}$ algorithm.

Notice that if we could somehow obtain samples from $D_{L,r/p}$ we would be done: using the procedure described above, we could solve $\mathrm{CVP}_{L^*,p/r}$, which is better than what we need. Unfortunately, it is not clear how to obtain such samples, even with the help of the LWE oracle. Nevertheless, here is an obvious way to obtain something similar to samples from $D_{L,r/p}$: just take the given samples from $D_{L,r}$ and divide them by p. This provides us with samples from $D_{L/p,r/p}$ where L/p is the lattice L scaled down by a factor of p. In the following we will show how to use these samples to solve $\mathrm{CVP}_{L^*,\alpha p/r}$.

Let us first try to understand what the distribution $D_{L/p,r/p}$ looks like. Notice that the lattice L/p consists of p^n translates of the original lattice L. Namely, for each $\mathbf{a} \in \mathbb{Z}_p^n$, consider the set

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

Then $\{L + L\mathbf{a}/p \mid \mathbf{a} \in \mathbb{Z}_p^n\}$ forms a partition of L/p. Moreover, it can be shown that since r/p is larger

than the smoothing parameter $\eta_\varepsilon(L)$, the probability given to each $L + L\mathbf{a}/p$ under $D_{L/p,r/p}$ is essentially the same, that is, p^{-n}. Intuitively, beyond the smoothing parameter, the Gaussian measure no longer 'sees' the discrete structure of L, so in particular it is not affected by translations (this will be shown in Claim 3.8).

This leads us to consider the following distribution, call it $\tilde{D}$. A sample from $\tilde{D}$ is a pair $(\mathbf{a}, \mathbf{y})$ where $\mathbf{y}$ is sampled from $D_{L/p,r/p}$, and $\mathbf{a} \in \mathbb{Z}_p^n$ is such that $\mathbf{y} \in L + L\mathbf{a}/p$. Notice that we can easily obtain samples from $\tilde{D}$ using the given samples 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 $D_{L+L\mathbf{a}/p,r/p}$. Hence, $\tilde{D}$ is essentially identical to the distribution on pairs $(\mathbf{a}, \mathbf{y})$ in which $\mathbf{a} \in \mathbb{Z}_p^n$ is chosen uniformly at random, and then $\mathbf{y}$ is sampled from $D_{L+L\mathbf{a}/p,r/p}$. From now on, we think of $\tilde{D}$ as being this distribution.

We now examine the Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ (see Figure 5). When $\mathbf{a}$ is zero, we already know that the Fourier transform is $f_{p/r}$. For general $\mathbf{a}$, a standard calculation shows that the Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ is given by

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

where $\tau(\mathbf{x}) \in \mathbb{Z}_p^n$ is defined as

$$\tau(\mathbf{x}) := (L^*)^{-1} \kappa_{L^*}(\mathbf{x}) \bmod p,$$

and $\kappa_{L^*}(\mathbf{x})$ denotes the (unique) closest vector in L^* to $\mathbf{x}$. In other words, $\tau(\mathbf{x})$ is the vector of coefficients of the vector in L^* closest to $\mathbf{x}$ when represented in the basis of L^*, reduced modulo p. So we see that the Fourier transform $D_{L+L\mathbf{a}/p,r/p}$ is essentially $f_{p/r}$, except that each 'hill' gets its own phase depending on the vector of coefficients of the lattice point in its center. The appearance of these phases is as a result of a well-known property of the Fourier transform, saying that translation is transformed to multiplication by phase.

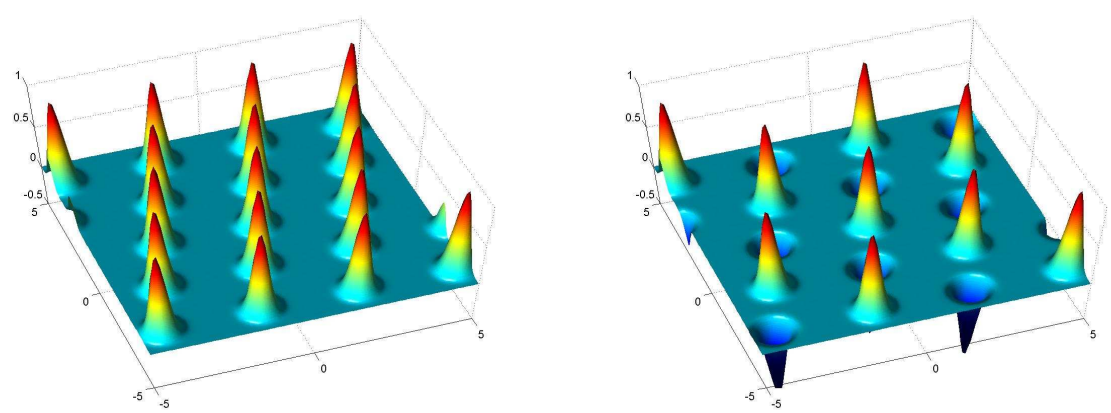

Figure 5: The Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$ with $n = 2$, $p = 2$, $\mathbf{a} = (0, 0)$ (left), $\mathbf{a} = (1, 1)$ (right).

Equipped with this understanding of the Fourier transform of $D_{L+L\mathbf{a}/p,r/p}$, we can get back to our task of solving $\mathrm{CVP}_{L^*,\alpha p/r}$. By the definition of the Fourier transform, we know that the average of $\exp\left(2\pi i \langle \mathbf{x}, \mathbf{y}\rangle\right)$ over $\mathbf{y} \sim D_{L+L\mathbf{a}/p,r/p}$ is given by (2). Assume for simplicity that $\mathbf{x} \in L^*$ (even though in this case finding the closest vector is trivial; it is simply $\mathbf{x}$ itself). In this case, (2) is equal to $\exp\left(2\pi i \langle \mathbf{a}, \tau(\mathbf{x})\rangle / p\right)$. Since the absolute value of this expression is 1, we see that for such $\mathbf{x}$, the random variable $\langle \mathbf{x}, \mathbf{y}\rangle \bmod 1$ (where

$\mathbf{y} \sim D_{L+L\mathbf{a}/p,r/p}$) must be deterministically equal to $\langle \mathbf{a}, \tau(\mathbf{x})\rangle/p \bmod 1$ (this fact can also be seen directly). In other words, when $\mathbf{x} \in L^*$, each sample $(\mathbf{a}, \mathbf{y})$ from $\tilde{D}$, provides us with a linear equation

$$\langle \mathbf{a}, \tau(\mathbf{x})\rangle = p\langle \mathbf{x}, \mathbf{y}\rangle \bmod p$$

with $\mathbf{a}$ distributed essentially uniformly in $\mathbb{Z}_p^n$. After collecting about n such equations, we can use Gaussian elimination to recover $\tau(\mathbf{x}) \in \mathbb{Z}_p^n$. And as we shall show in Lemma 3.5 using a simple reduction, the ability to compute $\tau(\mathbf{x})$ easily leads to the ability to compute the closest vector to $\mathbf{x}$.

We now turn to the more interesting case in which $\mathbf{x}$ is not in L^*, but only within distance $\alpha p/r$ of L^*. In this case, the phase of (2) is still equal to $\exp(2\pi i\langle \mathbf{a}, \tau(\mathbf{x})\rangle/p)$. Its absolute value, however, is no longer 1, but still quite close to 1 (depending on the distance of $\mathbf{x}$ from L^*). Therefore, the random variable $\langle \mathbf{x}, \mathbf{y}\rangle \bmod 1$, where $\mathbf{y} \sim D_{L+L\mathbf{a}/p,r/p}$, must be typically quite close to $\langle \mathbf{a}, \tau(\mathbf{x})\rangle/p \bmod 1$ (since, as before, the average of $\exp(2\pi i\langle \mathbf{x}, \mathbf{y}\rangle)$ is given by (2)). Hence, each sample $(\mathbf{a}, \mathbf{y})$ from $\tilde{D}$, provides us with a linear equation with error,

$$\langle \mathbf{a}, \tau(\mathbf{x})\rangle \approx \lfloor p\langle \mathbf{x}, \mathbf{y}\rangle \rceil \bmod p.$$

Notice that $p\langle \mathbf{x}, \mathbf{y}\rangle$ is typically not an integer and hence we round it to the nearest integer. After collecting a polynomial number of such equations, we call the LWE oracle in order to recover $\tau(\mathbf{x})$. Notice that $\mathbf{a}$ is distributed essentially uniformly, as required by the LWE oracle. Finally, as mentioned above, once we are able to compute $\tau(\mathbf{x})$, computing $\mathbf{x}$ is easy (this will be shown in Lemma 3.5).

The above outline ignores one important detail: what is the error distribution in the equations we produce? Recall that the LWE oracle is only guaranteed to work with error distribution $\bar{\Psi}_\alpha$. Luckily, as we will show in Claim 3.9 and Corollary 3.10 (using a rather technical proof), if $\mathbf{x}$ is at distance $\beta p/r$ from L^* for some $0 \le \beta \le \alpha$, then the error distribution in the equations is essentially $\bar{\Psi}_\beta$. (In fact, in order to get this error distribution, we will have to modify the procedure a bit and add a small amount of normal error to each equation.) We then complete the proof by noting (in Lemma 3.7) that an oracle for solving $\mathrm{LWE}_{p,\bar{\Psi}_\alpha}$ can be used to solve $\mathrm{LWE}_{p,\bar{\Psi}_\beta}$ for any $0 \le \beta \le \alpha$ (even if β is unknown).

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

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

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

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

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

(This step is actually unnecessary in the real procedure, since there we work in the basic parallelepiped of L^*; but for the present discussion, it is helpful to imagine that we start with this state.) On a separate register, we create a 'Gaussian state' of width $1/r$,

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

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

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

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

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

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

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

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 $\mathrm{CVP}_{L^*,\sqrt{n}/r}$ oracle. It can be shown that almost all the mass of $\exp\left(-\pi\|r\mathbf{z}\|^2\right)$ is on $\mathbf{z}$ such that $\|\mathbf{z}\| \leq \sqrt{n}/r$. 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 $D_{L,r}$ (or in fact from $D^2_{L,r} = D_{L,r/\sqrt{2}}$ 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. Some of the less standard notation is: the Gaussian function ρ (Eq. (4)), the Gaussian distribution ν (Eq. (5)), the periodic normal distribution Ψ (Eq. (7)), the discretization of a distribution on $\mathbb{T}$ (Eq. (8)), the discrete Gaussian distribution D (Eq. (6)), the unique closest lattice vector κ (above Lemma 2.3), and the smoothing parameter η (Definition 2.10).

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 $\lfloor x \rceil$ as the integer closest to x or, in case two such integers exist, the smaller of the two. For any integer $p \geq 2$, we write $\mathbb{Z}_p$ for the cyclic group $\{0, 1, \ldots, p-1\}$ with addition modulo p. We also write $\mathbb{T}$ for $\mathbb{R}/\mathbb{Z}$, i.e., the segment $[0, 1)$ with addition modulo 1.

We define a *negligible amount* in n as an amount that is asymptotically smaller than n^{-c} for any constant $c > 0$. More precisely, $f(n)$ is a negligible function in n if $\lim_{n\to\infty} n^c f(n) = 0$ for any $c > 0$. Similarly, a

non-negligible amount is one which is at least n^{-c} for some $c > 0$. Also, when we say that an expression is exponentially small in n we mean that it is at most $2^{-\Omega(n)}$. Finally, when we say that an expression (most often, some probability) is exponentially close to 1, we mean that it is $1 - 2^{-\Omega(n)}$.

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

Essentially all algorithms and reductions in this paper 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., [28]. In particular, this implies that the acceptance probability of any algorithm on inputs from X differs from its acceptance probability on inputs from Y by at most $\frac{1}{2}\Delta(X, Y)$ (the factor half coming from the choice of normalization in our definition of Δ).

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

$$\rho_s(\mathbf{x}) := \exp\left(-\pi\|\mathbf{x}/s\|^2\right) \tag{4}$$

be a Gaussian function scaled by a factor of s. We denote ρ_1 by ρ. Note that $\int_{\mathbf{x}\in\mathbb{R}^n} \rho_s(\mathbf{x}) d\mathbf{x} = s^n$. Hence,

$$\nu_s := \rho_s / s^n \tag{5}$$

is an n-dimensional probability density function and as before, we use ν to denote ν_1. 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.[3] Functions are extended to sets in the usual way; i.e.,

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

$\rho_s(A) = \sum_{\mathbf{x}\in A} \rho_s(\mathbf{x})$ 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 $\|\mathbf{x}\| \le t$ and $\|\mathbf{x} - \mathbf{y}\| \le l$,*

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

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

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

■

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

$$\forall \mathbf{x} \in A,\ D_{A,s}(\mathbf{x}) := \frac{\rho_s(\mathbf{x})}{\rho_s(A)}. \tag{6}$$

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),

$$\forall r \in [0, 1),\ \Psi_\beta(r) := \sum_{k=-\infty}^{\infty} \frac{1}{\beta} \cdot \exp\left(-\pi\Big(\frac{r-k}{\beta}\Big)^2\right). \tag{7}$$

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 $0 < \alpha < \beta \le 2\alpha$,*

$$\Delta(\Psi_\alpha, \Psi_\beta) \le 9\Big(\frac{\beta}{\alpha} - 1\Big).$$

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 $0 < \varepsilon \le 1$. Then the statistical distance that we wish to bound is given by

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

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

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

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 $\phi : \mathbb{T} \to \mathbb{R}^+$ and some integer $p \geq 1$ we define its discretization $\bar{\phi} : \mathbb{Z}_p \to \mathbb{R}^+$ as the discrete probability distribution obtained by sampling from ϕ, multiplying by p, and rounding to the closest integer modulo p. More formally,

$$\bar{\phi}(i) := \int_{(i-1/2)/p}^{(i+1/2)/p} \phi(x)dx. \tag{8}$$

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

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

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

Learning with errors. For an integer $p = p(n)$ and a distribution χ on $\mathbb{Z}_p$, we say that an algorithm solves $\mathrm{LWE}_{p,\chi}$ 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 $\mathrm{LWE}_{p,\phi}$ 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 [28]. A *lattice* in $\mathbb{R}^n$ is defined as the set of all integer combinations of n linearly independent vectors. This set of vectors is known as a *basis* of the lattice and is not unique. Given a basis $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ of a lattice L, the *fundamental parallelepiped* generated by this basis is defined as

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

When the choice of basis is clear, we write $\mathcal{P}(L)$ instead of $\mathcal{P}(\mathbf{v}_1, \ldots, \mathbf{v}_n)$. 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 absolute value of 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 $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ of a lattice, we define the dual basis as the set of vectors $(\mathbf{v}_1^*, \ldots, \mathbf{v}_n^*)$ such that $\langle \mathbf{v}_i, \mathbf{v}_j^* \rangle = \delta_{ij}$ for all $i, j \in [n]$ where δ_{ij} denotes the Kronecker delta, i.e., 1 if $i = j$ and 0 otherwise. With a slight abuse of notation, we sometimes write L for the $n \times n$ matrix whose columns are $\mathbf{v}_1, \ldots, \mathbf{v}_n$. With this notation, we notice that $L^* = (L^T)^{-1}$. From this it follows that $\det(L^*) = 1/\det(L)$. As another example of this notation, for a point $\mathbf{v} \in L$ we write $L^{-1}\mathbf{v}$ to indicate the 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 ([9], 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 mass given by the discrete Gaussian measure $D_{L,r}$ to points of norm greater than $\sqrt{n}r$ is at most exponentially small (the analogous statement for the continuous Gaussian ν_r is easy to establish).

Lemma 2.4 ([9], Lemma 1.4(i)) *For any lattice L and $a \geq 1$, $\rho_a(L) \leq a^n \rho(L)$.*

Lemma 2.5 ([9], Lemma 1.5(i)) *Let B_n denote the Euclidean unit ball. Then, for any lattice L and any $r > 0$, $\rho_r(L \setminus \sqrt{n}rB_n) < 2^{-2n} \cdot \rho_r(L)$, where $L \setminus \sqrt{n}rB_n$ is the set of lattice points of norm greater than $\sqrt{n}r$.*

In this paper we consider the following lattice problems. The first two, the decision version of the shortest vector problem (GAPSVP) 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.

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* $\text{GIVP}_\gamma^\varphi$ *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 [25]): For an n-dimensional lattice L, and some $d > 0$, we say that an algorithm solves $\mathrm{CVP}_{L,d}$ 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* [29]. Intuitively, this parameter provides the width beyond which the discrete Gaussian measure on a lattice behaves like a continuous one. 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 $\lim_{s\to 0} \rho_{1/s}(L^* \setminus \{\mathbf{0}\}) = \infty$ and $\lim_{s\to\infty} \rho_{1/s}(L^* \setminus \{\mathbf{0}\}) = 0$. So, the parameter $\eta_\varepsilon(L)$ is well defined for any $\varepsilon > 0$, and $\varepsilon \mapsto \eta_\varepsilon(L)$ is the inverse function of $s \mapsto \rho_{1/s}(L^* \setminus \{\mathbf{0}\})$. In particular, $\eta_\varepsilon(L)$ is also a continuous and strictly decreasing function of ε.

The motivation for this definition (and the name 'smoothing parameter') comes from the following result, shown in [29] (and included here as Claim 3.8). Informally, it says that if we choose a 'random' lattice point from an n-dimensional lattice L and add continuous Gaussian noise ν_s for some $s > \eta_\varepsilon(L)$ then the resulting distribution is within statistical distance ε of the 'uniform distribution on $\mathbb{R}^n$'. In this paper, we show (in Claim 3.9) 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 [29].

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,

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

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

$$\hat{h}(\mathbf{w}) = e^{2\pi i \langle \mathbf{v}, \mathbf{w} \rangle} \hat{g}(\mathbf{w}). \tag{9}$$

Similarly, if h is defined by $h(\mathbf{x}) = e^{2\pi i \langle \mathbf{x}, \mathbf{v} \rangle} g(\mathbf{x})$ for some function g and vector $\mathbf{v}$ then

$$\hat{h}(\mathbf{w}) = \hat{g}(\mathbf{w} - \mathbf{v}). \tag{10}$$

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

Lemma 2.14 (Poisson summation formula) *For any lattice L and any function $f : \mathbb{R}^n \to \mathbb{C}$,*

$$f(L) = \det(L^*) \hat{f}(L^*).$$

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 (Main theorem) *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* $\mathrm{LWE}_{p,\Psi_\alpha}$ *given a polynomial number of samples. Then there exists an efficient quantum algorithm for* $\mathrm{DGS}_{\sqrt{2n} \cdot \eta_\varepsilon(L)/\alpha}$.

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 the iterative step lemma, 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 the bootstrapping lemma, Lemma 3.2. Next, for $i = 3n, 3n-1, \ldots, 1$ 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, called the iterative step, 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. ■

3.1 Bootstrapping

Lemma 3.2 (Bootstrapping) *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 [24], 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 (The iterative step) *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 $\mathrm{LWE}_{p,\Psi_\alpha}$ given a polynomial number of samples. Then, there exists a constant $c > 0$ and an efficient quantum algorithm that, given any n-dimensional lattice L, a number $r > \sqrt{2}p\eta_\varepsilon(L)$, and n^c samples from $D_{L,r}$, 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 $\mathrm{CVP}_{L^*,\alpha p/(\sqrt{2}r)}$. The second part is shown in Lemma 3.14. There, we describe a quantum algorithm that, given an oracle that solves $\mathrm{CVP}_{L^*,\alpha p/(\sqrt{2}r)}$, outputs a sample from $D_{L,r\sqrt{n}/(\alpha p)}$. This is the only quantum component in this paper. We note that the condition in Lemma 3.14 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 (First part of iterative step) *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 $\mathrm{LWE}_{p,\Psi_\alpha}$ given a polynomial number of samples. Then, there exist a constant $c > 0$ and an efficient algorithm that, given any n-dimensional lattice L, a number $r > \sqrt{2}p\eta_\varepsilon(L)$, and n^c samples from $D_{L,r}$, solves $\mathrm{CVP}_{L^*,\alpha p/(\sqrt{2}r)}$.*

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 $\mathrm{CVP}^{(p)}_{L,d}$ 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 $\mathrm{CVP}_{L,d}$ to $\mathrm{CVP}^{(p)}_{L,d}$.

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

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

After n steps, we have a point $\mathbf{x}_{n+1}$ whose distance to the lattice is at most d/p^n. We now apply an algorithm for approximately solving the closest vector problem, such as Babai's nearest plane algorithm [8]. This yields a lattice point $L\mathbf{a}$ within distance $2^n \cdot d/p^n \leq d < \lambda_1(L)/2$ of $\mathbf{x}_{n+1}$. Hence, $L\mathbf{a}$ is the lattice point closest to $\mathbf{x}_{n+1}$ and we managed to recover $\mathbf{a}_{n+1} = \mathbf{a}$. Knowing $\mathbf{a}_{n+1}$ and $\mathbf{a}_n \bmod p$ (by using the oracle), we can now recover $\mathbf{a}_n = p\mathbf{a}_{n+1} + (\mathbf{a}_n \bmod p)$. Continuing this process, we can recover $\mathbf{a}_{n-1}, \mathbf{a}_{n-2}, \ldots, \mathbf{a}_1$. This completes the algorithm since $L\mathbf{a}_1$ is the closest point to $\mathbf{x}_1 = \mathbf{x}$. ■

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 $\mathrm{CVP}^{(p)}_{L^*,\alpha p/(\sqrt{2}r)}$. 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 3.11, 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 $\mathrm{CVP}^{(p)}_{L^*,\alpha p/(\sqrt{2}r)}$.

Lemma 3.6 (Verifying solutions of LWE) *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 $y_1, \ldots, y_n$ 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 expectation[4]

$$\tilde{z} := \operatorname*{Exp}_{y\sim\xi}[\cos(2\pi y)] = \int_0^1 \cos(2\pi y)\xi(y)dy = \mathrm{Re}\Big[\int_0^1 \exp(2\pi i y)\xi(y)dy\Big].$$

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

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

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. ■

Lemma 3.7 (Handling error Ψ_β for $\beta \leq \alpha$) *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 $\mathrm{LWE}_{p,\Psi_\alpha}$ 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.*

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 $A_{\mathbf{s},\Psi_\alpha}$ (or something sufficiently close to it). 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}. ■

For the analysis of our main procedure in Lemma 3.11, we will need to following claims regarding Gaussian measures on lattices. On first reading, the reader can just read the statements of Claim 3.8 and Corollary 3.10 and skip directly to Lemma 3.11. All claims show that in some sense, when working above the smoothing parameter, the discrete Gaussian measure behaves like the continuous Gaussian measure. We start with the following claim, showing that above the smoothing parameter, the discrete Gaussian measure is essentially invariant under shifts.

[4] We 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.

Claim 3.8 *For any lattice L, $\mathbf{c} \in \mathbb{R}^n$, $\varepsilon > 0$, and $r \geq \eta_\varepsilon(L)$,*

$$\rho_r(L + \mathbf{c}) \in r^n \det(L^*)(1 \pm \varepsilon).$$

Proof: Using the Poisson summation formula (Lemma 2.14) and the assumption that $\rho_{1/r}(L^* \setminus \{\mathbf{0}\}) \leq \varepsilon$,

$$\begin{aligned}
\rho_r(L + \mathbf{c}) = \sum_{\mathbf{x} \in L} \rho_r(\mathbf{x} + \mathbf{c}) &= \sum_{\mathbf{x} \in L} \rho_{r,-\mathbf{c}}(\mathbf{x}) \\
&= \det(L^*) \sum_{\mathbf{y} \in L^*} \widehat{\rho_{r,-\mathbf{c}}}(\mathbf{y}) \\
&= r^n \det(L^*) \sum_{\mathbf{y} \in L^*} \exp\left(2\pi i \langle \mathbf{c}, \mathbf{y} \rangle\right) \rho_{1/r}(\mathbf{y}) \\
&= r^n \det(L^*)(1 \pm \varepsilon).
\end{aligned}$$

■

The following claim (which is only used to establish the corollary following it) says that when adding a continuous Gaussian of width s to a discrete Gaussian of width r, with both r and s sufficiently greater than the smoothing parameter, the resulting distribution is very close to a continuous Gaussian of the width we would expect, namely $\sqrt{r^2 + s^2}$. To get some intuition on why we need to assume that *both* Gaussians are sufficiently wide, notice for instance that if the discrete Gaussian is very narrow, then it is concentrated on the origin, making the sum have width s. Also, if the continuous Gaussian is too narrow, then the discrete structure is still visible in the sum.

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

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

$$\begin{aligned}
Y(\mathbf{x}) &= \frac{1}{s^n \rho_r(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \rho_r(\mathbf{y}) \rho_s(\mathbf{x} - \mathbf{y}) \\
&= \frac{1}{s^n \rho_r(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \exp\left(-\pi(\|\mathbf{y}/r\|^2 + \|(\mathbf{x} - \mathbf{y})/s\|^2)\right) \\
&= \frac{1}{s^n \rho_r(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \exp\left(-\pi\left(\frac{r^2 + s^2}{r^2 \cdot s^2} \cdot \left\|\mathbf{y} - \frac{r^2}{r^2 + s^2}\mathbf{x}\right\|^2 + \frac{1}{r^2 + s^2}\|\mathbf{x}\|^2\right)\right) \\
&= \exp\left(-\frac{\pi}{r^2 + s^2}\|\mathbf{x}\|^2\right) \frac{1}{s^n \rho_r(L + \mathbf{u})} \sum_{\mathbf{y} \in L + \mathbf{u}} \exp\left(-\pi\left(\frac{r^2 + s^2}{r^2 \cdot s^2} \cdot \left\|\mathbf{y} - \frac{r^2}{r^2 + s^2}\mathbf{x}\right\|^2\right)\right) \\
&= \frac{1}{s^n} \rho_t(\mathbf{x}) \cdot \frac{\rho_{rs/t,(r/t)^2\mathbf{x}-\mathbf{u}}(L)}{\rho_{r,-\mathbf{u}}(L)} \\
&= \frac{1}{s^n} \rho_t(\mathbf{x}) \cdot \frac{\widehat{\rho_{rs/t,(r/t)^2\mathbf{x}-\mathbf{u}}}(L^*)}{\widehat{\rho_{r,-\mathbf{u}}}(L^*)} \\
&= \rho_t(\mathbf{x})/t^n \cdot \frac{(t/rs)^n \widehat{\rho_{rs/t,(r/t)^2\mathbf{x}-\mathbf{u}}}(L^*)}{(1/r)^n \widehat{\rho_{r,-\mathbf{u}}}(L^*)} \qquad (11)
\end{aligned}$$

where in the next-to-last equality we used Lemma 2.14. Using Eq. (9),

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

Hence,

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

where the last inequality follows from $1/r \le t/rs$. Hence, the quotient in (11) is between $(1-\varepsilon)/(1+\varepsilon) \ge 1-2\varepsilon$ and $(1+\varepsilon)/(1-\varepsilon) \le 1+4\varepsilon$. This implies that,

$$|Y(\mathbf{x}) - \rho_t(\mathbf{x})/t^n| \le \rho_t(\mathbf{x})/t^n\cdot 4\varepsilon.$$

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

Corollary 3.10 *Let L be a lattice, let $\mathbf{z},\mathbf{u}\in\mathbb{R}^n$ be vectors, and let $r,\alpha>0$ be two reals. Assume that $1/\sqrt{1/r^2+(\|\mathbf{z}\|/\alpha)^2} \ge \eta_\varepsilon(L)$ for some $\varepsilon<\frac{1}{2}$. Then the distribution of $\langle\mathbf{z},\mathbf{v}\rangle+e$ where $\mathbf{v}$ is distributed according to $D_{L+\mathbf{u},r}$ and e is a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, is within statistical distance 4ε of a normal variable with mean 0 and standard deviation $\sqrt{(r\|\mathbf{z}\|)^2+\alpha^2}/\sqrt{2\pi}$. In particular, since statistical distance cannot increase by applying a function, the distribution of $\langle\mathbf{z},\mathbf{v}\rangle + e \bmod 1$ is within statistical distance 4ε of $\Psi_{\sqrt{(r\|\mathbf{z}\|)^2+\alpha^2}}$.*

Proof: We first observe that the distribution of $\langle\mathbf{z},\mathbf{v}\rangle+e$ is exactly the same as that of $\langle\mathbf{z},\mathbf{v}+\mathbf{h}\rangle$ where $\mathbf{h}$ is distributed as the continuous Gaussian $\nu_{\alpha/\|\mathbf{z}\|}$. Next, by Claim 3.9, we know that the distribution of $\mathbf{v}+\mathbf{h}$ is within statistical distance 4ε of the continuous Gaussian $\nu_{\sqrt{r^2+(\alpha/\|\mathbf{z}\|)^2}}$. Taking the inner product of this continuous Gaussian with $\mathbf{z}$ leads to a normal distribution with mean 0 and standard deviation $\sqrt{(r\|\mathbf{z}\|)^2+\alpha^2}/\sqrt{2\pi}$, and we complete the proof by using the fact that statistical distance cannot increase by applying a function (inner product with $\mathbf{z}$ in this case). ■

Lemma 3.11 (Main procedure of the first part) *Let $\varepsilon=\varepsilon(n)$ be a negligible function, $p=p(n)\ge 2$ 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 $\beta\le\alpha$, 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 $\mathrm{CVP}^{(p)}_{L^*,\alpha p/(\sqrt{2}r)}$.*

Proof: We describe a procedure that given $\mathbf{x}$ within distance $\alpha p/(\sqrt{2}r)$ of L^*, outputs samples from the distribution $A_{\mathbf{s},\Psi_\beta}$ for some $\beta\le\alpha$ where $\mathbf{s}=(L^*)^{-1}\kappa_{L^*}(\mathbf{x}) \bmod p$. By running 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 $D_{L,r}$, and let $\mathbf{a}=L^{-1}\mathbf{v} \bmod p$. We then output

$$(\mathbf{a},\langle\mathbf{x},\mathbf{v}\rangle/p+e \bmod 1) \tag{12}$$

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 notice that the distribution of $\mathbf{a}$ is very close to uniform. Indeed, the probability of obtaining each $\mathbf{a} \in \mathbb{Z}_p^n$ is proportional to $\rho_r(pL + L\mathbf{a})$. Using $\eta_\varepsilon(pL) = p\eta_\varepsilon(L) < r$ and Claim 3.8, the latter is $(r/p)^n \det(L^*)(1 \pm \varepsilon)$, which implies that the statistical distance between the distribution of $\mathbf{a}$ and the uniform distribution is negligible.

Next, we condition on any fixed value of $\mathbf{a}$ and consider the distribution of the second element in (12). Define $\mathbf{x}' = \mathbf{x} - \kappa_{L^*}(\mathbf{x})$ and note that $\|\mathbf{x}'\| \leq \alpha p/(\sqrt{2}r)$. Then,

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

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 $\langle \mathbf{s}, \mathbf{a} \rangle / p \bmod 1$.

We complete the proof by applying Corollary 3.10, which shows that the distribution of the remaining part $\langle \mathbf{x}'/p, \mathbf{v} \rangle + e$ is within negligible statistical distance of Ψ_β for $\beta = \sqrt{(r\|\mathbf{x}'\|/p)^2 + \alpha^2/2} \leq \alpha$, as required. Here we used that the distribution of $\mathbf{v}$ is $D_{pL+L\mathbf{a},r}$ (since we are conditioning on $\mathbf{a}$), the distribution of e is normal with mean 0 and standard deviation $(\alpha/\sqrt{2})/\sqrt{2\pi}$, and that

$$1/\sqrt{1/r^2 + (\sqrt{2}\|\mathbf{x}'\|/p\alpha)^2} \geq r/\sqrt{2} > \eta_\varepsilon(pL).$$

■

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., [31] 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 width 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 3.12 *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*

$$\sum_{\mathbf{x} \in L} \sqrt{\rho_r(\mathbf{x})} |\mathbf{x}\rangle = \sum_{\mathbf{x} \in L} \rho_{\sqrt{2}r}(\mathbf{x}) |\mathbf{x}\rangle. \tag{13}$$

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

$$\sum_{x=-\sqrt{n}r}^{\sqrt{n}r} e^{-\pi(x/(\sqrt{2}r))^2}|x\rangle. \tag{14}$$

This state can be created efficiently using a technique by Grover and Rudolph [18] who show that in order to create such a state, it suffices to be able to compute for any $a, b \in \{-\sqrt{n}r, \ldots, \sqrt{n}r\}$ 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. (14), which can be written as

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

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

$$\sum_{\mathbf{x}\in\mathbb{Z}^n} \rho_{\sqrt{2}r}(\mathbf{x})|\mathbf{x}\rangle \tag{15}$$

and hence for our purposes we can assume that we have generated the state in Eq. (15).

Next, using the LLL basis reduction algorithm [24], 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 $\|\mathbf{y}\| \leq \mathrm{diam}(\mathcal{P}(L)) \leq n2^n\lambda_n(L)$. The state we obtain after the measurement is

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

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

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

Our goal is to show that this state is within ℓ_2 distance $2^{-\Omega(n)}$ of the one in Eq. (13). First, by Lemma 2.5, all but an exponentially small part of the ℓ_2 norm of the state in Eq. (13) 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. (13) 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 3.13 *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*

$$|\vartheta_1\rangle = \sum_{\mathbf{x}\in L/R, \|\mathbf{x}\|<\sqrt{n}} \rho(\mathbf{x})|\mathbf{x} \bmod \mathcal{P}(L)\rangle, \qquad \textit{and}$$
$$|\vartheta_2\rangle = \sum_{\mathbf{x}\in L/R} \rho(\mathbf{x})|\mathbf{x} \bmod \mathcal{P}(L)\rangle = \sum_{\mathbf{x}\in L/R\cap\mathcal{P}(L)} \sum_{\mathbf{y}\in L} \rho(\mathbf{x}-\mathbf{y})|\mathbf{x}\rangle$$

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,

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

■

We now prove the main lemma of this subsection.

Lemma 3.14 (Second part of iterative step) *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 $\mathrm{CVP}_{L^*,d}$, 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

$$\sum_{\mathbf{x}\in L^*/R\cap\mathcal{P}(L^*)} \sum_{\mathbf{y}\in L^*} \rho(\mathbf{x}-\mathbf{y})|\mathbf{x}\rangle. \tag{16}$$

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 3.12 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 3.13, this state is exponentially close to the required state (16).

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 (16) 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

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

where the last equality follows from Lemma 2.14 and Eq. (10). Hence, the resulting state can be equivalently written as

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

Notice that $\lambda_1(RL) = R\lambda_1(L) \geq R/\lambda_n(L^*) \geq 2^{3n}$. Hence, we can apply Claim 3.13 to the lattice RL, and obtain that this state is exponentially close to

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

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 [8]. 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 [29]. We include the proof since only a proof sketch was given there.

Lemma 3.15 *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 $(1, 0, \ldots, 0)$ is orthogonal to H. Using Lemma 2.14,

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

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

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

Clearly, if none of the B_i's happens, then $\dim \operatorname{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{n^2}) = n$. 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 $\mathbf{x}_1, \ldots, \mathbf{x}_{(i-1)n}$ such that $\dim \operatorname{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{(i-1)n}) < n$. By Lemma 3.15, the probability that

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

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

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 $\mathrm{GIVP}_{2\sqrt{2}n\eta_\varepsilon(L)/\alpha}$. By Lemma 2.12, this algorithm also solves $\mathrm{SIVP}_{\tilde{O}(n/\alpha)}$.

Lemma 3.17 *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* $\mathrm{GIVP}_{2\sqrt{n}\varphi}$ *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 parameter 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 [24] 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 $i \in \{0, \ldots, 2n\}$ 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 $S, S_0, S_1, \ldots, S_{2n}$, 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 $i \in \{0, \ldots, 2n\}$ be such that $\varphi(L) < r_i \leq 2\varphi(L)$. Such an i must exist by Claim 2.13. By Corollary 3.16, 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. ■

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 3.18 *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,* $\mathrm{dist}(\mathbf{t}, L) \leq d$*, whereas in* NO *instances,* $\mathrm{dist}(\mathbf{t}, L) > \gamma(n) \cdot d$.

Definition 3.19 *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,* $\mathrm{dist}(\mathbf{t}, L) \leq d$*. In* NO *instances,* $\lambda_1(L) > \gamma(n) \cdot d$ *and* $\mathrm{dist}(\mathbf{t}, L) > \gamma(n) \cdot d$.

In [17] it is shown that for any $\gamma = \gamma(n) \geq 1$, there is a polynomial time reduction from GapSVP_γ to GapCVP'_γ (see also Lemma 5.22 in [29]). 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 $\text{GapSVP}_{O(n/\alpha)}$).

Lemma 3.20 *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 $\mathbf{w}_1, \ldots, \mathbf{w}_N$ 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 $\mathbf{w}_1, \ldots, \mathbf{w}_N$ 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 $\mathbf{w}_1, \ldots, \mathbf{w}_N \in L^*$. We then apply $\mathcal{V}$ with L, $\mathbf{t}$, d, and the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_N$. 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 $\mathbf{w}_1, \ldots, \mathbf{w}_N$ 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 $f_\mathbf{t} : \mathbb{Z}_p^n \times \mathbb{Z}_p \to \mathbb{Z}_p^n \times \mathbb{Z}_p$ defined by

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

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

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

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

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

Lemma 4.3 (Discrete to Continuous) *Let $n, p \geq 1$ 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 $\mathrm{LWE}_{p,\bar{\phi}}$. Then, there exists an efficient algorithm W' that solves $\mathrm{LWE}_{p,\phi}$.*

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

By combining the three lemmas above, we obtain

Lemma 4.4 *Let $n \geq 1$ be an integer and $2 \leq p \leq \mathrm{poly}(n)$ 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 $A_{\mathbf{s},\bar{\phi}}$ from U for a non-negligible fraction of all possible $\mathbf{s}$. Then, there exists an efficient algorithm that solves $\mathrm{LWE}_{p,\phi}$.*

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 $\lim_{n\to\infty} \alpha(n) \cdot \sqrt{n}\log n = 0$. 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 $i = 1, \ldots, m$, choose m vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{Z}_p^n$ independently from the uniform distribution. Also choose elements $e_1, \ldots, e_m \in \mathbb{Z}_p$ independently according to χ. The public key is given by $(\mathbf{a}_i, b_i)_{i=1}^m$ where $b_i = \langle \mathbf{a}_i, \mathbf{s}\rangle + e_i$.
- **Encryption:** In order to encrypt a bit we choose a random 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 $\mathbf{a}_1, \ldots, \mathbf{a}_m$. 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 $b_1, \ldots, b_m$. 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 $\chi^{\star k}$ as the distribution obtained by summing together k independent samples from χ, where addition is performed in $\mathbb{Z}_p$ (for $k = 0$ we define $\chi^{\star 0}$ as the distribution that is constantly 0). For a probability distribution ϕ on $\mathbb{T}$ we define $\phi^{\star k}$ similarly. For an element $a \in \mathbb{Z}_p$ we define $|a|$ as the integer a if $a \in \{0, 1, \ldots, \lfloor \frac{p}{2} \rfloor\}$ and as the integer $p - a$ otherwise. In other words, $|a|$ represents the distance of a from 0. Similarly, for $x \in \mathbb{T}$, we define $|x|$ as x for $x \in [0, \frac{1}{2}]$ and as $1 - x$ otherwise.

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

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

Then, the probability of decryption error is at most δ. 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 $\chi^{\star|S|}$. According to our assumption, $|\sum_{i\in S} e_i|$ is less than $\lfloor \frac{p}{2} \rfloor/2$ with probability at least $1 - \delta$. In this case, it is closer to 0 than to $\lfloor \frac{p}{2} \rfloor$ and therefore the decryption is correct. The proof for an encryption of 1 is similar. ■

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

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

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

Proof: A sample from $\bar{\Psi}_\alpha^{\star k}$ can be obtained by sampling $x_1, \ldots, x_k$ from Ψ_α and outputting $\sum_{i=1}^k \lfloor p x_i \rceil \bmod p$. Notice that this value is at most $k \leq m < p/32$ away from $\sum_{i=1}^k p x_i \bmod p$. Hence, it is enough to show that $|\sum_{i=1}^k p x_i \bmod p| < p/16$ with high probability. This condition is equivalent to the condition that $|\sum_{i=1}^k x_i \bmod 1| < 1/16$. 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 special case of the leftover hash lemma that appears in [19]. We include a proof for completeness.

Claim 5.3 *Let G be some finite Abelian group and let l be some integer. For any l elements $g_1, \ldots, g_l \in G$ consider the statistical distance between the uniform distribution on G and the distribution given by the sum of a random subset of $g_1, \ldots, g_l$. Then the expectation of this statistical distance over a uniform choice of $g_1, \ldots, g_l \in G$ 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 $\mathbf{g} = (g_1, \ldots, g_l)$ of l elements from G, let $P_{\mathbf{g}}$ be the distribution of the sum of a random subsets of $g_1, \ldots, g_l$, i.e.,

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

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. The ℓ_2 norm of $P_{\mathbf{g}}$ is given by

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

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

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

Finally, the expected distance from the uniform distribution is

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

■

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 $A_{\mathbf{s},\chi}$ and U for a non-negligible fraction of all $\mathbf{s}$. If $\chi = \bar{\Psi}_\alpha$ and $p \leq \mathrm{poly}(n)$ is a prime, then by Lemma 4.4, this also implies an efficient (classical) algorithm that solves $\mathrm{LWE}_{p,\Psi_\alpha}$. This in turn implies, by Theorem 3.1, an efficient quantum algorithm for $\mathrm{DGS}_{\sqrt{2n}\cdot\eta_\varepsilon(L)/\alpha}$. Finally, by Lemma 3.17 we also obtain an efficient quantum algorithm for $\mathrm{SIVP}_{\tilde{O}(n/\alpha)}$ and by Lemma 3.20 we obtain an efficient quantum algorithm for $\mathrm{GAPSVP}_{O(n/\alpha)}$.

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 $A_{\mathbf{s},\chi}$ and U for a non-negligible fraction of all possible $\mathbf{s}$.*

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

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

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 $|\operatorname{Exp}_{\mathbf{s}}[p_0(\mathbf{s})] - \operatorname{Exp}_{\mathbf{s}}[p_u(\mathbf{s})]| \geq \frac{1}{2n^c}$. Define

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

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

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

with probability at least $\frac{1}{8n^c}$ over the choice of $(\mathbf{a}_i, b_i)_{i=1}^m$ from $A_{\mathbf{s},\chi}$. 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, Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, Miklos Santha, Madhu Sudan, and an anonymous referee 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] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Proc. of 6th IACR Theory of Cryptography Conference (TCC)*, pages 474–495, 2009.

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

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

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

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

[11] 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.

[12] 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.

[13] D. Cash, C. Peikert, and A. Sahai. Efficient circular-secure encryption from hard learning problems, 2009. Submitted.

[14] 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.

[15] 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.

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

[17] 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.

[18] L. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. In *quant-ph/0208112, http://xxx.lanl.gov*, 2002.

[19] 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.

[20] J. Katz and Y. Lindell. *Introduction to modern cryptography*. Chapman & Hall/CRC Cryptography and Network Security. Chapman & Hall/CRC, Boca Raton, FL, 2008.

[21] 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.

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

[23] 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.

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

[25] 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.

[26] V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem, 2009. Manuscript.

[27] 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.

[28] 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.

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

[30] 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.

[31] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[32] C. Peikert. Limits on the hardness of lattice problems in l_p norms. *Comput. Complexity*, 17(2):300–351, 2008. Preliminary version in CCC'07.

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

[34] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Advances in cryptology—CRYPTO '08*, volume 5157 of *Lecture Notes in Comput. Sci.*, pages 554–571. Springer, Berlin, 2008.

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

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

[37] 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.

[38] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoret. Comput. Sci.*, 53(2-3):201–224, 1987.