On the Complexity of Lattice Problems with Polynomial Approximation Factors

Oded Regev *

May 21, 2007

Abstract

Lattice problems are known to be hard to approximate to within sub-polynomial factors. For larger approximation factors, such as $\sqrt{n}$, lattice problems are known to be in complexity classes such as $\mathsf{NP} \cap \mathsf{coNP}$ and are hence unlikely to be NP-hard. Here we survey known results in this area. We also discuss some related zero-knowledge protocols for lattice problems.

1 Introduction

A *lattice* is the set of all integer combinations of n linearly independent vectors $v_1, \ldots, v_n$ in $\mathbb{R}^n$. These vectors are known as a *basis* of the lattice. Lattices have been investigated by mathematicians for decades, and have recently also attracted considerable attention in the computer science community following the discovery of the LLL algorithm by Lenstra, Lenstra, and Lovász [19]. Many different problems can be phrased as questions about lattices, such as integer programming [15], factoring polynomials with rational coefficients [19], integer relation finding [13], integer factoring, and Diophantine approximation [29]. More recently, the study of lattices attracted renewed attention due to the fact that lattice problems were shown by Ajtai [3] to possess a particularly desirable property for cryptography: worst-case to average-case reducibility.

Lattice problems, such as the shortest vector problem (SVP) and the closest vector problem (CVP), are fascinating from a computational complexity point of view (see Figure 1). On one hand, by the LLL algorithm [19] and subsequent improvements [28], we are able to efficiently approximate lattice problems to within essentially exponential factors, namely $2^{n(\log\log n)^2/\log n}$ where n is the dimension of the lattice. In fact, if we allow randomization, the approximation factor improves slightly to $2^{n\log\log n/\log n}$ [5]. On the other hand, we know that for some $c > 0$, no efficient algorithm can approximate lattice problems to within $n^{c/\log\log n}$ unless $\mathsf{P} = \mathsf{NP}$ or another unlikely event occurs. This was established in a long sequence of works, including [31, 2, 7, 9, 20, 16, 14]. See also Khot's chapter [17] in these proceedings.

Considering the above results, one immediate question arises: what can we say about approximation factors in between these two extremes? There is a very wide gap between the approximation

*Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Supported by the Binational Science Foundation, by the Israel Science Foundation, and by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848.

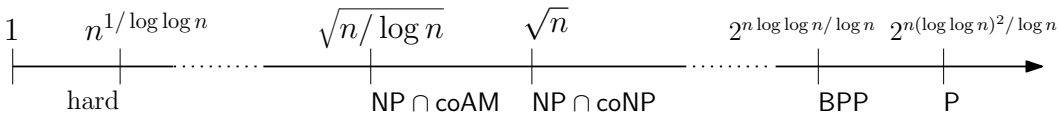

Figure 1: The complexity of lattice problems (some constants omitted)

factor achieved by the best known algorithm ($2^{n\log\log n/\log n}$), and the best known hardness result ($n^{c/\log\log n}$). Of particular importance is the range of polynomial approximation factors. The reason for this is that the security of lattice-based cryptographic constructions following Ajtai's seminal work [3] is based on the worst-case hardness of approximating lattice problems in this region (see also [4, 22, 27] and Micciancio's chapter [21] in these proceedings). If, for instance, we could prove that approximating lattice problems to within $O(n^2)$ is NP-hard then this would have the tremendous implication of a public key cryptosystem whose security is based solely on the $\mathsf{P} \neq \mathsf{NP}$ conjecture.

This scenario, however, is unlikely to happen. There are several results indicating that approximating lattice problems to within polynomial factors is unlikely to be NP-hard. These results are sometimes known as 'limits on inapproximability'. They are established by showing containment in complexity classes such as $\mathsf{NP} \cap \mathsf{coNP}$. As is well known, if a problem in $\mathsf{NP} \cap \mathsf{coNP}$ is NP-hard, then $\mathsf{NP} = \mathsf{coNP}$ and the polynomial hierarchy collapses. For lattice problems this is true even under Cook-reductions, as we show in Appendix A.

To state these results precisely, let us first recall the promise problems associated with the shortest vector problem and the closest vector problem. Below we use $\mathcal{L}(B)$ to denote the lattice generated by the basis B. Moreover, all distances and lengths in this survey are with respect to the ℓ_2 norm (but see [26] for an interesting extension of the results described here to other ℓ_p norms).

Definition 1.1 GapCVP_γ

YES *instances:* *triples* (B, v, d) *such that* $\mathrm{dist}(v, \mathcal{L}(B)) \leq d$

NO *instances:* *triples* (B, v, d) *such that* $\mathrm{dist}(v, \mathcal{L}(B)) > \gamma d$

where B is a basis for a lattice in $\mathbb{Q}^n$, $v \in \mathbb{Q}^n$ is a vector, and $d \in \mathbb{Q}$ is some number.

Definition 1.2 GapSVP_γ

YES *instances:* *pairs* (B, d) *such that* $\lambda_1(\mathcal{L}(B)) \leq d$

NO *instances:* *pairs* (B, d) *such that* $\lambda_1(\mathcal{L}(B)) > \gamma d$

where B is a basis for a lattice in $\mathbb{Q}^n$, $d \in \mathbb{Q}$ is some number, and λ_1 denotes the length of the shortest nonzero vector in a lattice.

Note that in both cases setting d to some fixed value (say 1) leads to an essentially equivalent definition (as one can easily rescale the input).

The oldest result showing a limit on the inapproximability of lattice problems is by Lagarias, Lenstra and Schnorr [18], who showed that $\mathsf{GapCVP}_{n^{1.5}}$ is in $\mathsf{NP} \cap \mathsf{coNP}$. As we mentioned above, this shows that $\mathsf{GapCVP}_{n^{1.5}}$ is highly unlikely to be NP-hard. Let us remark at the outset that showing containment in NP is trivial: a witness for $\mathrm{dist}(v, \mathcal{L}(B)) \leq d$ is simply a vector $u \in \mathcal{L}(B)$ such that $\|v - u\| \leq d$. The more interesting part is providing a witness for the fact that a point is *far* from the lattice. Some thought reveals that this is no longer a trivial task: there is a huge number of lattice vectors that can potentially be very close to v. The way containment in coNP is

usually shown is by utilizing properties of the *dual lattice*. Let us also mention that although we state this result and the results below only for GapCVP, they all hold also for GapSVP. This follows from a simple approximation preserving reduction from GapSVP to GapCVP [12], which we include for completeness in Appendix B.

An improvement of the Lagarias et al. result was obtained by Banaszczyk [6] who showed that GapCVP_n is in $\mathsf{NP} \cap \mathsf{coNP}$. This was recently further improved by Aharonov and Regev [1] to $\mathsf{GapCVP}_{\sqrt{n}}$.

Theorem 1.3 ([1]) *There exists $c > 0$ such that $\mathsf{GapCVP}_{c\sqrt{n}}$ is in $\mathsf{NP} \cap \mathsf{coNP}$.*

In their coNP proof, the witness simply consists of a list of short vectors in the dual lattice. The verifier then uses these vectors to determine the distance of the target vector v from the lattice. A sketch of this proof appears in Section 3.

Another 'limits on inapproximability' result is by Goldreich and Goldwasser [11], who showed that $\mathsf{GapCVP}_{\sqrt{n/\log n}}$ is in $\mathsf{NP} \cap \mathsf{coAM}$ (where containment in coAM means that the complement of the problem is in the class AM defined in Definition 2.1).

Theorem 1.4 ([11]) *For any $c > 0$, $\mathsf{GapCVP}_{c\sqrt{n/\log n}}$ is in $\mathsf{NP} \cap \mathsf{coAM}$.*

We present a proof of this theorem in Section 2. The proof uses an elegant protocol in which an all-powerful prover convinces a computationally limited verifier that a point v is far from the lattice. We note that their result is incomparable with that of [1] since it involves a slightly harder problem ($\mathsf{GapCVP}_{\sqrt{n/\log n}}$) but shows containment in a somewhat wider class (coAM). It is an interesting open question whether containment in $\mathsf{NP} \cap \mathsf{coNP}$ holds also for gaps between $\sqrt{n/\log n}$ and $\sqrt{n}$.

In Section 4 we will discuss the topic of *zero-knowledge protocols*. We will observe that the Goldreich-Goldwasser protocol is zero-knowledge (against honest verifiers). We will then describe two zero-knowledge protocols with efficient provers, one for coGapCVP and one for GapCVP.

We can summarize our current state of knowledge by saying that for approximation factors beyond $\sqrt{n/\log n}$, lattice problems are unlikely to be NP-hard. This naturally brings us to one of the most important questions regarding the complexity of lattice problems: is there an efficient algorithm for approximating lattice problem to within polynomial factors? Given how difficult it is to come up with algorithms that perform even slightly better than the exponential factor achieved by the LLL algorithm, many people conjecture that the answer is negative. This conjecture lies at the heart of latticed-based cryptographic constructions such as Ajtai's [3], and is therefore of central importance. How can we hope to show such hardness if we do not believe the problem is NP-hard? One promising direction is by relating lattice problems to other problems that are believed to be hard. For instance, a reduction from factoring to, say, GapSVP_{n^2} would give a strong evidence to the conjecture, and would also establish the remarkable fact that lattice-based cryptosystems are at least as secure as factoring based cryptosystems.

Outline: In Section 2 we present a proof of Theorem 1.4, including some of the technical details that go into making the proof completely rigorous. These technical details, especially how to work with periodic distributions, appear in many other lattice-related results, and are therefore discussed in detail. Then, in Section 3 we present a sketch of the proof of Theorem 1.3. This sketch contains all the important ideas of the proof, but proofs of technical claims are omitted. The two sections

are independent. Then, in Section 4 we discuss zero-knowledge proof systems for lattice problems, and in particular sketch the prover-efficient zero-knowledge protocol of Micciancio and Vadhan [23]. This section requires a basic understanding of Section 2. Finally, in Appendix A we show in what sense the two theorems above imply 'limits on inapproximability', and in Appendix B we show how to extend our results to GapSVP.

2 The Goldreich-Goldwasser Protocol

In this section we prove Theorem 1.4. For simplicity, we will show that $\mathsf{GapCVP}_{\sqrt{n}} \in \mathsf{coAM}$. A slightly more careful analysis of the same protocol yields a gap of $c\sqrt{n/\log n}$ for any constant $c > 0$. First, let us define the class AM.

Definition 2.1 *A promise problem is in* AM *if there exists a protocol with a constant number of rounds between a* BPP *machine Arthur and a computationally unbounded machine Merlin, and two constants* $0 \leq a < b \leq 1$ *such that*

- **Completeness***: for any* YES *input, there exists a strategy for Merlin such that Arthur accepts with probability at least b, and*
- **Soundness***: for any* NO *input, and any strategy for Merlin, Arthur accepts with probability at most a.*

In order to prove Theorem 1.4, we present a protocol that allows Arthur to verify that a point is far from the lattice. Specifically, given (B, v, d), Arthur accepts with probability 1 if $\mathrm{dist}(v, \mathcal{L}(B)) > \sqrt{n}d$ and rejects with some positive probability if $\mathrm{dist}(v, \mathcal{L}(B)) \leq d$.

Informally, the protocol is as follows. Arthur first flips a fair coin. If it comes up heads, he randomly chooses a 'uniform' point in the lattice $\mathcal{L}(B)$; if it comes up tails, he randomly chooses a 'uniform' point in the shifted lattice $v + \mathcal{L}(B)$. Let w denote the resulting point. Arthur randomly chooses a uniform point x from the ball of radius $\frac{1}{2}\sqrt{n}d$ around w and then sends x to Merlin. Merlin is supposed to tell Arthur if the coin came up heads or not.

The correctness of this protocol follows from the following two observations (see Figure 2). If $\mathrm{dist}(v, \mathcal{L}(B)) > \sqrt{n}d$ then the two distributions are disjoint and then Merlin can answer correctly with probability 1. On the other hand, if $\mathrm{dist}(v, \mathcal{L}(B)) \leq d$, then the overlap between the two distributions is large and Merlin must make a mistake with some positive probability.

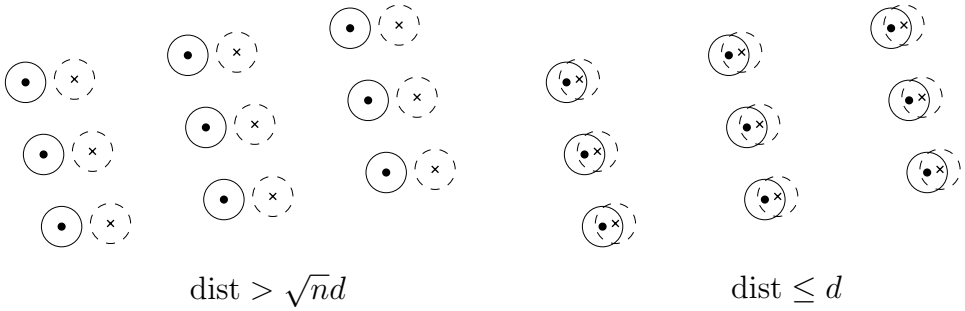

Figure 2: The two distributions

This informal description hides two technical problems. First, we cannot really work with the point x since it is chosen from a continuous distribution (and hence cannot be represented precisely in any finite number of bits). This is easy to take care of by working with an approximation of x with some polynomial number of bits. Another technical issue is the choice of a 'random' point from $\mathcal{L}(B)$. This is an infinite set and there is no uniform distribution on it. One possible solution is to take the uniform distribution on points in the intersection of $\mathcal{L}(B)$ with, say, some very large hypercube. This indeed solves the problem, but introduces some unnecessary complications to the proof since one needs to argue that the probability to fall close to the boundary of the hypercube is low. The solution we choose here is different and avoids this problem altogether by working with distributions on the basic parallelepiped of the lattice. We describe this solution in Subsection 2.3.

In the next few subsections, we present the necessary preliminaries for the proof.

2.1 Statistical Distance

Definition 2.2 *The* statistical distance *between two distributions X, Y on some set Ω is defined as*

$$\Delta(X,Y) = \max_{A \subseteq \Omega} |\Pr(X \in A) - \Pr(Y \in A)|.$$

One useful special case of this definition is the case where X and Y are discrete distributions over some countable set Ω. In this case, we have

$$\Delta(X,Y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr(X = \omega) - \Pr(Y = \omega)|.$$

Another useful special case is when X and Y are distributions on $\mathbb{R}^n$ with density functions f, g. In this case, we have

$$\Delta(X,Y) = \frac{1}{2} \int_{\mathbb{R}^n} |f(x) - g(x)| \, \mathrm{d}x.$$

For any distributions X, Y, $\Delta(X,Y)$ obtains values between 0 and 1. It is 0 if and only if X and Y are identical and 1 if and only if they are disjoint. It is helpful to consider the following interpretation of statistical distance. Assume we are given a sample that is taken from X with probability $\frac{1}{2}$ or from Y with probability $\frac{1}{2}$. Our goal is to decide which distribution the sample comes from. Then, it can be seen that our best strategy succeeds with probability $\frac{1}{2} + \frac{1}{2}\Delta(X,Y)$.

One important fact concerning the statistical distance is that it cannot increase by the application of a possibly randomized function. In symbols, $\Delta(f(X), f(Y)) \leq \Delta(X,Y)$ for any (possibly randomized) function f. This fact follows easily from the above interpretation of Δ.

2.2 Balls in n-dimensional Space

Let $\mathbf{B}(v,r)$ denote a ball of radius r around v. It is known that the volume of the unit ball $\mathbf{B}(0,1)$ in n dimensions is

$$V_n \stackrel{def}{=} \frac{\pi^{n/2}}{(n/2)!},$$

where we define $n! = n(n-1)!$ for $n \geq 1$ and $\frac{1}{2}! = \frac{1}{2}\sqrt{\pi}$. It can be shown that

$$\frac{(n+\frac{1}{2})!}{n!} \approx \frac{n!}{(n-\frac{1}{2})!} \approx \sqrt{n}.$$

Lemma 2.3 *For any $\varepsilon > 0$ and any vector v of length $\|v\| \leq \varepsilon$, the relative volume of the intersection of two unit balls whose centers are separated by v satisfies*

$$\frac{\mathrm{vol}(\mathbf{B}(0,1) \cap \mathbf{B}(v,1))}{\mathrm{vol}(\mathbf{B}(0,1))} \geq \varepsilon \frac{(1-\varepsilon^2)^{\frac{n-1}{2}}}{3} \sqrt{n}$$

Proof: It suffices to consider the case $\|v\| = \varepsilon$. As shown in Figure 3, the intersection contains a cylinder of height ε and radius $\sqrt{1-\varepsilon^2}$ centered around $v/2$. Hence, the volume of the intersection satisfies:

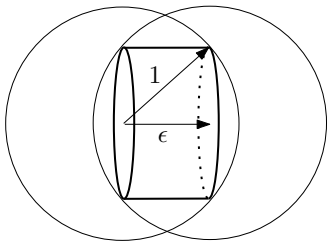

Figure 3: A cylinder in the intersection of two balls

$$\frac{\mathrm{vol}(\mathbf{B}(0,1) \cap \mathbf{B}(v,1))}{\mathrm{vol}(\mathbf{B}(0,1))} > \frac{\varepsilon V_{n-1}(\sqrt{1-\varepsilon^2})^{n-1}}{V_n} = \varepsilon(1-\varepsilon^2)^{\frac{n-1}{2}} \frac{\pi^{\frac{n-1}{2}}/(\frac{n-1}{2})!}{\pi^{\frac{n}{2}}/(\frac{n}{2})!} \approx \varepsilon(1-\varepsilon^2)^{\frac{n-1}{2}} \frac{\sqrt{n/2}}{\sqrt{\pi}}.$$

Notice that for $\varepsilon = \frac{2}{\sqrt{n}}$, the right hand side of the expression in Lemma 2.3 is bounded from below by some positive constant independent of n. This yields the following corollary.

Corollary 2.4 *There exists a constant $\delta > 0$ such that for any $d > 0$ and any $y \in \mathbb{R}^n$ such that $\|y\| \leq d$,*

$$\Delta\left(U(\mathbf{B}(0, \tfrac{1}{2}\sqrt{n}d)),\ U(\mathbf{B}(y, \tfrac{1}{2}\sqrt{n}d))\right) < 1 - \delta,$$

where $U(\cdot)$ denotes the uniform distribution on a set.

Proof: This statistical distance is exactly the volume of the symmetric difference of two balls divided by the sum of their volumes. According to the above lemma, this is bounded away from 1.

Remark: When $\varepsilon = c\sqrt{\log n/n}$ for some $c > 0$, the right hand side of the expression in Lemma 2.3 is still greater than some $1/\mathrm{poly}(n)$. Using this, one can obtain the improved result $\mathsf{GapCVP}_{c\sqrt{n/\log n}} \in \mathsf{coAM}$.

2.3 Working with Periodic Distributions

In the informal description above, we talked about the 'uniform distribution' on the lattice. This is clearly not defined. One possible solution is to restrict our attention to some large enough cube

$[-K, K]^n$. While possible, this solution introduces some technical annoyances as one has to argue that the probability to fall too close to the boundary of the cube (where the protocol might behave badly) is small.

Instead, our solution will be to work with only one period of the distribution. To demonstrate this approach, let us first consider the one-dimensional case. Assume we want to represent the distribution intuitively described as follows: choose a random point from the lattice $3\mathbb{Z}$ and add to it a number chosen uniformly from $[-0.1, 0.1]$. The first solution above would require us to take some large segment, say, $[-1000, 1000]$, and to restrict our distribution to it. Instead, we take one period of the distribution, say the segment $[0, 3]$, and consider the distribution on it. Hence, we obtain the uniform distribution on $[0, 0.1] \cup [2.9, 3]$. Notice that we could take another period, say the segment $[-3, 0]$, and work with it instead. Crucially, the transformation from one representation to another can be performed efficiently (by subtracting or adding 3 as needed).

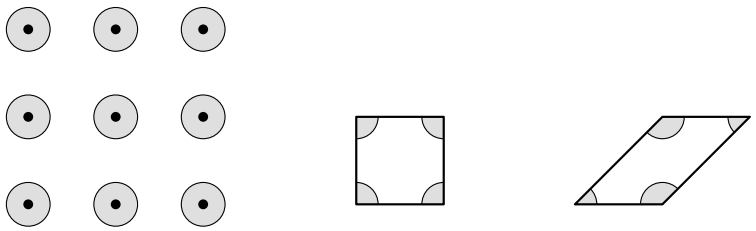

Figure 4: A periodic distribution on $\mathbb{Z}^2$ (left), restricted $\mathcal{P}((0,1),(1,0))$ (center) and to $\mathcal{P}((0,1),(1,1))$ (right).

A similar idea works for higher dimensions (see Figure 4). If we want to represent a periodic distribution on a lattice, we consider it as a distribution on some period of the lattice. A common choice is to take a *basic parallelepiped* of the lattice, defined as

$$\mathcal{P}(B) = \mathcal{P}(v_1, \ldots, v_n) = \left\{ \sum_{i=1}^{n} x_i v_i \;\middle|\; x_i \in [0, 1) \right\},$$

where $B = (v_1, \ldots, v_n)$ is some basis of the lattice. As before, we have several possible representations, depending on the choice of basis B. The transformation from a representation using B_1 to one using B_2 can be done efficiently by reducing points modulo $\mathcal{P}(B_2)$ (see Definition 2.5 below). Mathematically speaking, the objects we work with are distributions on the quotient $\mathbb{R}^n/\mathcal{L}(B)$, and $\mathcal{P}(B)$ is its set of representatives.

We emphasize that it is much easier to imagine 'periodic distributions' on $\mathbb{R}^n$. However, technically, it is much easier to work with distributions on $\mathcal{P}(B)$.

2.4 The Protocol

We will now show using Protocol 1 that $\mathsf{GapCVP}_{\sqrt{n}} \in \mathsf{coAM}$. The protocol uses the following definition.

Definition 2.5 *For $x \in \mathbb{R}^n$, $x \bmod \mathcal{P}(B)$ is the unique $y \in \mathcal{P}(B)$ satisfying $x - y \in \mathcal{L}(B)$.*

Remark: For simplicity, we ignore issues of finite precision; these can be dealt with by standard techniques. One issue that we do want to address is how to choose a point from the ball $\mathbf{B}(0, R)$

Protocol 1 The Goldreich-Goldwasser AM protocol

1. Arthur selects $\sigma \in \{0,1\}$ uniformly and a random point t in the ball $\mathbf{B}(0, \frac{1}{2}\sqrt{n}d)$. He then sends $x = (\sigma v + t) \bmod \mathcal{P}(B)$ to Merlin.
2. Merlin checks if $\mathrm{dist}(x, \mathcal{L}(B)) < \mathrm{dist}(x, v + \mathcal{L}(B))$. If so, he responds with $\tau = 0$; otherwise, he responds with $\tau = 1$.
3. Arthur accepts if and only if $\tau = \sigma$.

uniformly at random. One option is to use known algorithms for sampling (almost) uniformly from arbitrary convex bodies, and apply them to the case of a ball. A simpler solution is the following. Take n independent samples $u_1, \ldots, u_n \in \mathbb{R}$ from the standard normal distribution and let u be the vector $(u_1, \ldots, u_n) \in \mathbb{R}^n$. Then u is distributed according to the standard n-dimensional Gaussian distribution, which is rotationally invariant. Now, choose r from the distribution on $[0, R]$ whose probability density function is proportional to r^{n-1} (this corresponds to the $(n-1)$-dimensional surface area of a sphere of radius r). The vector $\frac{r}{\|u\|}u$ is distributed uniformly in $\mathbf{B}(0, R)$.

Claim 2.6 (Completeness) *If* $\mathrm{dist}(v, \mathcal{L}(B)) > \sqrt{n}d$ *then Arthur accepts with probability 1.*

Proof: Assume $\sigma = 0$. Then

$$\mathrm{dist}(x, \mathcal{L}(B)) = \mathrm{dist}(t, \mathcal{L}(B)) \leq \|t\| \leq \frac{1}{2}\sqrt{n}d.$$

On the other hand,

$$\mathrm{dist}(x, v + \mathcal{L}(B)) = \mathrm{dist}(t, v + \mathcal{L}(B)) = \mathrm{dist}(t - v, \mathcal{L}(B)) \geq \mathrm{dist}(v, \mathcal{L}(B)) - \|t\| > \frac{1}{2}\sqrt{n}d.$$

Hence, Merlin answers correctly and Arthur accepts. The case $\sigma = 1$ is similar.

Claim 2.7 (Soundness) *If* $\mathrm{dist}(v, \mathcal{L}(B)) \leq d$ *then Arthur rejects with some constant probability.*

Proof: Let y be the difference between v and its closest lattice point. So y is such that $v - y \in \mathcal{L}(B)$ and $\|y\| \leq d$. Let η_0 be the uniform distribution on $\mathbf{B}(0, \frac{1}{2}\sqrt{n}d)$ and let η_1 be the uniform distribution on $\mathbf{B}(y, \frac{1}{2}\sqrt{n}d)$. Notice that the point Arthur sends can be equivalently seen as a point chosen from η_σ reduced modulo $\mathcal{P}(B)$. According to Corollary 2.4, $\Delta(\eta_0, \eta_1)$ is smaller than $1 - \delta$. Since statistical distance cannot increase by the application of a function,

$$\Delta(\eta_0 \bmod \mathcal{P}(B), \eta_1 \bmod \mathcal{P}(B)) \leq \Delta(\eta_0, \eta_1) < 1 - \delta$$

and Arthur rejects with probability at least δ.

3 Containment in coNP

In this section we sketch the proof of Theorem 1.3. For more details, see [1]. As mentioned in the introduction, containment in NP is trivial and it suffices to prove, e.g., that $\mathsf{GapCVP}_{100\sqrt{n}}$ is in coNP (we make no attempt to optimize the constant 100 here). To show this we construct an NP verifier that given a witness of polynomial size, verifies that the given point v is *far* from the lattice. There are three steps to the proof.

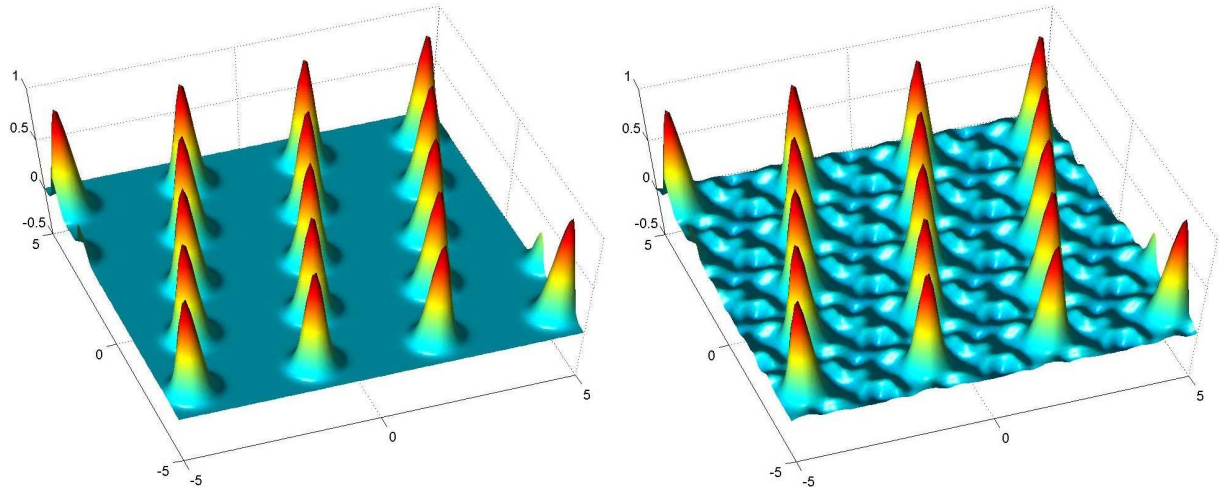

Figure 5: The function f (left) and its approximation f_W (right) for a two-dimensional lattice

1. **Define** f

   In this part we define a function $f : \mathbb{R}^n \to \mathbb{R}^+$ that is periodic over the lattice $\mathcal{L}$, i.e., for all $x \in \mathbb{R}^n$ and $y \in \mathcal{L}$ we have $f(x) = f(x + y)$ (see Figure 5). For any lattice $\mathcal{L}$, the function f satisfies the following two properties: it is non-negligible (i.e., larger than some $1/\text{poly}(n)$) for any point that lies within distance $\sqrt{\log n}$ from a lattice point, and is exponentially small at distance $\geq \sqrt{n}$ from the lattice. Hence, given the value $f(v)$, one can tell whether v is far or close to the lattice.

2. **Encode** f

   We show that there exists a succinct description (which we denote by W) of a function f_W that approximates f at *any* point in $\mathbb{R}^n$ to within polynomially small additive error (see Figure 5). We use W as the witness in the NP proof.

3. **Verify** f

   We construct an efficient NP verifier that, given a witness W, verifies that v is *far* from the lattice. The verifier verifies first that $f_W(v)$ is small, and also that $f_W(x) \geq 1/2$ for any x that is close to the lattice.

We now explain each of these steps in more detail. For all missing proofs and more details, see [1].

3.1 Step 1: Define f

Define the function $g : \mathbb{R}^n \to \mathbb{R}$ as

$$g(x) = \sum_{y \in \mathcal{L}} e^{-\pi \|x-y\|^2},$$

and let

$$f(x) = \frac{g(x)}{g(0)}.$$

Hence, f is a sum of Gaussians centered around each lattice point, and is normalized to be 1 at lattice points. See Figure 5 for a plot of f. The function f was originally used by Banaszczyk [6] to prove 'transference theorems', i.e., theorems relating parameters of a lattice to those of its dual.

The two properties mentioned above can be stated formally as follows.

Lemma 3.1 *Let $c > \frac{1}{\sqrt{2\pi}}$ be a constant. Then for any $x \in \mathbb{R}^n$, if $d(x, \mathcal{L}) \geq c\sqrt{n}$ then $f(x) = 2^{-\Omega(n)}$.*

Lemma 3.2 *Let $c > 0$ be a constant. Then for any $x \in \mathbb{R}^n$, if $d(x, \mathcal{L}) \leq c\sqrt{\log n}$ then $f(x) > n^{-10c^2}$.*

3.2 Step 2: Encode f

This step is the core of the proof. Here we show that the function f can be approximated pointwise by a polynomial size circuit with only an inverse polynomial additive error. A naive attempt would be to store f's values on some finite subset of its domain, and use these points for approximation on the rest of the domain. However, it seems that for this to be meaningful, we would have to store an exponential number of points.

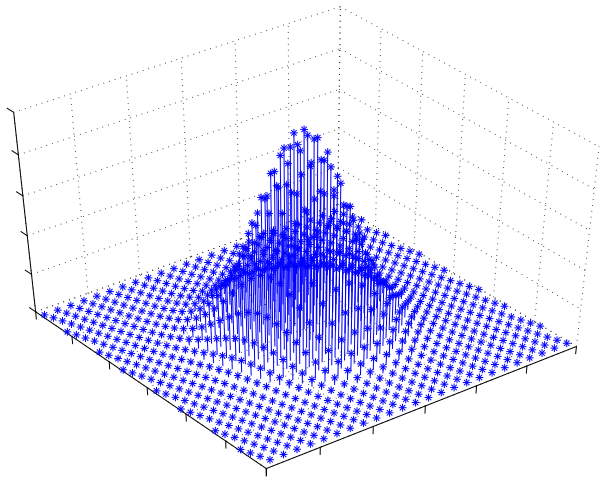

Figure 6: The Fourier series $\hat{f}$ of f

Instead, we consider the *Fourier series* of f, which is a function $\hat{f}$ whose domain is the dual lattice $\mathcal{L}^*$ (defined as the set of all points in $\mathbb{R}^n$ with integer inner product with all lattice points). For any $w \in \mathcal{L}^*$ it is given by

$$\hat{f}(w) = \frac{1}{\det(B)} \int_{z \in \mathcal{P}(B)} f(z) e^{-2\pi i \langle w, z \rangle} dz,$$

where B is some basis of $\mathcal{L}$. (It can be shown that this definition is independent of the basis we choose for $\mathcal{L}$.) A short calculation, which we omit here, shows that $\hat{f}$ has a nice form, namely

$$\hat{f}(w) = \frac{e^{-\pi\|w\|^2}}{\sum_{z\in\mathcal{L}^*} e^{-\pi\|z\|^2}}.$$

See Figure 6 for a plot of $\hat{f}$. One very useful and crucial property of $\hat{f}$ is that it is a probability distribution over the dual lattice $\mathcal{L}^*$. In other words, it is a non-negative function and the sum of all its values is 1.

A basic result in Fourier analysis is the Fourier inversion formula. It says that a function f can be recovered from its Fourier series $\hat{f}$ by using the formula

$$f(x) = \sum_{w\in\mathcal{L}^*} \hat{f}(w) e^{2\pi i\langle w,x\rangle}.$$

Since in our case both f and $\hat{f}$ are real, we can simplify it to

$$f(x) = \sum_{w\in\mathcal{L}^*} \hat{f}(w)\cos(2\pi\langle w,x\rangle)$$

by taking the real part of both sides. By thinking of $\hat{f}$ as a probability distribution, we can rewrite this as

$$f(x) = \mathrm{E}_{w\sim\hat{f}}\left[\cos(2\pi\langle w,x\rangle)\right].$$

Hence $f(x)$ can be seen as the expectation of $\cos(2\pi\langle w,x\rangle)$ (whose values range between -1 and 1), where w is chosen according to the probability distribution $\hat{f}$.

This brings us to the main idea of this step: we can approximate f by replacing the expectation with an average over a large enough sample from $\hat{f}$. More formally, for some large enough $N = \mathrm{poly}(n)$, let $W = (w_1,\ldots,w_N)$ be N vectors in the dual lattice chosen randomly and independently from the distribution $\hat{f}$, and define

$$f_W(x) \stackrel{def}{=} \frac{1}{N}\sum_{i=1}^{N}\cos(2\pi\langle x,w_i\rangle). \tag{1}$$

See Figure 5 for a plot of f_W. Then one can show that with high probability, $|f_W(x)-f(x)| \le n^{-10}$ for all $x\in\mathbb{R}^n$. The proof of this statement is based on the Chernoff-Hoeffding bound.

Given the above, it is natural to choose our NP witness to be the list $W = (w_1,\ldots,w_N)$ of vectors in the dual lattice. We note that these vectors are typically short and hence computing them directly seems difficult.

3.3 Step 3: Verify f

Here we construct an efficient NP verifier that, given the witness W, verifies that a point is *far* from the lattice. More precisely, given a lattice $\mathcal{L}$ and a vector v, it accepts if the distance of v from $\mathcal{L}$ is greater than $\sqrt{n}$ and rejects if this distance is less than $1/100$. This shows that $\mathsf{GapCVP}_{100\sqrt{n}}$ is in coNP (after appropriate rescaling).

The verifier starts by performing the following test: compute $f_W(v)$, as defined in (1), and reject if it is at least, say, $1/2$. We can do this because when the distance of v from $\mathcal{L}$ is greater than

$\sqrt{n}$, $f(v)$ is exponentially small by Lemma 3.1 and hence $f_W(v)$ must be at most $1/\text{poly}(n) < 1/2$ (assuming the witness W is chosen from $\hat{f}$ as it should be).

This verifier, however, is clearly not strong enough: the prover can 'cheat' by sending w_i's that have nothing to do with $\hat{f}$ or with the lattice, and for which $f_W(v)$ is small even though v is within distance $1/100$ of the lattice. One might try to avoid such cheating strategies by verifying that f_W is close to f everywhere, or, alternatively, that the w_i's were indeed chosen from the correct distribution $\hat{f}$. It is not known how to construct such a verifier. Instead, we will show now a somewhat weaker verifier. (This weaker verifier is what limits the proof to a gap of $\sqrt{n}$ and not $\sqrt{n/\log n}$ as one could expect given the properties of f stated in Lemmas 3.1 and 3.2.)

To test the witness W, we verify that the w_i's 'look like' vectors chosen from $\hat{f}$, according to some simple statistical tests. We will later see that these tests suffice to provide soundness. But what do vectors chosen from $\hat{f}$ look like? We identify two important properties. First, by definition we see that all the w_i's are in $\mathcal{L}^*$. Second, it turns out that with high probability, for any unit vector $u \in \mathbb{R}^n$ it holds that $\frac{1}{N}\sum_{i=1}^{N} \langle u, w_i \rangle^2$ is bounded from above by some constant, say 3. Intuitively, this follows from the fact that the length of the w_i's is roughly $\sqrt{n}$ and that they are not concentrated in any particular direction (the proof of this fact is not trivial, and is based on a lemma by Banaszczyk [6]).

Fortunately, the verifier can check these two properties efficiently. The first property is easy to check by, say, solving linear equations. But how can we check the second property efficiently? It seems that we have to check it for all unit vectors u. The main observation here is that we can equivalently check that the largest eigenvalue of the $n \times n$ matrix $W \cdot W^T$, where W is the $n \times N$ matrix whose columns are the vectors $w_1, \ldots, w_N$, is at most $3N$. This can be done in polynomial time by known algorithms for computing the eigenvalues of a matrix.

To summarize, the verifier performs the following three tests and accepts if and only if all of them are satisfied:

(a) Checks that $f_W(v) < 1/2$;

(b) Checks that W consists of vectors in the dual lattice $\mathcal{L}^*$;

(c) Checks that the maximal eigenvalue of the $n \times n$ positive semidefinite matrix WW^T is at most $3N$.

As mentioned above, if v is a YES instance, i.e., its distance from $\mathcal{L}$ is at least $\sqrt{n}$, then a witness W chosen according to $\hat{f}$ satisfies all the tests with high probability. Hence completeness holds. To complete the proof, we need to prove soundness. We will show that any witness W that passes tests (b) and (c) must satisfy $f_W(x) \geq 1/2$ for all x within distance $1/100$ from the lattice. In particular, if v is a NO instance, i.e., its distance from $\mathcal{L}$ is at most $1/100$, then test (a) must reject.

To see this, we note that by the definition of f_W, the fact that W consists of vectors in $\mathcal{L}^*$ guarantees that the function f_W is periodic on $\mathcal{L}$. Indeed, for any $v \in \mathcal{L}$,

$$\langle v + x, w_i \rangle = \langle v, w_i \rangle + \langle x, w_i \rangle$$

with the first term being integer by the definition of a dual lattice. Hence, it suffices to show that $f_W(x) \geq 1/2$ for any x satisfying $\|x\| \leq 1/100$. For such x, the eigenvalue test implies that for

most i's, $|\langle x, w_i\rangle|$ is small. Therefore, for such x most of the cosines in the definition of $f_W(x)$ are close to 1. This implies that $f_W(x)$ is greater than $1/2$ and soundness follows. In more detail, let x be such that $\|x\| \leq 1/100$. Since test (c) accepts, we have that

$$\frac{1}{N}\sum_{j=1}^{N}\langle x, w_j\rangle^2 = \frac{1}{N}x^T WW^T x \leq \frac{1}{N}\frac{3N}{10000} = \frac{3}{10000}$$

where the inequality follows by expressing x in the eigenvector basis of WW^T. Using the inequality $\cos x \geq 1 - x^2/2$ (valid for any $x \in \mathbb{R}$) we get

$$f_W(x) = \frac{1}{N}\sum_{j=1}^{N}\cos(2\pi\langle x, w_j\rangle) \geq 1 - \frac{4\pi^2}{2N}\sum_{j=1}^{N}\langle x, w_j\rangle^2 \geq 1 - \frac{6\pi^2}{10000} > \frac{1}{2}.$$

4 Zero-Knowledge Proof Systems

The containments in NP, coNP, and coAM discussed in the previous sections can be stated equivalently in terms of proof systems between a computationally unbounded prover and a polynomial time verifier. For instance, Theorem 1.3 gives a proof system for $\mathsf{coGapCVP}_{\sqrt{n}}$ in which the prover simply sends one message to the verifier who then decides whether to accept or reject. Similarly, Theorem 1.4 gives a proof system for $\mathsf{coGapCVP}_{\sqrt{n/\log n}}$ in which the prover and verifier exchange a small number of messages. Finally, for any γ, GapCVP_γ clearly has a proof system in which the prover simply sends the nearby lattice point.

In addition to the usual requirements of completeness and soundness, one can ask for proof systems that satisfy the *zero-knowledge* property. Intuitively, we say that a proof system is zero-knowledge if in the case of a true statement the verifier learns nothing beyond the validity of the statement. There are in fact two natural notions of zero-knowledge: the first is zero-knowledge against *honest verifiers*, which are verifiers that obey the protocol but still try to extract some information from the interaction. The second and stronger notion is zero-knowledge against all verifiers, which says that even if the verifier deviates from the protocol he can still learn nothing from the interaction with the prover.

Although for our purposes the above intuitive description suffices, let us mention that the formal definition of zero-knowledge uses the notion of a *simulator*. Specifically, one says that a proof system is (statistical) zero-knowledge against honest verifiers if there exists an efficient algorithm, known as a simulator, that produces communication transcripts whose distribution is statistically close to that of the actual transcripts of communication between the verifier and the prover. The existence of such a simulator captures the intuitive idea that the verifier learns nothing from the interaction. A similar definition exists for zero-knowledge against all verifiers. The concept of zero-knowledge has led to many important developments in cryptography and complexity over the past two decades. For the formal definition and further discussion see [30].

Among the three proof systems mentioned above, the only one that is zero-knowledge is the one by Goldreich and Goldwasser. (The other two are clearly not zero-knowledge since the verifier receives the witness.) Indeed, consider the protocol described in Subsection 2.4 in the case of a true statement, i.e., $\mathrm{dist}(v, \mathcal{L}(B)) > \sqrt{n}d$. Notice that the answer τ received by the verifier is always identical to his bit σ. Hence, the verifier *already knows* the answer the prover is about to send him,

and therefore can learn nothing from the protocol (beyond the fact that $\mathrm{dist}(v, \mathcal{L}(B)) > \sqrt{n}d$). This argument (once written formally) establishes that the Goldreich-Goldwasser protocol is a statistical (and in fact perfect) zero-knowledge protocol against honest verifiers, or in complexity-theoretic terms, that $\mathsf{coGapCVP}_{\sqrt{n/\log n}}$ is contained in a complexity class known as Honest Verifier Statistical Zero Knowledge, or HVSZK. This protocol is not zero-knowledge against dishonest verifiers, since by deviating from the protocol a dishonest verifier can find out if certain points are close to the lattice or not (which seems to be something he cannot do without the help of the prover). Still, using the remarkable fact that $\mathsf{HVSZK} = \mathsf{SZK}$ [30], we obtain that $\mathsf{coGapCVP}_{\sqrt{n/\log n}} \in \mathsf{SZK}$, i.e., that $\mathsf{coGapCVP}_{\sqrt{n/\log n}}$ has a zero-knowledge proof system that is secure also against dishonest verifiers. Another truly remarkable fact regarding zero-knowledge proof systems is that SZK is closed under complement [25, 30]. This implies that we also have that $\mathsf{GapCVP}_{\sqrt{n/\log n}} \in \mathsf{SZK}$, i.e., there exists a zero-knowledge proof system that allows a prover to convince a verifier that a point is close to the lattice.

4.1 Proof Systems with Efficient Provers

In the traditional complexity-theoretic definition of zero-knowledge protocols, the complexity of the prover does not play any role. However, from a cryptographic standpoint, in order for these proof systems to be useful the prover must be efficiently implementable. This gives rise to the following question: do all problems in $\mathsf{NP} \cap \mathsf{SZK}$ have a statistical zero-knowledge proof system in which the prover can be implemented efficiently when given an NP witness? Note that without providing the prover with an NP witness this task is clearly impossible. This is also the reason the question only makes sense for problems in $\mathsf{NP} \cap \mathsf{SZK}$.

In the context of lattice problems, this question was raised by Micciancio and Vadhan [23], who also made some progress towards answering the question for general problems in $\mathsf{NP} \cap \mathsf{SZK}$. Building on their work, Nguyen and Vadhan [24] very recently gave a positive answer to the question: any problem in $\mathsf{NP} \cap \mathsf{SZK}$ has a statistical zero-knowledge proof system with an efficient prover. Their protocol is secure even against dishonest verifiers.

From a theoretical point of view, Nguyen and Vadhan's exciting result gives a complete answer to our question. Yet, their construction is very complicated, and does not seem to yield protocols that are efficient in practice. For this reason, we will now describe two examples of 'practical' proof systems for lattice problems. Such direct constructions of proof systems with efficient provers have applications in cryptography, as described in [23].

The first problem we consider is $\mathsf{coGapCVP}$. As we have seen, $\mathsf{coGapCVP}_{\sqrt{n}}$ is in $\mathsf{NP} \cap \mathsf{SZK}$. However, in the Goldreich-Goldwasser proof system, the prover is required to solve a non-trivial problem, namely to tell whether a point x is within distance $\frac{1}{2}\sqrt{n}d$ from $\mathcal{L}(B)$ or within distance $\frac{1}{2}\sqrt{n}d$ from $v + \mathcal{L}(B)$ under the assumption that $\mathrm{dist}(v, \mathcal{L}(B)) > \sqrt{n}d$. This seems like a hard problem, even when given the NP witness described in Section 3. However, the Goldreich-Goldwasser protocol as described in Subsection 2.4 *does* have an efficient prover if we consider it as a protocol for the (easier) problem $\mathsf{coGapCVP}_n$. Indeed, the prover's task in this protocol is to tell whether a point x is within distance $\frac{1}{2}\sqrt{n}d$ from $\mathcal{L}(B)$ or within distance $\frac{1}{2}\sqrt{n}d$ from $v + \mathcal{L}(B)$ under the assumption that $\mathrm{dist}(v, \mathcal{L}(B)) > nd$. Notice that in the latter case the distance of x from $\mathcal{L}(B)$ is at least $nd - \frac{1}{2}\sqrt{n}d \geq nd/2$. Hence, the gap between the two cases is at least $\sqrt{n}$ and therefore

the prover can distinguish between them by using the witness described in Section 3. This proof system, just like the original Goldreich-Goldwasser protocol, is secure only against honest verifiers.

The second problem we consider is $\mathsf{GapCVP}_{\sqrt{n}}$. Here the prover's task is to convince the verifier through a zero-knowledge protocol that a point v is close to the lattice. An elegant protocol for this task was presented by Micciancio and Vadhan in [23]. Their protocol is secure even against dishonest verifiers, and in addition the prover's strategy can be efficiently implemented given a lattice point close to v. The main component in their protocol is given as Protocol 2. We use D_0 to denote the set of points that are within distance $\frac{1}{2}\sqrt{n}d$ of the lattice $\mathcal{L}(B)$ and D_1 to denote the set of points that are within distance $\frac{1}{2}\sqrt{n}d$ of the shifted lattice $v + \mathcal{L}(B)$ (see Figure 2).

Protocol 2 Part of the Micciancio-Vadhan zero-knowledge protocol for $\mathsf{GapCVP}_{\sqrt{n}}$

1. The prover chooses uniformly a bit $\sigma \in \{0, 1\}$ and sends to the verifier a point x chosen 'uniformly' from D_σ.
2. The verifier then challenges the prover by sending him a uniformly chosen bit τ.
3. The prover is supposed to reply with a point y.
4. The verifier accepts if and only if $\mathrm{dist}(x, y) \leq \frac{1}{2}\sqrt{n}d$ and $y \in \tau v + \mathcal{L}(B)$ (i.e., y is a lattice point if $\tau = 0$ and a point in the shifted lattice if $\tau = 1$).

The soundness of this protocol is easy to establish: if $\mathrm{dist}(v, \mathcal{L}(B)) > \sqrt{n}d$ then the verifier accepts with probability at most $\frac{1}{2}$ no matter what strategy is played by the prover, since no point x can be within distance $\frac{1}{2}\sqrt{n}d$ both from $\mathcal{L}(B)$ and from $v+\mathcal{L}(B)$. To prove completeness, consider the case $\mathrm{dist}(v, \mathcal{L}(B)) \leq d/10$. Using a proof similar to the one of Lemma 2.3, one can show that the relative volume of the intersection of two balls of radius $\frac{1}{2}\sqrt{n}d$ whose centers differ by at most $d/10$ is at least 0.9. This means that with probability at least 0.9, the point x chosen by the prover from D_σ is also in $D_{1-\sigma}$. In such a case, the prover is able to reply to both possible challenges τ and the verifier accepts. Notice, moreover, that the prover can be efficiently implemented if given a lattice point w within distance $d/10$ of v: by adding or subtracting $w - v$ as necessary, the prover can respond to both challenges in case x falls in $D_0 \cap D_1$.

Unfortunately, Protocol 2 is *not* zero knowledge. Intuitively, the reason for that is that when the prover is unable to answer the verifier's challenge, the verifier learns that x is outside $D_0 \cap D_1$, a fact which he most likely could not have established alone. We can try to mend this by modifying the prover to only send points x that are in $D_0 \cap D_1$. This still doesn't help since now the verifier obtains a uniform point x in $D_0 \cap D_1$, and it seems that he could not sample from this distribution alone. (This modification does, however, allow us to obtain perfect completeness.)

Instead, the solution taken by [23] is to 'amplify' Protocol 2 so as to make the information leakage negligible. Instead of just sending one point x, the prover now sends a list of $2k$ points $x_1, \ldots, x_{2k}$ each chosen independently as in the original protocol, where k is some parameter. The verifier again challenges the prover with a random bit τ. The prover is then supposed to reply with a list of points $y_1, \ldots, y_{2k}$. The verifier accepts if and only if for all i, $\mathrm{dist}(x_i, y_i) \leq \frac{1}{2}\sqrt{n}d$ and y_i is either in $\mathcal{L}(B)$ or in $v + \mathcal{L}(B)$, and moreover, the number of y_i's contained in $\mathcal{L}(B)$ is even if $\tau = 0$ and odd otherwise. The idea in this modified protocol is to allow the prover to respond to the

challenge whenever there is at least one point x_i that falls in $D_0 \cap D_1$. This reduces the probability of failure from a constant to an exponentially small amount in k. The soundness, completeness, prover efficiency, and zero-knowledge property of the modified protocol are established similarly to those of the original protocol. For further details, see [23].

Acknowledgments

This chapter is partly based on lecture notes scribed by Michael Khanevsky as well as on the paper [1] coauthored with Dorit Aharonov. I thank Ishay Haviv and the anonymous reviewers for their comments on an earlier draft. I also thank Daniele Micciancio for pointing out that the argument in Section A extends to the search version.

References

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

[2] M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract) 10-19. In *Proc. 30th ACM Symp. on Theory of Computing (STOC)*, pages 10–19. ACM, 1998.

[3] 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.

[4] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th ACM Symp. on Theory of Computing (STOC)*, pages 284–293. ACM, 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. ACM, 2001.

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

[7] J.-Y. Cai and A. Nerurkar. Approximating the SVP to within a factor $(1+1/\dim^{\varepsilon})$ is NP-hard under randomized reductions. *J. Comput. System Sci.*, 59(2):221–239, 1999.

[8] J.-Y. Cai and A. Nerurkar. A note on the non-NP-hardness of approximate lattice problems under general Cook reductions. *Inform. Process. Lett.*, 76(1-2):61–66, 2000.

[9] I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003.

[10] O. Goldreich, 2003. A comment available online at `http://www.wisdom.weizmann.ac.il/~oded/p_lp.html`.

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

[12] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inform. Process. Lett.*, 71(2):55–61, 1999.

[13] J. Håstad, B. Just, J. C. Lagarias, and C.-P. Schnorr. Polynomial time algorithms for finding integer relations among real numbers. *SIAM J. Comput.*, 18(5):859–881, 1989.

[14] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proc. 39th ACM Symp. on Theory of Computing (STOC)*, 2007.

[15] R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proc. 15th ACM Symp. on Theory of Computing (STOC)*, pages 193–206. ACM, 1983.

[16] S. Khot. Hardness of approximating the shortest vector problem in lattices. In *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 126–135. IEEE, 2004.

[17] S. Khot. Inapproximability results for computational problems on lattices, 2007. These proceedings.

[18] J. C. Lagarias, H. W. Lenstra, Jr., and C.-P. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.

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

[20] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, Mar. 2001. Preliminary version in FOCS 1998.

[21] D. Micciancio. Cryptographic functions from worst-case complexity assumptions, 2007. These proceedings.

[22] 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.

[23] D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In D. Boneh, editor, *Advances in cryptology - CRYPTO 2003, proceedings of the 23rd annual international cryptology conference*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298, Santa Barbara, California, USA, Aug. 2003. Springer-Verlag.

[24] M.-H. Nguyen and S. Vadhan. Zero knowledge with efficient provers. In *Proc. 38th ACM Symp. on Theory of Computing (STOC)*, pages 287–295. ACM, 2006.

[25] T. Okamoto. On relationships between statistical zero-knowledge proofs. In *Proc. 28th ACM Symp. on Theory of Computing (STOC)*, pages 649–658. ACM, 1996.

[26] C. J. Peikert. Limits on the hardness of lattice problems in ℓ_p norms. In *Proc. of 22nd IEEE Annual Conference on Computational Complexity (CCC)*, 2007.

[27] O. Regev. Lattice-based cryptography. In *Advances in cryptology (CRYPTO)*, pages 131–141, 2006.

[28] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2-3):201–224, 1987.

[29] C.-P. Schnorr. Factoring integers and computing discrete logarithms via diophantine approximation. In *Proc. of Eurocrypt '91*, volume 547, pages 171–181. Springer-Verlag, 1991.

[30] S. P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, MIT, 1999.

[31] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical report, University of Amsterdam, Department of Mathematics, Netherlands, 1981. Technical Report 8104.

A NP-hardness

In this section we show that Theorem 1.3 implies that $\mathsf{GapCVP}_{\sqrt{n}}$ is unlikely to be NP-hard, even under Cook reductions. One can also show that Theorem 1.4 implies that $\mathsf{GapCVP}_{\sqrt{n/\log n}}$ is unlikely to be NP-hard. However, for simplicity, we show this only for a $\sqrt{n}$ gap. Our proof is based on [22, 8, 10].

First, let us consider the simpler case of Karp reductions. If a problem in coNP is NP-hard under a Karp reduction (i.e., there is a many-to-one reduction from SAT to our problem) then the following easy claim shows that NP $\subseteq$ coNP (and hence the polynomial hierarchy collapses).

Claim A.1 *If a promise problem* $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ *is in* coNP *and is* NP*-hard under Karp reductions, then* NP $\subseteq$ coNP*.*

Proof: Take any language L in NP. By assumption, there exists an efficient procedure R that maps any $x \in L$ to $R(x) \in \Pi_{\mathrm{YES}}$ and any $x \notin L$ to $R(x) \in \Pi_{\mathrm{NO}}$. Since $\Pi \in$ coNP, we have an NP verifier V such that for any $y \in \Pi_{\mathrm{NO}}$ there exists a w such that $V(y, w)$ accepts, and for any $y \in \Pi_{\mathrm{YES}}$ and any w, $V(y, w)$ rejects. Consider the verifier $U(x, w)$ given by $V(R(x), w)$. Notice that for all $x \notin L$ there exists a w such that $U(x, w)$ accepts and moreover, for all $x \in L$ and all w $U(x, w)$ rejects. Hence, $L \in$ coNP.

The case of Cook reductions requires some more care. For starters, there is nothing special about a problem in coNP that is NP-hard under Cook reductions (for example, coSAT is such a problem). Instead, we would like to show that if a problem in NP $\cap$ coNP is NP-hard under Cook reductions, the polynomial hierarchy collapses. This implication is not too difficult to show for *total* problems (i.e., languages). However, we are dealing with *promise* problems and for such problems this implication is not known to hold (although still quite believable). In a nutshell, the difficulty arises because a Cook reduction might perform queries that are neither a YES instance nor a NO instance and for such queries we have no witness.

This issue can be resolved by using the fact that not only $\mathsf{GapCVP}_{\sqrt{n}} \in$ NP but also $\mathsf{GapCVP}_1 \in$ NP. In other words, no promise is needed in order to show that a point is close to the lattice. In the following, we show that any problem with the above properties is unlikely to be NP-hard.

Lemma A.2 *Let* $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ *be a promise problem and let* Π_{MAYBE} *denote all instances outside* $\Pi_{\text{YES}} \cup \Pi_{\text{NO}}$*. Assume that* Π *is in* coNP *and that the (non-promise) problem* $\Pi' = (\Pi_{\text{YES}} \cup \Pi_{\text{MAYBE}}, \Pi_{\text{NO}})$ *is in* NP*. Then, if* Π *is NP-hard under Cook reductions then* NP $\subseteq$ coNP *and the polynomial hierarchy collapses.*

Proof: Take any language L in NP. By assumption, there exists a Cook reduction from L to Π. That is, there exists a polynomial time procedure T that solves L given access to an oracle for Π. The oracle answers YES on queries in Π_{YES} and NO on queries in Π_{NO}. Notice, however, that its answers on queries from Π_{MAYBE} are arbitrary and should not affect the output of T.

Since $\Pi \in$ coNP, there exists a verifier V_1 and a witness $w_1(x)$ for every $x \in \Pi_{\text{NO}}$ such that V_1 accepts $(x, w_1(x))$. Moreover, V_1 rejects (x, w) for any $x \in \Pi_{\text{YES}}$ and any w. Similarly, since $\Pi' \in$ NP, there exists a verifier V_2 and a witness $w_2(x)$ for every $x \in \Pi_{\text{YES}} \cup \Pi_{\text{MAYBE}}$ such that V_2 accepts $(x, w_2(x))$. Moreover, V_2 rejects (x, w) for any $x \in \Pi_{\text{NO}}$ and any w.

We now show that L is in coNP by constructing an NP verifier. Let Φ be an input to L and let $x_1, \ldots, x_k$ be the set of oracle queries which T performs on input Φ. Our witness consists of k pairs, one for each x_i. For $x_i \in \Pi_{\text{NO}}$ we include the pair $(\text{NO}, w_1(x_i))$ and for $x_i \in \Pi_{\text{YES}} \cup \Pi_{\text{MAYBE}}$ we include the pair $(\text{YES}, w_2(x_i))$. The verifier simulates T; for each query x_i that T performs, the verifier reads the pair corresponding to x_i in the witness. If the pair is of the form (YES, w) then the verifier checks that $V_2(x_i, w)$ accepts and then returns YES to T. Similarly, if the pair is of the form (NO, w) then the verifier checks that $V_1(x_i, w)$ accepts and then returns NO to T. If any of the calls to V_1 or V_2 rejects, then the verifier rejects. Finally, if T decides that $\Phi \in L$, the verifier rejects and otherwise it accepts.

The completeness follows easily. More specifically, if $\Phi \notin L$ then the witness described above will cause the verifier to accept. In order to prove soundness, assume that $\Phi \in L$ and let us show that the verifier rejects. Notice that for each query $x_i \in \Pi_{\text{NO}}$ the witness must include a pair of the form (NO, w) because otherwise V_2 would reject. Similarly, for each query $x_i \in \Pi_{\text{YES}}$ the witness must include a pair of the form (YES, w) because otherwise V_1 would reject. This implies that T receives the correct answers for all of its queries inside $\Pi_{\text{NO}} \cup \Pi_{\text{YES}}$ and must therefore output the correct answer, i.e., that $\Phi \in L$ and then the verifier rejects. ☗

We just saw that the promise problem $\text{GapCVP}_{\sqrt{n}}$ is unlikely to be NP-hard, even under Cook reductions. Consider now the *search problem* $\text{CVP}_{\sqrt{n}}$ where given a lattice basis B and a vector v, the goal is to find a lattice vector $w \in \mathcal{L}(B)$ such that $\text{dist}(v, w) \leq \sqrt{n}\,\text{dist}(v, \mathcal{L}(B))$. This problem is clearly at least as hard as $\text{GapCVP}_{\sqrt{n}}$. Can it possibly be NP-hard (under Cook reductions)? A similar argument to the one used above shows that this is still unlikely, as it would imply NP $\subseteq$ coNP. Let us sketch this argument. Assume we have a Cook reduction from any NP language L to the search problem $\text{CVP}_{\sqrt{n}}$. Then we claim that $L \in$ coNP. The witness used to show this is a list of valid answers by the $\text{CVP}_{\sqrt{n}}$ oracle to the questions asked by the reduction, together with a witness that each answer is correct. More precisely, for each question (B, v), the witness is supposed to contain the vector $w \in \mathcal{L}(B)$ closest to v together with an NP proof that the instance $(B, v, \text{dist}(v, w)/\sqrt{n})$ is a NO instance of $\text{GapCVP}_{\sqrt{n}}$. Having the NP proof for each answer w assures us that $\text{dist}(v, w) \leq \sqrt{n}\,\text{dist}(v, \mathcal{L}(B))$ and hence w is a valid answer of the $\text{CVP}_{\sqrt{n}}$ oracle.

B Reducing GapSVP to GapCVP

Both Theorem 1.3 and Theorem 1.4 hold also for GapSVP. The following lemma shows this for Theorem 1.3. A similar argument shows this for Theorem 1.4.

Lemma B.1 *If for some $\beta = \beta(n)$, GapCVP_β is in coNP then so is GapSVP_β.*

Proof: Consider an instance of GapSVP_β given by the lattice $\mathcal{L}$ whose basis is $(b_1, \ldots, b_n)$ (in this proof we use Definitions 1.1 and 1.2 with d fixed to 1). We map it to n instances of GapCVP_β where the ith instance, $i = 1, \ldots, n$, is given by the lattice $\mathcal{L}_i$ spanned by $(b_1, \ldots, b_{i-1}, 2b_i, b_{i+1}, \ldots, b_n)$ and the target vector b_i. In the following we show that this mapping has the property that if $\mathcal{L}$ is a YES instance of GapSVP_β then at least one of $(\mathcal{L}_i, b_i)$ is a YES instance of GapCVP_β and if $\mathcal{L}$ is a NO instance then all n instances $(\mathcal{L}_i, b_i)$ are NO instances. This will complete the proof of the lemma since a NO witness for $\mathcal{L}$ can be given by n NO witnesses for $(\mathcal{L}_i, b_i)$.

Consider the case where $\mathcal{L}$ is a YES instance. In other words, if

$$u = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

denotes the shortest vector, then its length is at most 1. Notice that not all the a_i's are even for otherwise the vector $u/2$ is a shorter lattice vector. Let j be such that a_j is odd. Then the distance of b_j from the lattice $\mathcal{L}_j$ is at most $\|u\| \leq 1$ since $b_j + u \in \mathcal{L}_j$. Hence, $(\mathcal{L}_j, b_j)$ is a YES instance of GapCVP_β. Now consider the case where $\mathcal{L}$ is a NO instance of GapSVP_β, i.e., the length of the shortest vector in $\mathcal{L}$ is more than β. Fix any $i \in [n]$. By definition, $b_i \notin \mathcal{L}_i$ and therefore for any $w \in \mathcal{L}_i$ the vector $b_i - w \neq 0$. On the other hand, $b_i - w \in \mathcal{L}$ and hence $\|b_i - w\| > \beta$. This shows that $d(b_i, \mathcal{L}_i) > \beta$ and hence $(\mathcal{L}_i, b_i)$ is a NO instance of GapCVP_β.