Hardness of Lattice Problems for Use in Cryptography

Nathan Manohar

Advisor: Professor Boaz Barak

An undergraduate thesis submitted to the The School of Engineering and Applied Sciences
in partial fulfillment of the requirements for the joint degree of Bachelor of Arts in
Computer Science and Mathematics with Honors

Harvard University
Cambridge, Massachusetts
21 March 2016

Abstract

Lattice based cryptography has recently become extremely popular due to its perceived resistance to quantum attacks and the many amazing and useful cryptographic primitives that can be constructed via lattices. The security of these cryptosystems relies on the hardness of various lattice problems upon which they are based.

In this thesis, we present a number of known hardness results for lattice problems and connect these hardness results to cryptography. In particular, we show NP-hardness results for the exact versions of the lattice problems SVP, CVP, and SIVP. We also discuss the known hardness results for approximate versions of these problems and the fastest known algorithms for both exact and approximate versions of these problems. Additionally, we prove several new exponential time hardness results for these lattice problems under reasonable complexity assumptions.

We then detail how some of these hardness results can be used to construct provably secure cryptographic schemes and survey some of the recent breakthroughs in lattice based cryptography.

Acknowledgments

First, I would like to thank my thesis advisor, Prof. Boaz Barak, for spending countless hours discussing these topics and helping me compose this thesis. I would also like to thank Prof. Vinod Vaikuntanathan for advising me over the summer and posing interesting research questions pertaining to lattice based cryptography. Prof. Vinod Vaikuntanathan introduced me to many of the topics in this thesis and, through the many conversations we had, helped me to get a better sense of the current state of cryptography research.

I would also like to thank Professors Salil Vadhan and Leslie Valiant, who have both been extremely inspirational over the past four years. Through their dedicated teaching and the many conversations we had regarding research, they enabled me to find my passion for theoretical computer science, and for that I am very grateful.

I would also like to thank my friends, Prabhanjan Ananth, Prashant Vasudevan, and Akshay Degwekar, for the many interesting discussions we had over the summer regarding cryptography research and for their support during the research process.

Finally, I would like to thank my family and friends for their constant support throughout the writing of this thesis.

Contents

1 Introduction

In cryptography, the security of cryptosystems relies on the fact that there are certain problems for which no efficient algorithms are known. For example, the commonly used RSA cryptosystem is considered secure because there is no known sufficiently fast classical algorithm that computes $\phi(n)$ given n, where $\phi(n)$ is the Euler Totient function. However, fast quantum algorithms are known that can solve this problem, which means that these common cryptosystems will become insecure if a sufficiently large quantum computer can be built. Because of this possibility, cryptographers have been interested in lattice based cryptosystems, where the underlying problems for which no efficient algorithms are known are suspected to be hard even with quantum computation. In this thesis, we begin by developing the basic definitions and properties of lattices and then proceed to establish the connection between hard lattice problems and cryptography. Three such hard lattice problems are the shortest vector problem (SVP), the closest vector problem (CVP), and the shortest independent vectors problem (SIVP). In fact, even finding approximate solutions to these problems is generally hard. The approximate version of these problems come with an approximation factor $\gamma > 1$ that denotes the factor that we allow an acceptable answer to be off by. Currently, it is known how to base cryptography off of hard lattice problems when the approximation factor γ is a polynomial in n, but it is not known how to use the exact versions of these lattice problems to construct cryptographic schemes [Pei16]. Nevertheless, a considerable amount of work has been done to show hardness results for these lattice problems. In this thesis, I will mainly focus on the exact versions of these lattice problems by first presenting known hardness results and then give a few new results which I have obtained. Finally, we connect these hardness results to cryptography by showing how to use approximate versions of these lattice problems to construct cryptosystems and other useful cryptographic primitives.

The outline of the thesis is as follows: In §2, I provide a basic overview of lattices and review some fundamental results about them.

In §3, I define the key lattice problems that are applicable to cryptography and discuss the current state of knowledge by detailing the known complexity results and algorithms for these lattice problems. Furthermore, I establish the fundamental result that SVP is easier to solve than CVP [GMSS99].

In §4, I show known complexity results for the three aforementioned lattice problems, including the surprising result that CVP and SIVP are equivalent under rank-preserving reductions [Mic08].

In §5, I present my new results, namely exponential time hardness results for CVP and SIVP in any l_p norm and for SVP in the l_∞ norm assuming the exponential time hypothesis. While the results for CVP and SIVP are not difficult to show given the material in §4, they provide an interesting statement about the hardness of these problems and to the best of my knowledge, these results cannot be found in the literature. The derivation of the hardness result for SVP in the l_∞ norm is more involved than the CVP and SIVP cases, and the fact that this hardness result could only be shown for the l_∞ norm makes sense given that SVP is easier to solve than CVP and SIVP and that SVP in the l_p norm becomes an increasingly more difficult problem as p becomes larger.

In §6, I provide an overview of cryptosystems and connect the hard lattice problems

previously discussed to cryptography by giving a construction of a lattice based encryption scheme whose security follows from the hardness of approximate SVP and approximate SIVP [Reg05]. Furthermore, I describe some of the other amazing cryptographic constructions that can be derived from lattices.

Finally, in §7, I briefly discuss some open problems, which if solved, would greatly increase our confidence in the security of lattice based cryptosystems.

2 Preliminaries

In order to understand how hard lattice problems can be used to construct cryptographic schemes, we must first review some background material on lattices. The material covered in this section can be found in [MG02]. Unless otherwise specified, the norm in this section is the standard Euclidean l_2 norm.

Definition 2.1. A lattice in $\mathbb{R}^m$ is given by the set

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}, \tag{1}$$

where $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are n linearly independent vectors in $\mathbb{R}^m$ with $m \geq n$.

We say that m is the dimension of the lattice and n is the rank. If $n = m$, the lattice is said to be full rank. Virtually all of the results in this thesis will be dealing with full rank lattices, but many of the results also apply to lattices that are not full rank. Rather than write $\mathbf{b}_1, \ldots, \mathbf{b}_n$ every time, it is common practice to use the matrix

$$\mathbf{B} = [\mathbf{b}_1, \ldots \mathbf{b}_n] \in \mathbb{R}^{m \times n}, \tag{2}$$

where the $\mathbf{b}_i$'s form the columns of $\mathbf{B}$. Using this notation, we can then describe a lattice in terms of the matrix $\mathbf{B}$ and write

$$\mathcal{L}(\mathbf{B}) = \{ \mathbf{B}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n \}. \tag{3}$$

In general, all vectors $\mathbf{x}$ will be assumed to be column vectors. A simple example of a lattice Λ is

$$\Lambda = \{(a, b) : a, b \in \mathbb{Z}\}, \tag{4}$$

the set of all points in $\mathbb{R}^2$ with integer coordinates. Taking

$$\mathbf{B}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

we see that $\Lambda = \mathcal{L}(\mathbf{B}_1)$. However, this choice of basis is not unique. If we had taken

$$\mathbf{B}_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},$$

it is also true that $\Lambda = \mathcal{L}(\mathbf{B}_2)$. Figure 1 illustrates this point, showing that both $\mathbf{B}_1$ and $\mathbf{B}_2$ generate the same lattice, the set of all points in $\mathbb{R}^2$ with integer coordinates.

So, we see that multiple bases can generate the same lattice. In fact, it turns out that

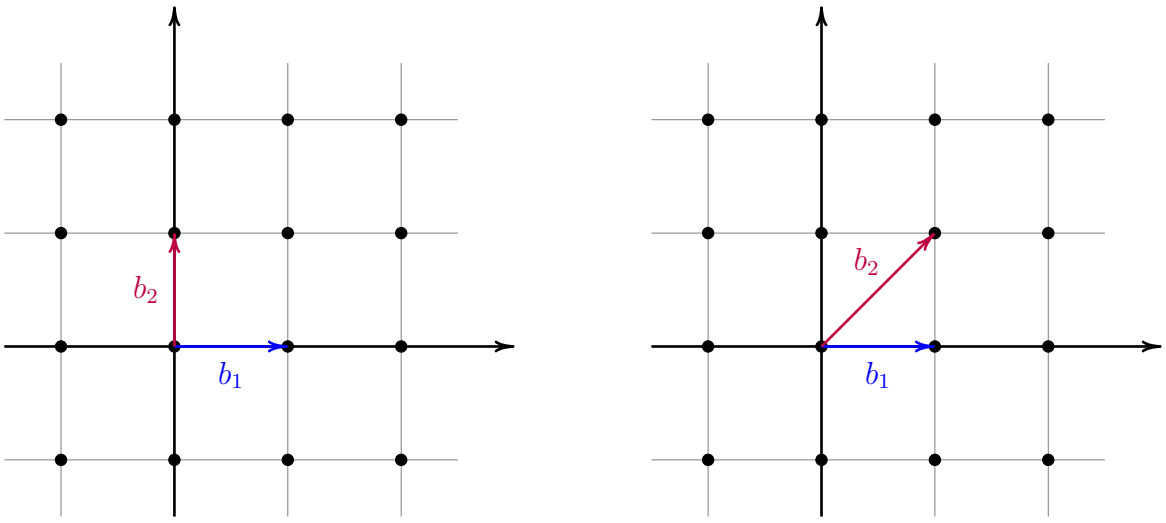

Figure 1: Two different sets of basis vectors $\{\mathbf{b}_1, \mathbf{b}_2\}$ that generate the same lattice.

Proposition 2.2. *Every lattice Λ of rank $n \geq 2$ has infinitely many bases.*

Proof. Since Λ is a lattice, it must have some basis $\mathbf{b}_1, \dots \mathbf{b}_n$. Consider the infinite set of bases given by $\mathbf{B}_r = [\mathbf{b}_1, \dots, \mathbf{b}_{n-1}, \mathbf{b}_n + r\mathbf{b}_1]$ for $r \in \mathbb{Z}$. By definition, $\Lambda = \mathcal{L}(\mathbf{B}_0)$. Now, consider any point $\mathbf{p} \in \Lambda$. We can write

$$\mathbf{p} = \sum_{i=1}^{n} x_i \mathbf{b}_i$$

for $x_i \in \mathbb{Z}$. So

$$\mathbf{p} = (x_1 - rx_n)\mathbf{b}_1 + \sum_{i=2}^{n-1} x_i \mathbf{b}_i + x_n(\mathbf{b}_n + r\mathbf{b}_1)$$

$\implies$ $\mathbf{p} \in \mathcal{L}(\mathbf{B}_r)$ and so $\Lambda \subseteq \mathcal{L}(\mathbf{B}_r)$. Conversely, for any point $\mathbf{p} \in \mathcal{L}(\mathbf{B}_r)$, we have that

$$\mathbf{p} = \sum_{i=1}^{n-1} x_i \mathbf{b}_i + x_n(\mathbf{b}_n + r\mathbf{b}_1)$$

for $x_i \in \mathbb{Z}$ and so

$$\mathbf{p} = (x_1 + rx_n)\mathbf{b}_1 + \sum_{i=2}^{n} x_i \mathbf{b}_i$$

$\implies$ $\mathbf{p} \in \Lambda$ and so $\mathcal{L}(\mathbf{B}_r) \subseteq \Lambda$. Therefore, $\Lambda = \mathcal{L}(\mathbf{B}_r)$ for all $r \in \mathbb{Z}$ and so there exist infinitely many bases. □

So, a reasonable question to ask is how does one determine if two bases generate the same lattice? We shall do this in two different ways. First, we present a geometric way of determining if two bases generate the same lattice and then we present an algebraic way.

2.1 A Geometric View of Lattices

Given a basis $\mathbf{B}$, we define the fundamental parallelepiped of $\mathbf{B}$ as follows.

Definition 2.3. Given n linearly independent vectors $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, their fundamental parallelepiped is given by

$$\mathcal{P}(\mathbf{B}) = \{\mathbf{Bx} : x_i \in \mathbb{R}, 0 \leq x_i < 1\}. \tag{5}$$

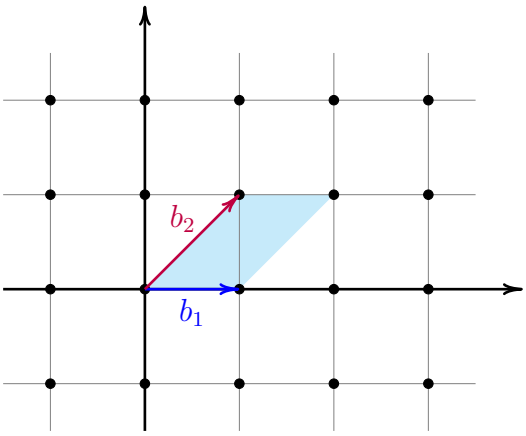

Figure 2: The fundamental parallelepiped of a two dimensional lattice.

Using the fundamental parallelepiped of a basis, we arrive at the following theorem to characterize bases of a lattice [Vai15].

Theorem 2.4. *Let Λ be a full-rank n-dimensional lattice. Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{n \times n}$ be a matrix whose column vectors are linearly independent vectors in Λ. Then, $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are a basis for $\Lambda \iff \mathcal{P}(\mathbf{B}) \cap \Lambda = \{\mathbf{0}\}$.*

Proof. ($\implies$) Suppose that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ form a basis for Λ. Then, let

$$\mathbf{p} = \mathbf{Bx} \in \Lambda \cap \mathcal{P}(\mathbf{B}) .$$

Since $\mathbf{p} \in \mathcal{P}(\mathbf{B})$, it follows that $\mathbf{x} \in [0, 1)^n$. Additionally, since $\mathbf{p} \in \Lambda$, it follows that $\mathbf{x} \in \mathbb{Z}^n$, which implies that $\mathbf{x} = \mathbf{0}$ and so $\mathbf{p} = \mathbf{0}$.

($\impliedby$) Suppose that $\mathcal{P}(\mathbf{B}) \cap \Lambda = \{\mathbf{0}\}$. Since $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are n linearly independent vectors in Λ, it follows that $\mathcal{L}(\mathbf{B}) \subseteq \Lambda$. Now, take any point $\mathbf{p} \in \Lambda$ and write $\mathbf{p} = \mathbf{Bx}$ where $\mathbf{x} \in \mathbb{R}^n$. This can be done since the span of n linearly independent vectors in $\mathbb{R}^n$ is $\mathbb{R}^n$. Consider the point $\mathbf{p}'$ defined by $\mathbf{p}' = \mathbf{B}\lfloor \mathbf{x} \rfloor$ where $\lfloor \mathbf{x} \rfloor$ is the vector whose ith component is $\lfloor x_i \rfloor$ for all i. Since all its coefficients are integers, it follows that $\mathbf{p}' \in \Lambda$. Therefore, $\mathbf{p} - \mathbf{p}' \in \Lambda$. So,

$$\mathbf{p} - \mathbf{p}' = \mathbf{B}(\mathbf{x} - \lfloor \mathbf{x} \rfloor) \in \mathcal{P}(\mathbf{B}) .$$

Thus, $\mathbf{p} - \mathbf{p}' \in \mathcal{P}(\mathbf{B}) \cap \Lambda \implies \mathbf{p} - \mathbf{p}' = \mathbf{0}$. Therefore, $\mathbf{x} - \lfloor \mathbf{x} \rfloor = \mathbf{0}$, which implies that $\mathbf{x} \in \mathbb{Z}^n$ and therefore $\mathbf{p} \in \mathcal{L}(\mathbf{B})$, so $\Lambda \subseteq \mathcal{L}(\mathbf{B})$. Therefore, we conclude that $\Lambda = \mathcal{L}(\mathbf{B})$ and that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is a basis for Λ. □

2.2 An Algebraic View of Lattices

In order to describe an algebraic characterization of when two bases generate the same lattice, we must first define what is called a unimodular matrix [Vai15].

Definition 2.5. A matrix $\mathbf{U} \in \mathbb{Z}^{n \times n}$ is said to be unimodular if $\det(\mathbf{U}) = \pm 1$.

Proposition 2.6. *If* $\mathbf{U}$ *is a unimodular matrix, then so is its inverse,* $\mathbf{U}^{-1}$.

Proof. Since $\det(\mathbf{U}) \neq 0$, $\mathbf{U}^{-1}$ exists and is given by $\mathbf{U}^{-1} = \frac{1}{\det(\mathbf{U})}\mathbf{C}^T$ where $\mathbf{C}$ is the matrix of cofactors. Since $\mathbf{U} \in \mathbb{Z}^{n \times n}$, it follows that all the cofactors are integers and so $\mathbf{C} \in \mathbb{Z}^{n \times n}$. Since $\det(\mathbf{U}) = \pm 1$, we see that $\mathbf{U}^{-1} \in \mathbb{Z}^{n \times n}$. Additionally, $\det(\mathbf{U}^{-1}) = \frac{1}{\det(\mathbf{U})} = \pm 1$, and so we conclude that $\mathbf{U}^{-1}$ is unimodular. □

Our algebraic characterization is then given by the following theorem, the proof of which is found in [Vai15].

Theorem 2.7. *Let* $\mathbf{B}, \mathbf{B}' \in \mathbb{R}^{n \times n}$ *be full-rank bases. Then,* $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$ *if and only if there exists a unimodular matrix* $\mathbf{U}$ *such that* $\mathbf{B}' = \mathbf{B}\mathbf{U}$.

Proof. ($\implies$) Suppose that $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$. Then, there exist integer matrices $\mathbf{V}, \mathbf{V}'$ such that $\mathbf{B} = \mathbf{B}'\mathbf{V}'$ and $\mathbf{B}' = \mathbf{B}\mathbf{V}$. Combining these two equations, it follows that $\mathbf{B}' = \mathbf{B}'\mathbf{V}'\mathbf{V}$. Since $\mathbf{B}'$ is a full-rank matrix, it is invertible and multiplying both sides of the equation by $(\mathbf{B}')^{-1}$, we see that $\mathbf{V}'\mathbf{V} = \mathbf{I}_n$ where $\mathbf{I}_n$ is the n dimensional identity matrix. Using the fact that $\det(\mathbf{X}\mathbf{Y}) = \det(\mathbf{X})\det(\mathbf{Y})$, it follows that $\det(\mathbf{V}')\det(\mathbf{V}) = 1$, which implies that $\det(\mathbf{V}) = \det(\mathbf{V}') = \pm 1$ since $\mathbf{V}, \mathbf{V}' \in \mathbb{Z}^{n \times n}$. Therefore, $\mathbf{V}$ and $\mathbf{V}'$ are unimodular matrices, as desired.

($\impliedby$) Suppose that there exists a unimodular matrix $\mathbf{U}$ such that $\mathbf{B}' = \mathbf{B}\mathbf{U}$. Since $\mathbf{U} \in \mathbb{Z}^{n \times n}$, it follows that we can write every column vector of $\mathbf{B}'$ as a linear combination of the column vectors of $\mathbf{B}$, which implies that $\mathcal{L}(\mathbf{B}') \subseteq \mathcal{L}(\mathbf{B})$. Similarly, $\mathbf{B} = \mathbf{B}'\mathbf{U}^{-1}$ implies that $\mathcal{L}(\mathbf{B}) \subseteq \mathcal{L}(\mathbf{B}')$ by an analogous argument since $\mathbf{U}^{-1}$ is also a unimodular matrix as shown by Prop. 2.6. Therefore, $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$, and we are done. □

2.3 The Determinant of a Lattice

Using what we have shown in §2.1 and §2.2, we are now ready to define a fundamental quantity known as the determinant of a lattice.

Definition 2.8. The determinant $\det(\Lambda)$ of a lattice Λ is the n-dimensional volume of the fundamental parallelepiped $\mathcal{P}(\mathbf{B})$ for some basis $\mathbf{B}$ of Λ.

One thing to immediately note is that $\det(\Lambda) = |\det(\mathbf{B})|$. Therefore, the determinant is well defined since if $\mathbf{B}$ and $\mathbf{B}'$ are two different bases for Λ, then by Thm. 2.7, we have that $\mathbf{B}' = \mathbf{B}\mathbf{U}$ for some unimodular matrix $\mathbf{U}$ and therefore $\det(\mathbf{B}') = \det(\mathbf{B})\det(\mathbf{U}) \implies |\det(\mathbf{B}')| = |\det(\mathbf{B})|$.

2.3.1 Gram-Schmidt Orthogonalization

Computing the determinant of a lattice Λ can easily be done via the Gram-Schmidt orthogonalization process from linear algebra. Given a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of a lattice $\mathcal{L}(\mathbf{B})$, its Gram-Schmidt orthogonalization is given by $\mathbf{b}_1^*, \ldots \mathbf{b}_n^*$ where

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$$

and

$$\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}.$$

The notation $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle$ denotes the standard inner product. Essentially what this process does is it removes from each $\mathbf{b}_i$ its projection onto the previous vectors $\mathbf{b}_1, \ldots \mathbf{b}_{i-1}$. The resulting vectors $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ span the same linear subspace as $\mathbf{b}_1, \ldots, \mathbf{b}_n$. However, it is important to note that the Gram-Schmidt orthogonalized vectors do not necessarily form a basis for $\mathcal{L}(\mathbf{B})$ since a lattice need not have an orthogonal basis. On the other hand, the Gram-Schmidt orthogonalization remains a useful notion as it allows us to easily compute the determinant of a lattice. This is because the process is a volume preserving transformation and so it essentially transforms a lattice basis $\mathbf{B}$ into a new lattice basis $\mathbf{B}^*$ whose fundamental parallelepiped is a rectangular, which makes its volume trivial to compute [Vai15]. It therefore follows that the determinant of a lattice is given by

$$\det(\mathcal{L}(\mathbf{B})) = \prod_{i=1}^{n} \|\mathbf{b}_i^*\| .$$

2.4 Successive Minima of a Lattice

One of the most fundamental properties of a lattice Λ are its successive minima. These are denoted $\lambda_1(\Lambda), \ldots, \lambda_n(\Lambda)$ and can be thought of as follows: $\lambda_i(\Lambda)$ is the radius of the smallest open ball centered at $\mathbf{0}$ that contains i linearly independent vectors of Λ. We make this notion precise via the following definitions [MG02].

Definition 2.9 (Open Ball)**.** The n-dimensional open ball of radius r centered at $\mathbf{p} \in \mathbb{R}^n$, $B_r(\mathbf{p})$, is given by $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{p}\| < r\}$.

Definition 2.10. The ith minimum $\lambda_i(\Lambda)$ of a lattice Λ is given by

$$\lambda_i(\Lambda) = \inf\{r : \dim(\text{span}(B_r(\mathbf{0}) \cap \Lambda)) \geq i\} . \tag{6}$$

We note that the successive minima of a lattice are defined with respect to any norm. In particular, they are defined with respect to all l_p norms.

The first thing we would like to show is that these successive minima actually give us useful information. In other words, it would be nice to show that $\lambda_i(\Lambda) \neq 0$, $i \geq 1$, or equivalently that there are not lattice vectors whose lengths are arbitrarily close to 0. We are most interested in $\lambda_1(\Lambda)$, which we can think of as the length of the shortest nonzero vector in the lattice. It turns out that using the Gram-Schmidt orthogonalization of the

lattice basis, we can show that $\lambda_1(\Lambda) > 0$ and so all successive minima of a lattice are bounded away from 0.

Theorem 2.11. *Let $\mathcal{L}(\mathbf{B})$ be a lattice and let $\mathbf{B}^*$ be the Gram-Schmidt orthogonalization of $\mathbf{B}$. Then,*

$$\lambda_1(\mathcal{L}(\mathbf{B})) \geq \min_i \|\mathbf{b}_i^*\| > 0\,.$$

Proof. Let $\mathbf{v} = \mathbf{B}\mathbf{x}$ for $\mathbf{x} \in \mathbb{Z}^n$ where $\mathbf{x} \neq \mathbf{0}$ denote a nonzero lattice vector. Let j be the largest index such that the component x_j of $\mathbf{x}$ is nonzero. We will show that $\|\mathbf{v}\| \geq \|\mathbf{b}_j^*\| \geq \min_i \|\mathbf{b}_i^*\|$. To do this, we first show that $|\langle \mathbf{v}, \mathbf{b}_j^* \rangle| \geq \|\mathbf{b}_j^*\|$. We have that

$$\begin{aligned} |\langle \mathbf{v}, \mathbf{b}_j^* \rangle| &= |\sum_{i \leq j} \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle x_i| \\ &= |\langle \mathbf{b}_j, \mathbf{b}_j^* \rangle x_j| \\ &= |\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle x_j| = \|\mathbf{b}_j^*\|^2 |x_j| \end{aligned}$$

Since $|x_j| \geq 1$, it follows that

$$\|\mathbf{v}\|\|\mathbf{b}_j^*\| \geq |\langle \mathbf{v}, \mathbf{b}_j^* \rangle| \geq \|\mathbf{b}_j^*\|^2,$$

and so

$$\|\mathbf{v}\| \geq \|\mathbf{b}_j^*\|,$$

since $\|\mathbf{b}_j^*\| \neq 0$ since the $\mathbf{b}_i$'s are linearly independent. Thus, for every nonzero lattice vector $\mathbf{v}$, there exists some j such that $\|\mathbf{v}\| \geq \|\mathbf{b}_j^*\|$. Therefore, all nonzero lattice vectors $\mathbf{v}$ satisfy $\|\mathbf{v}\| \geq \min_i \|\mathbf{b}_i^*\|$ and so it follows that

$$\lambda_1(\mathcal{L}(\mathbf{B})) \geq \min_i \|\mathbf{b}_i^*\| > 0$$

as desired. □

Now that we have bounded $\lambda_1(\Lambda)$ away from 0, it is not particularly hard to show that there exists a lattice vector that achieves this bound. In fact, there exists lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ whose norms are $\lambda_1, \ldots, \lambda_n$, respectively, and so for every successive minimum of a lattice, there exists a lattice vector whose norm is the value of the successive minimum [MG02]. We will now prove this result for the first successive minimum, $\lambda_1(\Lambda)$. The proofs for the other successive minima are analogous.

Theorem 2.12. *Let $\mathcal{L}(\mathbf{B})$ be a lattice. Then, there exists some $\mathbf{x} \in \mathbb{Z}^n$, $\mathbf{x} \neq \mathbf{0}$ such that $\|\mathbf{B}\mathbf{x}\| = \lambda_1(\mathcal{L}(\mathbf{B}))$.*

Proof. By the definition of $\lambda_1(\mathcal{L}(\mathbf{B}))$, there must exist a sequence of vectors $\mathbf{x}_i \in \mathbb{Z}^n$ such that $\lim_{i\to\infty} \|\mathbf{B}\mathbf{x}_i\| = \lambda_1(\mathcal{L}(\mathbf{B}))$. Now, consider the closed ball of radius r centered at $\mathbf{0}$ for some $r > \lambda_1(\mathcal{L}(\mathbf{B}))$. For sufficiently large i, it must be that the $\mathbf{B}\mathbf{x}_i$'s lie in this closed ball. But this closed ball is a compact set, which means that there must exist a convergent subsequence $\mathbf{x}_j$ such that $\lim_{j\to\infty} \mathbf{B}\mathbf{x}_j = \mathbf{v}$ where $\|\mathbf{v}\| = \lambda_1(\mathcal{L}(\mathbf{B}))$. All that remains to be shown is that $\mathbf{v}$ is a lattice vector. Since the $\mathbf{B}\mathbf{x}_j$'s converge to $\mathbf{v}$, it must be that for all

sufficiently large j, $\|\mathbf{B}\mathbf{x}_j - \mathbf{v}\| < \frac{\lambda_1(\mathcal{L}(\mathbf{B}))}{2}$. But then by the triangle inequality, it follows that for any $t > j$ that

$$\|\mathbf{B}\mathbf{x}_j - \mathbf{B}\mathbf{x}_t\| \leq \|\mathbf{B}\mathbf{x}_j - \mathbf{v}\| + \|\mathbf{v} - \mathbf{B}\mathbf{x}_t\| < \lambda_1(\mathcal{L}(\mathbf{B}))\,.$$

However, both $\mathbf{B}\mathbf{x}_j$ and $\mathbf{B}\mathbf{x}_t$ are lattice vectors, so $\mathbf{B}\mathbf{x}_j - \mathbf{B}\mathbf{x}_t$ is also a lattice vector and so it must be $\mathbf{0}$ by the definition of the first successive minima. Therefore, the sequence converges to $\mathbf{v} = \mathbf{B}\mathbf{x}_j$ and so $\mathbf{v}$ is a lattice vector. □

Now that we have succeeded in bounding $\lambda_1(\Lambda)$ away from 0 and showing that a vector whose norm is $\lambda_1(\Lambda)$ exists, it would be helpful if we could obtain some estimate of exactly how short the shortest vector is. This will be especially important when constructing cryptosystems based on lattices where we want some guarantee that sufficiently short vectors in the lattice exist. The standard way of upper bounding the successive minima of a lattice is via two theorems by Minkowski, one which provides an upper bound on $\lambda_1(\Lambda)$ and another which provides an upper bound on the product of all the successive minima. First, we will show the result known as Minkowski's first theorem, which states that

$$\lambda_1(\Lambda) < \sqrt{n}\det(\Lambda)^{1/n}.$$

To do this, we will follow the approach in [MG02] and first show two theorems, called Blichfeldt's Theorem and Minkowski's Convex Body Theorem.

Theorem 2.13 (Blichfeldt's Theorem)**.** *Let Λ be a lattice and let S be a measurable set contained in the span of Λ. If vol$(S) > \det(\Lambda)$, then there exist $\mathbf{x}_1, \mathbf{x}_2 \in S$ with $\mathbf{x}_1 \neq \mathbf{x}_2$ such that $\mathbf{x}_1 - \mathbf{x}_2 \in \Lambda$.*

Proof. Let Λ be a lattice with basis $\mathbf{B}$ and let S be a measurable set contained in the span of Λ such that $\mathrm{vol}(S) > \det(\Lambda)$. We will break up S into pairwise disjoint subsets by creating a subset $S_\mathbf{x}$ for each lattice point $\mathbf{x} \in \Lambda$. We will define $S_\mathbf{x}$ by

$$S_\mathbf{x} = (\mathcal{P}(\mathbf{B}) + \mathbf{x}) \cap S\,.$$

Addition of the set $\mathcal{P}(\mathbf{B})$ by $\mathbf{x}$ means that we get the set that is the points in $\mathcal{P}(\mathbf{B})$ translated by $\mathbf{x}$. We note that the sets $\mathcal{P}(\mathbf{B}) + \mathbf{x}$ for $\mathbf{x} \in \Lambda$ are pairwise disjoint and span span(Λ) and so it follows that the $S_\mathbf{x}$ are pairwise disjoint and span S. Furthermore, since the $S_\mathbf{x}$'s partition S and the set of lattice points is a countable set, it follows that the volume of S can expressed as the sum of the volumes of the $S_\mathbf{x}$'s. In particular, we have that

$$\mathrm{vol}(S) = \sum_{\mathbf{x}\in\Lambda} \mathrm{vol}(S_\mathbf{x})\,.$$

We now also define the sets

$$S'_\mathbf{x} = S_\mathbf{x} - \mathbf{x}\,.$$

We note immediately that $\mathrm{vol}(S'_\mathbf{x}) = \mathrm{vol}(S_\mathbf{x})$. Additionally, since $\mathrm{vol}(S) > \det(\Lambda) = \mathrm{vol}(\mathcal{P}(\mathbf{B}))$, we see that

$$\sum_{\mathbf{x}\in\Lambda} \mathrm{vol}(S'_\mathbf{x}) > \mathrm{vol}(\mathcal{P}(\mathbf{B}))\,,$$

and so the $S'_{\mathbf{x}}$'s are not pairwise disjoint since they all are contained in $\mathcal{P}(\mathbf{B})$. Let $S'_{\mathbf{x}'_1}$ and $S'_{\mathbf{x}'_2}$ be two such sets such that $S'_{\mathbf{x}'_1} \cap S'_{\mathbf{x}'_2} \neq \emptyset$. Let $\mathbf{x}$ be some point in the intersection of these two distinct sets and define

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x} + \mathbf{x}'_1\,, \\ \mathbf{x}_2 &= \mathbf{x} + \mathbf{x}'_2\,. \end{aligned}$$

We see that $\mathbf{x}_1 \in S_{\mathbf{x}'_1}$ and $\mathbf{x}_2 \in S_{\mathbf{x}'_2}$, and so it follows that $\mathbf{x}_1, \mathbf{x}_2 \in S$. Additionally, we see that $\mathbf{x}_1 - \mathbf{x}_2 = \mathbf{x}'_1 - \mathbf{x}'_2 \in \Lambda$, which is our desired result. □

Blichfeldt's Theorem is a very important result and can be used to easily show Minkowski's Convex Body Theorem.

Theorem 2.14 (Minkowski's Convex Body Theorem)**.** *Let Λ be a lattice of rank n and let S be a convex set that is symmetric about the origin and is contained in the span of Λ. Then, if $vol(S) > 2^n \det(\Lambda)$, S contains a nonzero lattice point.*

Proof. Let Λ be a rank n lattice and let S be a convex set symmetric about the origin that is contained in $\operatorname{span}(\Lambda)$. Consider the set $S' = \{\mathbf{x} : 2\mathbf{x} \in S\}$. Since S is contained in a subspace of dimension n, it follows that

$$\operatorname{vol}(S') \geq 2^{-n} \operatorname{vol}(S) > \det(\Lambda)\,.$$

Therefore, Blichfeldt's Theorem tells us that there must be distinct $\mathbf{x}_1, \mathbf{x}_2 \in S'$ such that $\mathbf{x}_1 - \mathbf{x}_2 \in \Lambda$. Additionally, we have that $2\mathbf{x}_1, 2\mathbf{x}_2 \in S$ and since S is symmetric about the origin, we have that $-2\mathbf{x}_2 \in S$. By convexity, it follows that the midpoint of $2\mathbf{x}_1$ and $-2\mathbf{x}_2$, namely

$$\frac{2\mathbf{x}_1 - 2\mathbf{x}_2}{2} = \mathbf{x}_1 - \mathbf{x}_2 \in S.$$

Therefore, we see that $\mathbf{x}_1 - \mathbf{x}_2$ is a nonzero lattice point in S. □

From Minkowski's Convex Body Theorem, a bound on the first successive minima of a lattice, $\lambda_1(\Lambda)$, can easily be shown.

Theorem 2.15 (Minkowski's First Theorem)**.** *Let Λ be a lattice of rank n. Then, the length of the shortest vector in the lattice, $\lambda_1(\Lambda)$ satisfies*

$$\lambda_1(\Lambda) < \sqrt{n} \det(\Lambda)^{1/n}.$$

Proof. Let Λ be a rank n lattice. Consider the set

$$S = B_{\sqrt{n}\det(\Lambda)^{1/n}}(\mathbf{0}) \cap \operatorname{span}(\Lambda),$$

the part of the ball of radius $\sqrt{n}\det(\Lambda)^{1/n}$ centered at the origin that lies in the span of Λ. Since S contains the n-dimensional hypercube with side length $2\det(\Lambda)^{1/n}$, it follows that

$$\operatorname{vol}(S) > 2^n \det(\Lambda),$$

and so by Minkowski's Convex Body Theorem, there exists some nonzero lattice vector $\mathbf{x}$ with $\mathbf{x} \in S$, meaning that $\|\mathbf{x}\| < \sqrt{n}\det(\Lambda)^{1/n}$. So, it immediately follows that

$$\lambda_1(\Lambda) < \sqrt{n}\det(\Lambda)^{1/n}$$

as desired. □

We can also use Minkowski's Convex Body Theorem to show that $\sqrt{n}\det(\Lambda)^{1/n}$ is also an upper bound of the geometric mean of all the successive minima of a lattice. This is known as Minkowski's Second Theorem, which we state below.

Theorem 2.16 (Minkowski's Second Theorem)**.** *Let Λ be a lattice of rank n. Then, the successive minima of Λ, $\lambda_1(\Lambda), \ldots, \lambda_n(\Lambda)$ satisfy*

$$\left(\prod_{i=1}^{n} \lambda_i(\Lambda)\right)^{1/n} < \sqrt{n}\det(\Lambda)^{1/n}.$$

Proof. Let $\Lambda = \mathcal{L}(\mathbf{B})$ be a lattice of rank n. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be linearly independent lattice vectors whose norms are the successive minima. In other words, for each i, we have that $\|\mathbf{x}_i\| = \lambda_i(\Lambda)$. Let $\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*$ denote the Gram-Schmidt orthogonalized vectors for $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Additionally, define the linear transformation T as the transformation that maps $\mathbf{x}_i^*$ to $\lambda_i(\Lambda)\mathbf{x}_i^*$ so we have that

$$T(\sum_{i=1}^{n} c_i\mathbf{x}_i^*) = \sum_{i=1}^{n} \lambda_i(\Lambda)c_i\mathbf{x}_i^*$$

Consider the set $S = B_1(\mathbf{0}) \cap \text{span}(\mathbf{B})$ which is simply the open ball of radius 1 centered at the origin intersected with $\text{span}(\mathbf{B})$ which is necessary in case the space we are working in has dimension larger than n. For the sake of contradiction, assume that the theorem does not hold and that $\prod_{i=1}^{n} \lambda_i(\Lambda) \geq n^{n/2}\det(\Lambda)$. Then, when we apply the linear transformation T to the set S, we see that

$$\begin{aligned} \text{vol}(T(S)) &= \left(\prod_{i=1}^{n} \lambda_i(\Lambda)\right)\text{vol}(S) \\ &\geq n^{n/2}\det(\Lambda)\,\text{vol}(S)\,. \end{aligned}$$

Since $\text{vol}(S)$ is the ball of radius 1 in n dimensions, we see that $\text{vol}(\sqrt{n}S) = \sqrt{n}^n\,\text{vol}(S) = n^{n/2}\,\text{vol}(S)$ since $\sqrt{n}S$ is the ball of radius $\sqrt{n}$ in n dimensions. Therefore, we have that

$$\text{vol}(T(S)) \geq \det(\Lambda)\,\text{vol}(\sqrt{n}S)\,.$$

By a similar argument to the one given before, $\text{vol}(\sqrt{n}S) > 2^n$ because $\sqrt{n}S$ contains the hypercube with edge length 2, which has volume 2^n. So, we have that

$$\text{vol}(T(S)) > 2^n\det(\Lambda)\,.$$

Additionally, we note that $T(S)$ is a convex set that is symmetric about the origin since S was convex and linear transformations preserve convexity. Additionally, S was centered around

the origin, and T did not have any translational element, since it merely scaled all the basis vectors. Therefore, by Minkowski's Convex Body Theorem, it follows that $T(S)$ contains some nonzero $\mathbf{w} \in \Lambda$. This result implies that there exists some $\mathbf{v} \in S$ such that $T(\mathbf{v}) = \mathbf{w}$, which implies that $\|\mathbf{v}\| < 1$ by the definition of S. Since the $\mathbf{x}_i^*$'s form an orthogonal basis for the span of $\mathbf{B}$, we can express both $\mathbf{v}$ and $\mathbf{w}$ in terms of this orthogonalized basis, where

$$\mathbf{v} = \sum_{i=1}^{n} c_i \mathbf{x}_i^* \,,$$

and

$$\mathbf{w} = \sum_{i=1}^{n} d_i \mathbf{x}_i^* \,,$$

where $d_i = \lambda_i(\Lambda) c_i$ for all i. Since $\mathbf{v} \neq \mathbf{0}$, there exists some $c_i \neq 0$. Let j be such that $c_j \neq 0$ and $c_i = 0$ for all $i > j$. Additionally, let $j' \leq j$ be such that $\lambda_j(\Lambda) = \lambda_{j'}(\Lambda)$ and $\lambda_i(\Lambda) < \lambda_j(\Lambda)$ for all $i < j'$. In other words, j is the largest index for which c_i is nonzero and j' is the smallest index where the successive minimum is equal to the jth successive minimum. Since the component of $\mathbf{w}$ dependent on $\mathbf{x}_j^*$ is $d_j \|\mathbf{x}_j^*\| \neq 0$, it follows that $\mathbf{w}$ is linearly independent from $\mathbf{x}_1^*, \ldots, \mathbf{x}_{j-1}^*$ and therefore $\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}$. However, if we consider $\|\mathbf{w}\|$, we see that $\|\mathbf{w}\| = \sqrt{\sum_{i=1}^{n} d_i^2 \|\mathbf{x}_i^*\|^2}$ and since $d_i = 0$ for $i > j$, it follows that

$$\begin{aligned}
\|\mathbf{w}\|^2 &= \sum_{i=1}^{j} d_i^2 \|\mathbf{x}_i^*\|^2 \\
&= \sum_{i=1}^{j} \lambda_i(\Lambda)^2 c_i^2 \|\mathbf{x}_i^*\|^2 \\
&\leq \sum_{i=1}^{j} \lambda_j(\Lambda)^2 c_i^2 \|\mathbf{x}_i^*\|^2 \\
&= \lambda_j(\Lambda)^2 \sum_{i=1}^{j} c_i^2 \|\mathbf{x}_i^*\|^2 \\
&= \lambda_j(\Lambda)^2 \|\mathbf{v}\|^2 \\
&< \lambda_j(\Lambda)^2 \,,
\end{aligned}$$

where the last inequality follows from the fact that $\mathbf{v} \in S$ and so $\|\mathbf{v}\| < 1$. Therefore, we have found that $\mathbf{x}_1, \ldots, \mathbf{x}_{j'-1}, \mathbf{w}$ are j' linearly independent vectors each with norm strictly less than $\lambda_{j'}(\Lambda)$, which contradicts the definition of the j'th successive minimum. Therefore, it follows that $\prod_{i=1}^{n} \lambda_i(\Lambda) < n^{n/2} \det(\Lambda)$ and so we conclude that

$$\left(\prod_{i=1}^{n} \lambda_i(\Lambda) \right)^{1/n} < \sqrt{n} \det(\Lambda)^{1/n} \,.$$

□

2.5 The Dual Lattice

When dealing with lattices, it is often useful to make use of the dual lattice, which is defined as follows [Mic08].

Definition 2.17 (Dual Lattice)**.** Given a lattice Λ, its dual lattice, Λ^*, is given by

$$\Lambda^* = \{\mathbf{x} \in \text{span}(\Lambda) : \forall \mathbf{y} \in \Lambda, \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}.$$

The dual lattice Λ^* has the following properties.

Proposition 2.18. *Let $\Lambda = \mathcal{L}(\mathbf{B})$ be a lattice and let Λ^* denote its dual. Then, $\mathbf{B}^* = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$ is a basis for Λ^*. Additionally, $\det(\Lambda^*) = \frac{1}{\det(\Lambda)}$.*

Proof. The fact that $\mathbf{B}^*$ is a basis for Λ^* can easily be established by set inclusion in both directions. The property of the determinant for the dual lattice follows immediately by properties of the determinant and the definition of $\mathbf{B}^*$. The full details of the above can be found in Regev's lecture notes [Reg04].

One of the most useful results about the successive minima in dual lattices is Banaszczyk's transference theorem.

Theorem 2.19 (Banaszczyk's Transference Theorem [Ban93])**.** *For any rank n lattice Λ, we have that*

$$1 \leq \lambda_1(\Lambda) \cdot \lambda_n(\Lambda^*) \leq n\,.$$

We omit the proof of this theorem, but this bound is often helpful in various reductions that require passing to the dual lattice.

3 Lattice Problems

Several computationally hard lattice problems are central to lattice based cryptography. The most common ones are the shortest vector problem (SVP), the closest vector problem (CVP), and the shortest independent vectors problem (SIVP). We define the search version of each problem over a full-rank lattice below. By the search version of the problem, we mean the problem that requires the solver to find the exact answer (the shortest vector, the closest vector, etc.). In contrast, in the decisional version of the problem (for SVP), we merely ask that the solver distinguish between cases where the shortest vector is shorter than some distance parameter $r > 0$ and cases where it is longer. The decisional version of the other problems is defined analogously. In general, all hardness results will be shown for the decisional versions of these problems, and since the decisional versions are easier problems, these hardness results also hold for the search problems as well [MG02].

Definition 3.1 (Shortest Vector Problem (SVP))**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, find a nonzero lattice vector $\mathbf{Bx}$ for $\mathbf{x} \in \mathbb{Z}^n, \mathbf{x} \neq \mathbf{0}$ such that $\|\mathbf{Bx}\| \leq \|\mathbf{By}\|$ for all $\mathbf{y} \in \mathbb{Z}^n, \mathbf{y} \neq \mathbf{0}$.

Definition 3.2 (Closest Vector Problem (CVP))**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and a target vector $\mathbf{t} \in \mathbb{Z}^n$, find a lattice vector $\mathbf{v} = \mathbf{Bx}$ for some $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{By} - \mathbf{t}\|$ for all $\mathbf{y} \in \mathbb{Z}^n$.

Definition 3.3 (Shortest Independent Vectors Problem (SIVP))**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, find lattice vectors $\mathbf{Bx}_1, \ldots \mathbf{Bx}_n \in \mathbb{Z}^m$ where $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}^n$ that are linearly independent with $\|\mathbf{Bx}_i\| \leq \lambda_n(\mathcal{L}(\mathbf{B}))$ for $1 \leq i \leq n$.

The decisional version of SVP is simply: given $(\mathbf{B}, r)$, determine if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq r$ or not. These situations correspond to YES instances and NO instances, respectively. Similarly, the decisional version of CVP is: given $(\mathbf{B}, \mathbf{t}, r)$, determine if there exists an $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\| \leq r$. If this is the case, $(\mathbf{B}, \mathbf{t}, r)$ is a YES instance, and otherwise, it is a NO instance. Finally, the decisional version of SIVP is: given $(\mathbf{B}, r)$ determine if there are n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ in the lattice given by $\mathbf{B}$ such that $\|\mathbf{v}_i\| \leq r$ for $1 \leq i \leq n$. We note that in the above decisional versions of the problem, we require that $r \in \mathbb{Q}$ be a rational number.

Furthermore, there are also what are called approximation versions of these problems. These versions take an approximation factor $\gamma > 1$. In the search version, it is required to find a lattice vector or vectors that are within a factor of γ from the optimal solution. So, for example, the approximate search version of SVP with approximation factor $\gamma > 1$ requires the solver to output a lattice vector whose norm is $\leq \gamma \|\mathbf{v}\|$ for any nonzero lattice vector $\mathbf{v}$. The approximate decisional versions of these problems are similar and are promise problems where the solver need not be able to distinguish instances where the true answer lies within a factor γ of the distance parameter. We use the term GAP to refer to the fact that we are dealing with a promise problem. For example, the approximate decisional version of SVP with approximation factor $\gamma > 1$ is denoted GAPSVP$_\gamma$ and is defined as follows.

Definition 3.4 (GAPSVP$_\gamma$)**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and a distance $r > 0$, $(\mathbf{B}, r)$ is a YES instance if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq r$ and is a NO instance if $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma \cdot r$.

GAPCVP$_\gamma$ and GAPSIVP$_\gamma$ are defined analogously.

3.1 Summary of Known Results and Algorithms

The above lattice problems have been studied for many years, and various results exist in the literature suggesting that they are hard to solve. We will focus on the shortest vector problem (SVP) in this section, since the known results for the other lattice problems are very similar in nature, but it can be shown that SVP is easier to solve than the other lattice problems. The LLL algorithm, due to A. Lenstra, H. Lenstra, and Lovász, is the most widely known algorithm to solve SVP and is capable of finding a $2^{O(n)}$ approximation to the shortest vector in polynomial time [LLL82]. Additionally, using the LLL algorithm, it is possible to solve SVP exactly in $2^{O(n^2)}$ time [Vai15]. However, despite much work on the problem, not many improvements have been made to the LLL algorithm. Currently, the best approximation algorithm for SVP that runs in polynomial time is due to [Sch87] running [AKS01] as a subroutine. It is capable of achieving an approximation factor of $2^{O(\frac{n \log \log n}{\log n})}$ [Pei16]. We note that this approximation factor is only a small improvement over the exponential approximation factor due to the LLL algorithm [LLL82], and the difficulty in lowering the approximation factor this small amount suggests that it is not possible to approximate SVP in polynomial time to within a factor significantly smaller than exponential. On the other side of the spectrum, if we are interested in the fastest algorithm that can

solve SVP exactly, [ADRSD15] gives an algorithm that runs in time 2^n, and it is currently the fastest known algorithm for exact SVP. This algorithm was discovered just last year and is still an exponential time algorithm, which suggests that it is not possible to solve SVP exactly in time significantly smaller than exponential. This makes sense since exact SVP was shown to be NP-hard under randomized reductions [Ajt98], a result which we present later in §4.3. In general, there is a tradeoff we can obtain between the running time and the approximation factor of an algorithm for SVP. Using the results in [Sch87], it is possible to obtain a 2^k-approximation algorithm for SVP that runs in time $2^{\tilde{O}(n/k)}$ [Pei16]. Another significant result is that the results just mentioned are also the best known result for quantum algorithms [Pei16]. In contrast to cryptosystems based on the hardness of factoring or discrete log, which would break down in a quantum setting due to Shor's algorithm [Sho97], cryptosystems based on the hardness of lattice problems seem to be resilient to quantum attacks. This fact makes lattice based cryptography an attractive prospect since lattice based cryptosystems will not be immediately broken should a sufficiently large quantum computer be invented.

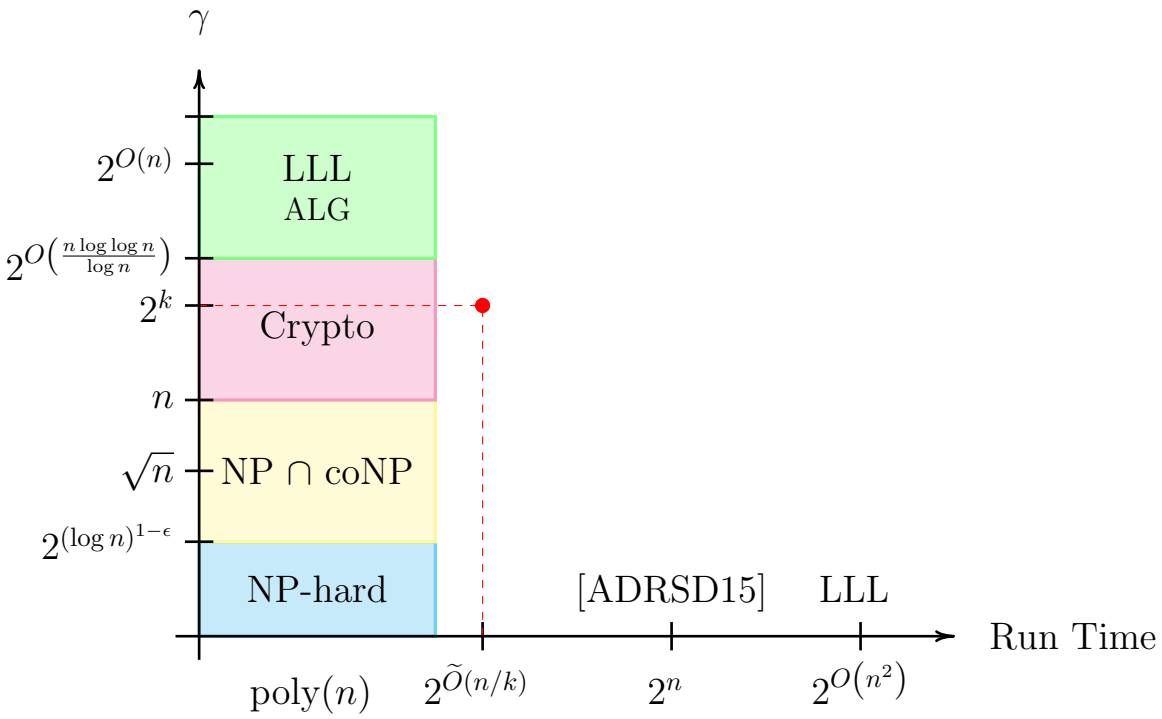

Figure 3: Known Results for SVP [Vai15].

In addition to the work on constructing algorithms for these lattice problems, a lot of work has been done to show complexity results for these lattice problems. For SVP, it was shown in [HR07] that it is NP-hard to approximate SVP to within a factor of $2^{(\log n)^{1-\varepsilon}}$ under the reasonable complexity assumption that NP $\not\subseteq$ RTIME($2^{\text{poly}(\log n)}$). Additionally, it was shown in [AR05] that for approximation factors $\gamma \geq \sqrt{n}$, SVP lies in NP $\cap$ coNP. Unfortunately, all the known cryptographic constructions based on lattice problems only hold for lattice problems with approximation factors that are polynomial in n [Pei16], so NP-

hardness results do not apply in these cases. Figure 3, which can be found in [Vai15], presents a nice overview of the known results for SVP. In particular, we notice that there is a gap between the approximation factor that is suitable for cryptography and the approximation factors for which strong hardness results are known. In the future, it would be nice to be able to bridge this gap and find a way to base cryptography on SVP with smaller approximation factors and to show stronger hardness results for SVP with larger approximation factors.

3.2 SVP is Easier to Solve than CVP

The first result we will show about lattice problems is the elementary result that SVP is easier to solve than CVP in the sense that SVP reduces to polynomially many instances of CVP of the same rank and dimension. This result makes sense intuitively since CVP asks us to find a lattice point that is closest to any target vector $\mathbf{t}$, while SVP asks us to find the shortest vector in the lattice, which can be thought of as the closest nonzero lattice point to $\mathbf{0}$. However, the fact that SVP requires us to find the shortest nonzero vector in the lattice makes the reduction not completely trivial. For example, if our reduction was to simply query our CVP oracle on the same lattice basis with target vector $\mathbf{0}$, it would output $\mathbf{0}$, which is not a valid solution for SVP. However, as our reduction shows, it is not too difficult to get around this problem. In particular, if we take our basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ and consider the n bases of the form $\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, 2\mathbf{b}_j, \mathbf{b}_{j+1}, \ldots, \mathbf{b}_n$, we will show that finding the closest vector in the jth lattice to $\mathbf{b}_j$ for all $1 \leq j \leq n$ is equivalent to finding the shortest vector in the original lattice. The intuition is that by doubling one of the basis vectors, we make it so that that basis vector is no longer in the lattice and the distance between $2\mathbf{b}_j$ and $\mathbf{b}_j$ is the same as the distance between $\mathbf{b}_j$ and $\mathbf{0}$, so finding the closest vector in the modified lattice can give us useful information about the shortest vector in the original lattice. Formally, we present the reduction from SVP to CVP which can be found in [GMSS99].

Theorem 3.5. *For any $p \geq 1$, SVP in the l_p norm on a rank n lattice is polynomial time reducible to n instances CVP in the l_p norm via a rank-preserving reduction. In particular, solving the search version for SVP on a rank n lattice requires querying a CVP oracle n times on lattices of the same dimension and rank as the SVP instance.*

Proof. We will show the result for the search version of these problems. The result for the decisional version follows via an analogous proof. Let $\mathbf{B} = \mathbf{b}_1, \ldots, \mathbf{b}_n$ denote the lattice basis for the SVP instance. Consider for each j with $1 \leq j \leq n$, the modified basis

$$\mathbf{B}_j = [\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, 2\mathbf{b}_j, \mathbf{b}_{j+1}, \ldots, \mathbf{b}_n] .$$

For each of these bases, query the CVP instance with target vector $\mathbf{b}_j$ to arrive at some output $\mathbf{w}_j$. Then, output the vector $\mathbf{v} \in \{\mathbf{w}_j - \mathbf{b}_j : 1 \leq j \leq n\}$ with the smallest norm as the shortest vector in $\mathcal{L}(\mathbf{B})$. If we wanted to solve the decisional version of the problem, we would have some distance parameter r associated with our SVP instance. Then, we would query a CVP oracle on these n instances along with distance parameter r and accept if and only if the CVP oracle accepted on at least one of our queries.

To justify that $\mathbf{v}$ is indeed the shortest vector in $\mathcal{L}(\mathbf{B})$, we will show that there is a correspondence between vectors in $\mathcal{L}(\mathbf{B})$ and vectors in $\mathcal{L}(\mathbf{B}_j)$. Let $\mathbf{v} = \sum_i c_i \mathbf{b}_i$ with $c_i \in \mathbb{Z}$

be the shortest vector in $\mathcal{L}(\mathbf{B})$. We know that there must exist some j such that c_j is odd since otherwise, we would have that $\sum_i \frac{c_i}{2}\mathbf{b}_i$ is a shorter vector in $\mathcal{L}(\mathbf{B})$ than $\mathbf{v}$. So, suppose that c_j is odd. Then, consider the vector

$$\mathbf{w} = \left(\frac{c_j + 1}{2}\right) 2\mathbf{b}_j + \sum_{i \neq j} c_i \mathbf{b}_i \,.$$

We see that $\mathbf{w} \in \mathcal{L}(\mathbf{B}_j)$ since $\frac{c_j+1}{2} \in \mathbb{Z}$ and that $\|\mathbf{w} - \mathbf{b}_j\| = \|\mathbf{v}\|$, so we can conclude that

$$\min_j \{\|\mathbf{w}_j - \mathbf{b}_j\|\} \leq \|\mathbf{v}\| \,,$$

where $\mathbf{v}$ is a vector in $\mathcal{L}(\mathbf{B})$ of smallest norm.

Conversely, suppose that

$$\mathbf{w}_j = c_j(2\mathbf{b}_j) + \sum_{i \neq j} c_i \mathbf{b}_i$$

is a lattice vector in $\mathcal{L}(\mathbf{B}_j)$. Then,

$$\mathbf{v} = (2c_j - 1)\mathbf{b}_j + \sum_{i \neq j} c_i \mathbf{b}_i$$

is a nonzero lattice vector in $\mathcal{L}(\mathbf{B})$. This follows since $2c_j - 1$ is odd, so $\mathbf{v} \neq \mathbf{0}$. Additionally, we note that

$$\|\mathbf{w}_j - \mathbf{b}_j\| = \|(2c_j - 1)\mathbf{b}_j + \sum_{i \neq j} c_i \mathbf{b}_i\| = \|\mathbf{v}\| \,,$$

and so it follows that

$$\min_j \{\|\mathbf{w}_j - \mathbf{b}_j\|\} \geq \|\mathbf{v}\| \,,$$

where $\mathbf{v}$ is the shortest vector in $\mathcal{L}(\mathbf{B})$. Combining these two inequalities, we conclude that

$$\|\mathbf{v}\| = \min_j \{\|\mathbf{w}_j - \mathbf{b}_j\|\} \,,$$

and so this procedure correctly outputs the shortest vector in $\mathcal{L}(\mathbf{B})$. Therefore, we see that solving SVP can be reduced to solving n CVP instances each of the same rank and dimension as the original SVP instance. □

4 NP-hardness of Lattice Problems

One of the most important results that we can show about these lattice problems which suggests that they are difficult to solve is that they are NP-hard. NP encapsulates the set of problems that are very useful for cryptography, namely those that are hard to solve, but for which solutions can be easily verified. By showing NP-hardness for these lattice problems, we provide a very strong result, namely that these problems cannot be solved in polynomial time unless P = NP. In this section, we will be showing NP-completeness for the decisional versions of these problems. Since the search versions of these problems are harder, this result also implies that the search versions of these problems are NP-hard. However, there is no obvious witness (e.g. given a vector how do you verify it is the shortest vector in a lattice), so it is not clear whether or not the search versions of these problems are in NP.

4.1 NP-completeness of Decisional CVP

One of the most important hardness results for lattice problems is the NP-completeness of decisional CVP, shown by Micciancio [Mic01a]. This result is shown via a polynomial time reduction from subset sum, a problem known to be NP-complete [Sip12]. First, we note that CVP is clearly in NP since given the coordinates of a lattice vector with respect to the lattice basis, the norm of the difference between the lattice vector and the target vector can easily be computed in polynomial time. To proceed with the reduction, we note that the subset sum problem is defined as follows [MG02].

Definition 4.1 (Subset Sum Problem)**.** Given $(a_1, \ldots, a_n, s) \in \mathbb{Z}^{n+1}$, find a vector $(x_1, \ldots, x_n) \in \{0,1\}^n$ such that $\sum_i x_i a_i = s$ if one exists.

The decision version of subset sum is analogous and is simply: given $(a_1, \ldots, a_n, s) \in \mathbb{Z}^{n+1}$, determine if there exists a vector $(x_1, \ldots, x_n) \in \{0,1\}^n$ such that $\sum_i x_i a_i = s$. If so, $(a_1, \ldots, a_n, s) \in \mathbb{Z}^{n+1}$ is a YES instance and otherwise, it is a NO instance. We will show the NP-completeness of decisional CVP via a reduction from the decisional subset sum problem, following the original proof of Micciancio presented in [Mic01a].

Theorem 4.2. *For any $p \geq 1$, decisional CVP in the l_p norm is NP-complete.*

Proof. First, we note that CVP is clearly in NP since given $(\mathbf{B}, \mathbf{t}, r)$, a valid witness is simply an $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\| \leq r$, which can be computed in polynomial time. To show NP-hardness, we will show a polynomial time reduction from subset sum to CVP. Given a subset sum instance $(a_1, \ldots, a_n, s) \in \mathbb{Z}^{n+1}$, we map it to the CVP instance $(\mathbf{B}, \mathbf{t}, r)$ as follows.

$$\mathbf{B} = \begin{pmatrix} 2\mathbf{a} \\ 2\mathbf{I}_n \end{pmatrix},$$

where $\mathbf{a} = (a_1, \ldots, a_n)$.

$$\mathbf{t} = \begin{pmatrix} 2s \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

where there are n 1's. r is chosen to be any rational in the interval $[n^{1/p}, (n+1)^{1/p})$. If $p = \infty$, $r = 1$. We will now show that $(\mathbf{B}, \mathbf{t}, r)$ is a YES instance of CVP if and only if $(a_1, \ldots, a_n, s)$ was a YES instance of subset sum.

($\implies$) Suppose that $(\mathbf{B}, \mathbf{t}, r)$ is a YES instance of CVP, meaning that there exists some $x \in \mathbb{Z}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\| \leq r$. We have that

$$\mathbf{Bx} - \mathbf{t} = \begin{pmatrix} 2(\sum_i x_i a_i - s) \\ 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \end{pmatrix}. \tag{7}$$

First, suppose that $p \neq \infty$. Then,

$$\|\mathbf{Bx} - \mathbf{t}\|^p = \left| 2(\sum_i x_i a_i - s) \right|^p + \sum_{i=1}^{n} |2x_i - 1|^p < n + 1 \tag{8}$$

by assumption. However, since $\mathbf{x} \in \mathbb{Z}^n$, it follows that $2x_i - 1$ is an odd integer for all i and therefore $\sum_{i=1}^n |2x_i - 1|^p \geq n$. Additionally, $\sum_{i=1}^n |2x_i - 1|^p < n+1$ if and only if $\mathbf{x} \in \{0,1\}^n$. Therefore, in order for (8) to be satisfiable, we must have that $\sum_{i=1}^n |2x_i - 1|^p = n$ and (8) is then still only satisfiable if $|2(\sum_i x_i a_i - s)|^p < 1$, which implies that $\sum_i x_i a_i - s = 0$ since the x_i's, a_i's, and s are integers. Therefore, it follows that $(a_1, \ldots, a_n, s) \in \mathbb{Z}^{n+1}$ is a YES instance of subset sum as desired. If $p = \infty$, then we see that $\|\mathbf{Bx} - \mathbf{t}\| = \max\{|2(\sum_i x_i a_i - s)|, |2x_1 - 1|, \ldots, |2x_n - 1|\} \leq 1$. This is clearly only possible if $\mathbf{x} \in \{0,1\}^n$ and $\sum_i x_i a_i - s = 0$, which implies that $(a_1, \ldots, a_n, s)$ is a YES instance of subset sum.

($\Longleftarrow$) Suppose that $(a_1, \ldots, a_n, s) \in \mathbb{Z}^{n+1}$ is a YES instance of subset sum. Then, there exists $\mathbf{x} \in \{0,1\}^n$ such that $\sum_i x_i a_i = s$. Then for $p \neq \infty$, we have that

$$\|\mathbf{Bx} - \mathbf{t}\|^p = \left|2(\sum_i x_i a_i - s)\right|^p + \sum_{i=1}^n |2x_i - 1|^p = n \leq r^p \,,$$

and so $\|\mathbf{Bx} - \mathbf{t}\| \leq r$. Similarly, if $p = \infty$, then $\max\{|2(\sum_i x_i a_i - s)|, |2x_1 - 1|, \ldots, |2x_n - 1|\} = 1 = r$ and so $\|\mathbf{Bx} - \mathbf{t}\| \leq r$. So, it follows that $(\mathbf{B}, \mathbf{t}, r)$ is a YES instance of CVP, as desired. □

4.2 NP-completeness of Decisional SIVP

We immediately note that SIVP is in NP since determining if a vector is in a lattice and determining if n vectors are linearly independent can both be done in polynomial time [MG02]. To show that SIVP is NP-hard, we present a reduction from CVP to SIVP shown in [BS99]. The general idea behind the reduction is to take the lattice basis for the CVP problem and adjoin an additional dimension that includes the target vector $\mathbf{t}$ of the CVP problem and a distance that we know is an upper bound on both the distance r of the CVP problem and $\lambda_n(\mathcal{L}(\mathbf{B}))$. Then, if the original lattice contained a vector close to $\mathbf{t}$, we can find a lattice vector in this new dimension that has a small norm, but if not, then any vector that includes the basis vector in the new dimension will have too large a norm. Having established a sketch of the main idea of the reduction, we are now ready to prove the statement formally.

Theorem 4.3. *For any $p \geq 1$, decisional SIVP in the l_p norm is NP-complete.*

Proof. We will prove the theorem for the l_2 norm, but the proof easily generalizes to any l_p norm. The fact that SIVP is in NP was shown above. NP-hardness of SIVP is shown by the following reduction. Given a CVP instance $(\mathbf{B}, \mathbf{t}, r)$, we map it to the SIVP instance $(\mathbf{B}', r')$ where

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{t} \\ \mathbf{0} & R \end{pmatrix},$$

$R = \max\{r + 1, \lceil n^{n/2} \det(\mathcal{L}(\mathbf{B})) \rceil\}$, and $r' = \|(r, R)\|$. The second term in the maximum in the definition of R follows from the fact that we are dealing with integer lattices and Minkowski's second theorem, which implies that

$$\lambda_n(\mathcal{L}(\mathbf{B})) \leq n^{n/2} \det(\mathcal{L}(\mathbf{B})) \,.$$

So we note that R is strictly greater than both r and $\lambda_n(\mathcal{L}(\mathbf{B}))$. This is the only part of the proof where we use the fact that we are working in the l_2 norm. To show the statement for any l_p norm, we simply replace the bound by Minkowski's second theorem with the appropriate one for the norm with which we are working.

Suppose that $(\mathbf{B}, \mathbf{t}, r)$ is a YES instance of CVP. Then, there exists some $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{Bx}-\mathbf{t}\| \leq r$. Writing $\mathbf{B} = [\mathbf{b}_1 \dots \mathbf{b}_n]$, we see that $(\mathbf{b}_1, 0)^T, \dots, (\mathbf{b}_n, 0)^T, (\mathbf{Bx}-\mathbf{t}, -R)^T$ are $n+1$ linearly independent vectors each of whose length is bounded by $\|(r, R)\|$. Therefore, $(\mathbf{B}', r')$ is a YES instance of SIVP.

Conversely, suppose that $(\mathbf{B}, \mathbf{t}, r)$ is a NO instance of CVP. Then, for all $\mathbf{x} \in \mathbb{Z}^n$, we have that $\|\mathbf{Bx} - \mathbf{t}\| > r$. For any set of $n+1$ linearly independent vectors in $\mathbf{B}'$, at least one of them must depend on the $(\mathbf{t}, R)^T$ column of $\mathbf{B}'$. Since this vector $\mathbf{v}$ must be a lattice vector, the coefficient c of $(\mathbf{t}, R)^T$ must be an integer. If $|c| \geq 2$, then $\|\mathbf{v}\| \geq \|2R\| > \|(r, R)\|$. If $|c| = 1$ and $\|\mathbf{v}\| \leq \|(r, R)\|$, then it follows that $\|\mathbf{v} - (\mathbf{t}, 0)^T\| \leq r$, which after removing the extraneous coordinate contradicts the fact that $(\mathbf{B}, \mathbf{t}, r)$ was a NO instance of CVP. Therefore, it follows that $\|\mathbf{v}\| > \|(r, R)\|$, and so $(\mathbf{B}', r')$ is a NO instance of SIVP. □

4.3 NP-completeness of Decisional SVP under Randomized Reductions

While showing NP-completeness for CVP and SIVP was not too difficult, showing NP-completeness for SVP is much harder. In fact, Ajtai showed in 1998 in [Ajt98] that SVP is NP-hard under randomized reductions and although many improvements have been made since then in showing hardness results for approximating SVP (for example, the results in [Mic01b], [Kho05], [HR07]), it is still unknown if SVP is NP-complete under standard deterministic reductions. The statement that SVP is NP-complete under randomized reductions is the weaker statement that SVP cannot be solved in polynomial time unless RP = NP. To show this, we will follow the proof by Micciancio in [Mic01b] which is simpler to understand than Ajtai's original proof and which can also be modified slightly to work not only for exact SVP, but also for the problem of approximating SVP to within any factor $1 \leq \gamma < 2^{1/p}$. In this thesis, however, we will only deal with the exact versions of these problems.

In order to show this result for SVP, we will present a polynomial time randomized reduction from GAPCVP$'_\gamma$ to SVP where GAPCVP$'_\gamma$ is a variant of the decisional version of the closest vector problem that is defined as follows.

Definition 4.4 (GAPCVP$'_\gamma$)**.** Let $\gamma \geq 1$ be the approximation parameter. Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, a target vector $\mathbf{t} \in \mathbb{Z}^n$, and a distance $r > 0$, $(\mathbf{B}, \mathbf{t}, r)$ is a YES instance if there exists $\mathbf{x} \in \{0, 1\}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\| \leq r$. $(\mathbf{B}, \mathbf{t}, r)$ is a NO instance if for all $\mathbf{x} \in \mathbb{Z}^n$ and $j \in \mathbb{Z} \setminus \{0\}$, we have that $\|\mathbf{Bx} - j\mathbf{t}\| > \gamma r$.

We immediately note that GAPCVP$'_\gamma$ is a promise problem, meaning that there are some instances $(\mathbf{B}, \mathbf{t}, r)$ that are neither YES instances nor NO instances. We do not worry about these instances since we only care about whether we can distinguish between YES and NO instances. Furthermore, it was shown in [ABSS97] that GAPCVP$'_\gamma$ is NP-complete, so a randomized reduction from GAPCVP$'_\gamma$ to SVP would in fact show the NP-completeness of SVP under randomized reductions. The idea for the reduction goes as follows. Suppose we

are given a GAPCVP$'_\gamma$ instance $(\mathbf{B}, \mathbf{t}, r)$. To convert this to an instance of SVP, we could simply adjoin $\mathbf{t}$ to the lattice basis $\mathbf{B}$ and consider the lattice whose basis is given by the matrix $(\mathbf{B}|\mathbf{t})$. If $(\mathbf{B}, \mathbf{t}, r)$ was a NO instance, then for all $\mathbf{x} \in \mathbb{Z}^{n+1}$, $\|(\mathbf{B}|\mathbf{t})\,\mathbf{x}\| > r$ and so this would be a NO instance of SVP. The problem arises when we consider YES instances. It could be that the shortest vector in the lattice with basis $(\mathbf{B}|\mathbf{t})$ is of the form $\mathbf{Bx}$ and does not depend on $\mathbf{t}$ at all. In order to get around this, we will embed the lattice with basis $(\mathbf{B}|\mathbf{t})$ into a higher dimensional space. Essentially, in the new dimensions, we will have another lattice with the property that it does not have any short vectors but has vectors close enough to some fixed point. To show the existence of such a lattice, we make use of the following lemma.

Lemma 4.5. *For any l_p norm for $p \geq 1$ and any constant $\gamma \in [1, \sqrt[p]{2})$, there exists a probabilistic algorithm that on an input n outputs in polynomial time, two positive integers $m, r \in \mathbb{Z}^+$, a lattice basis $\mathbf{B} \in \mathbb{Z}^{(m+1)\times m}$, a vector $\mathbf{v} \in \mathbb{Z}^{m+1}$, and a linear integer transformation $\mathbf{T} \in \mathbb{Z}^{n\times m}$ such that $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma r$ and with probability at least $1 - 1/poly(n)$, for all $\mathbf{x} \in \{0,1\}^n$, there exists some $\mathbf{y} \in \mathbb{Z}^m$ such that $\mathbf{Ty} = \mathbf{x}$ and $\mathbf{By} \in B_r(\mathbf{v})$.*

The proof of the above lemma can be found in [Mic01b], but we omit it here since it is extremely tedious and not particularly enlightening. The important point of Lemma 4.5 is that it shows that it is possible to construct a lattice Λ with no "short" vectors (all have length $> r$, but for which there exists some point $\mathbf{v}$ for which there are exponentially many close vectors (all within distance r)). Furthermore, there exists an integer linear transformation $\mathbf{T}$ with the property that for any vector $\mathbf{x} \in \{0,1\}^n$, we can find a preimage $\mathbf{y} \in \mathbb{Z}^m$ that represents the coordinates of a lattice vector close to $\mathbf{v}$.

Armed with this lemma, we can now proceed with the proof of the NP-completeness of SVP under randomized reductions.

Theorem 4.6. *For any $p \geq 1$, decisional SVP is NP-complete under randomized reductions in the l_p norm with inverse polynomial error probability*

Proof. We first note that it is clear that SVP is in NP since given the coordinates of a lattice vector, it is easy to verify if its norm is smaller than the distance parameter r. As stated above, we will show the NP-hardness of SVP under randomized reductions via a reduction from GAPCVP$'_\gamma$. Fix γ' to be a rational between 1 and $\sqrt[p]{2}$ and let γ be a rational greater than

$$(1 - \gamma'^{-p})^{-1/p}\,.$$

Suppose we are given a GAPCVP$'_\gamma$ instance $(\mathbf{B}, \mathbf{t}, r)$. We map this to the SVP instance $(\mathbf{L}, s)$. We will show that NO instances of GAPCVP$'_\gamma$ are mapped to NO instances of SVP, and YES instances of GAPCVP$'_\gamma$ are mapped to YES instances of SVP with probability $1 - 1/\operatorname{poly}(n)$, where n is the dimension of the lattice $\mathcal{L}(\mathbf{B})$. To obtain $(\mathbf{L}, s)$, we do the following. First, we run the probabilistic algorithm from Lemma 4.5 on input n and parameter γ' to obtain $m, d \in \mathbb{Z}^+$, a lattice basis $\mathbf{B'} \in \mathbb{Z}^{(m+1)\times m}$, a vector $\mathbf{v} \in \mathbb{Z}^{m+1}$ and a linear transformation $\mathbf{T} \in \mathbb{Z}^{n\times m}$, where $\lambda_1(\mathcal{L}(\mathbf{B'})) > \gamma' d$ and with probability $\geq 1 - 1/\operatorname{poly}(n)$, for all $\mathbf{x} \in \{0,1\}^n$, there exists some $\mathbf{y} \in \mathbb{Z}^m$ such that $\mathbf{Ty} = \mathbf{x}$ and $\mathbf{B'y} \in B_d(\mathbf{v})$. Let

$$\mathbf{L} = \begin{pmatrix} d\gamma' \cdot \mathbf{BT} & d\gamma' \cdot \mathbf{t} \\ r\gamma \cdot \mathbf{B'} & r\gamma \cdot \mathbf{v} \end{pmatrix}.$$

Define $s = \gamma\gamma' rd$. We now show that $(\mathbf{L}, s)$ satisfies our desired properties. Suppose that $(\mathbf{B}, \mathbf{t}, r)$ was a NO instance of GAPCVP$'_\gamma$. Let

$$\mathbf{a} = \begin{pmatrix} \mathbf{b} \\ c \end{pmatrix}$$

be any nonzero vector in $\mathbb{Z}^{m+1}$. Then, it follows that

$$\|\mathbf{La}\| = \|(d\gamma'\mathbf{BTb} + d\gamma' c\mathbf{t}, r\gamma\mathbf{B}'\mathbf{b} + r\gamma c\mathbf{v})\|\,,$$

and so if we are working in the l_p norm (assume $p \neq \infty$ since this case must be treated slightly differently and the proof technique is immediately obvious from the following), we have that

$$\|\mathbf{La}\|_p^p = (d\gamma'\|\mathbf{BTb} + c\mathbf{t}\|_p)^p + (r\gamma\|\mathbf{B}'\mathbf{b} + c\mathbf{v}\|_p)^p\,.$$

We will now analyze two cases, namely the case where $c = 0$ and the case where $c \neq 0$. Suppose that $c = 0$. Then, we have that $\mathbf{b} \neq \mathbf{0}$ and so

$$r\gamma\|\mathbf{B}'\mathbf{b} + c\mathbf{v}\|_p = r\gamma\|\mathbf{B}'\mathbf{b}\|_p > \gamma\gamma' rd = s$$

since $\lambda_1(\mathcal{L}(\mathbf{B}')) > \gamma' d$ and so $(\mathbf{L}, s)$ is a NO instance of SVP. Alternatively, if $c \neq 0$, then since $(\mathbf{B}, \mathbf{t}, r)$ is a GAPCVP$'_\gamma$ instance, $\|\mathbf{Bx} + c\mathbf{t}\|_p > \gamma r$ for any $x \in \mathbb{Z}^n$, and so it follows that

$$d\gamma'\|\mathbf{BTb} + c\mathbf{t}\|_p > \gamma\gamma' rd = s\,,$$

and so $(\mathbf{L}, s)$ is a NO instance.

Suppose that $(\mathbf{B}, \mathbf{t}, r)$ was a YES instance of GAPCVP$'_\gamma$ and suppose that the probabilistic algorithm from Lemma 4.5 succeeds in producing a $\mathbf{T}$ with the desired properties, which occurs with probability $\geq 1 - 1/\operatorname{poly}(n)$. Since $(\mathbf{B}, \mathbf{t}, r)$ is a YES instance of GAPCVP$'_\gamma$, there exists some $x \in \{0,1\}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\|_p \leq r$. By our construction, there exists some preimage $y \in \mathbb{Z}^m$ such that $\mathbf{Ty} = \mathbf{x}$ and $\mathbf{B}'\mathbf{y} \in B_d(\mathbf{v})$. Therefore, if we take the vector

$$\mathbf{z} = \begin{pmatrix} \mathbf{y} \\ -1 \end{pmatrix}\,,$$

we find that

$$\|\mathbf{La}\| = \|(d\gamma'\mathbf{BTy} - d\gamma'\mathbf{t}, r\gamma\mathbf{B}'\mathbf{y} - r\gamma\mathbf{v})\|\,,$$

and so it follows that

$$\begin{aligned}
\|\mathbf{La}\|_p^p &= (d\gamma'\|\mathbf{Bx} - \mathbf{t}\|_p)^p + (r\gamma\|\mathbf{B}'\mathbf{y} - \mathbf{v}\|_p)^p \\
&\leq (\gamma' rd)^p + (\gamma rd)^p \\
&= \left(\frac{s}{\gamma}\right)^p + \left(\frac{s}{\gamma'}\right)^p \\
&= s^p\left(\frac{1}{\gamma^p} + \frac{1}{\gamma'^p}\right) \\
&\leq s^p\left(\left(1 - \frac{1}{\gamma'^p}\right) + \frac{1}{\gamma'^p}\right) \\
&= s^p\,,
\end{aligned}$$

and so it follows that $\|\mathbf{La}\|_p \leq s$ and so $(\mathbf{L}, s)$ is a YES instance of SVP. □

4.4 Equivalence of CVP and SIVP under Rank-Preserving Reductions

Although we have succeeded in showing the NP-completeness of these lattice problems under polynomial time reductions (deterministic in the case of CVP and SIVP and randomized in the case of SVP), we have done little in the way of constructing a procedure for actually finding the closest vector in a lattice to some target vector (search CVP) or a set of n linearly independent vectors whose norms are all $\leq \lambda_n(\Lambda)$ (search SIVP). In practice, it is often not enough to be able to simply determine if there is a lattice vector within a certain distance of some target vector; we want to actually find the closest lattice vector. In 2008, Micciancio was able to improve on some of the previous constructions described earlier and provided a method for finding the closest lattice vector to a target vector given an oracle to SIVP and conversely, a method for finding n linearly independent vectors whose norms are all $\leq \lambda_n(\Lambda)$ given a CVP oracle [Mic08]. Moreover, these constructions are rank-preserving meaning that in order to solve CVP on a lattice of rank n, one must only query the SIVP oracle on rank n lattices and vice versa. The importance of this fact will become clear later on when we discuss exponential time hardness. Intuitively, this means that CVP and SIVP are equally hard to solve since solving a rank n instance of one problem reduces to solving a rank n instance of another. However, as shown in Section 4.3, solving a rank n instance of CVP reduces to solving a rank n^c instance of SVP for some $c > 1$, suggesting that SVP is an easier problem to solve than CVP since we must solve a harder instance of SVP (an SVP instance of larger dimension) in order to solve CVP. At the conclusion of this section, we will have shown that CVP and SIVP (and all less standard lattice problems used as intermediaries in our reductions) are equally hard to solve and that SVP is at most as hard to solve as these other problems, but is most likely easier to solve.

In order to show the equivalence between CVP and SIVP, we first show a rank-preserving reduction from SIVP to CVP. To accomplish this, we make use of two additional lattice problems, SMP and SVP$'$, that are defined as follows.

Definition 4.7 (Successive Minima Problem (SMP))**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, find lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n \in \mathbb{Z}^m$, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}^n$, that are linearly independent with $\|\mathbf{Bx}_i\| \leq \lambda_i(\mathcal{L}(\mathbf{B}))$ for $1 \leq i \leq n$.

Definition 4.8 (SVP')**.** Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and an index $i \in \{1, \ldots, n\}$, find a lattice vector $\mathbf{v} = \mathbf{Bx}$ for some $\mathbf{x} \in \mathbb{Z}^n$ with $x_i \neq 0$ such that $\|\mathbf{v}\| \leq \|\mathbf{Bx}'\|$ for all $\mathbf{x}' \in \mathbb{Z}^n$ with $x'_i \neq 0$.

Looking at the definition of SMP, we see that it is simply a more restrictive version of SIVP. In particular, a solution to SMP is necessarily a solution to SIVP. From the definition of SVP$'$, we see that this problem is simply a more difficult version of SVP. In particular, if one can solve SVP$'$ for all indices $i \in \{1, \ldots, n\}$, then one can trivially solve SVP by simply comparing the norms of the n vectors output by the SVP$'$ oracle for $i = 1, \ldots, n$ and outputting the one with the smallest norm.

Following the proof of Micciancio in [Mic08], we will reduce SIVP to SMP, then reduce SMP to SVP$'$, and finally reduce SVP$'$ to CVP. The reduction from SIVP to SMP is trivial as it is simply the identity map since a solution to SMP for a given lattice is also a solution

to SIVP as noted above. The reduction from SMP to SVP$'$ is more involved and makes use of the following lemma.

Lemma 4.9. *There exists a polynomial time algorithm that given a lattice basis $\mathbf{B} \in \mathbb{Q}^{m \times n}$ and a linear subspace S, outputs a new basis $\mathbf{B}'$ such that $\mathcal{L}(\mathbf{B}') = \mathcal{L}(\mathbf{B})$ and $\mathcal{L}(\mathbf{b}'_1, \ldots, \mathbf{b}'_d) = \mathcal{L}(\mathbf{B}) \cap S$ where $d = dim(span(\mathbf{B}) \cap S)$.*

We omit the proof here, but the result can be found in [Mic08]. Using this lemma, the idea behind the reduction is as follows. By using an oracle for SVP$'$, we will proceed to build up a set of n linearly independent vectors satisfying the properties specified by SMP one at a time. If we have found j vectors so far, we let the span of these j vectors be the linear subspace S and then apply Lemma 4.9 to arrive at a new lattice basis. Using this, we can run the SVP$'$ oracle to find short vectors in the lattice that are linearly independent of the ones previously found since these only depend on the first j basis vectors. We then show that one of these short vectors found by the SVP$'$ oracle must be short enough for our purposes and we can extend our set of linearly independent vectors by 1. Repeating this process n times starting from $S = \emptyset$ yields a set of n linearly independent vectors that are a solution to SMP. Formally, we have the following.

Theorem 4.10. *For any l_p norm, there is a polynomial time reduction from SMP to SVP$'$ that preserves the rank and dimension of the input lattice.*

Proof. Suppose we are given a lattice basis $\mathbf{B}$ and we want to solve SMP. The idea is to compute vectors $\mathbf{v}_i \in \mathcal{L}(\mathbf{B})$ with $\|\mathbf{v}_i\| \leq \lambda_i(\mathcal{L}(\mathbf{B}))$. We will begin with the empty set and iteratively obtain one vector to add to our set until we have found n linearly independent vectors. Suppose that we have found j lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_j$ with $\|\mathbf{v}_i\| \leq \lambda_i(\mathcal{L}(\mathbf{B}))$ for $1 \leq i \leq j$. Let $S = \text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_j)$. Run the algorithm given by Lemma 4.9 on $\mathbf{B}$ and S to find a basis $\mathbf{B}'$ such that $\mathcal{L}(\mathbf{b}'_1, \ldots, \mathbf{b}'_j) = \mathcal{L}(\mathbf{B}) \cap S$. This follows since the dimension of S is j since the j basis vectors of S are linearly independent and lie in $\mathcal{L}(\mathbf{B})$. Next, run the SVP$'$ oracle on input $\mathbf{B}'$ and i for all i where $j + 1 \leq i \leq n$. Define $\mathbf{v}_{j+1}$ to be the shortest of the vectors output by the $n - j$ calls to the SVP$'$ oracle. Since the ith coordinate of $\mathbf{v}_{j+1}$ is not 0 for some i with $j + 1 \leq i \leq n$, it follows that $\mathbf{v}_{j+1} \notin S$. Additionally, we see that $\|\mathbf{v}_{j+1}\| \leq \lambda_{j+1}(\mathcal{L}(\mathbf{B}))$. To see this, we note that there exist $j+1$ linearly independent vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{j+1}$ in $\mathcal{L}(\mathbf{B}')$ such that $\|\mathbf{w}_i\| \leq \lambda_{j+1}(\mathcal{L}(\mathbf{B}'))$ since $\mathbf{B}$ and $\mathbf{B}'$ are bases for the same lattice. Since S has dimension j, there must exist some $\mathbf{w}_t \notin S$ for some $t \in \{1, \ldots, j + 1\}$. Since $\mathbf{w}_t \in \mathcal{L}(\mathbf{B}')$, there must exist some $\mathbf{y} \in \mathbb{Z}^n$ such that $\mathbf{w}_t = \mathbf{B}'\mathbf{y}$. Additionally, we must have that $y_k \neq 0$ for some index $k \geq j + 1$ since otherwise we would have that $\mathbf{w}_t \in S$ since $\text{span}(\mathbf{b}'_1, \ldots, \mathbf{b}'_j) \in S$. Therefore, when our algorithm calls the SVP$'$ oracle on input $\mathbf{B}'$ and k, it must output some vector whose norm is at most $\|\mathbf{w}_t\| \leq \lambda_{j+1}(\mathcal{L}(\mathbf{B}))$. So, it follows that the shortest vector returned by the $n - j$ calls to the SVP$'$ oracle will have norm at most $\lambda_{j+1}(\mathcal{L}(\mathbf{B}))$. So, by applying this procedure n times, we can construct a set of n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ such that $\|\mathbf{v}_i\| = \lambda_i(\mathcal{L}(\mathbf{B}))$ for each i. The equality in the above statement follows from the definition of $\lambda_i(\mathcal{L}(\mathbf{B}))$, which states that there cannot exist i linearly independent vectors in $\mathcal{L}(\mathbf{B})$ that all have norm strictly less than $\lambda_i(\mathcal{L}(\mathbf{B}))$. So, we have shown how to solve SMP given an SVP$'$ oracle, completing the proof. □

We now proceed to the second step of the reduction, where we reduce SVP′ to CVP. The reduction between SVP′ and CVP is similar to the reduction between SVP and CVP in §3.2 in that in both reductions we modify one of the basis vectors in our SVP or SVP′ instance in order to arrive at a CVP instance. The general idea behind this reduction is that if we are given an SVP′ instance $(\mathbf{B}, i)$, we modify the lattice basis $\mathbf{B}$ so that the ith basis vector is much larger than normal. We then set our target vector $\mathbf{t}$ to be a different large multiple of the ith basis vector. Intuitively, this makes it so that the closest vector in the new basis to $\mathbf{t}$ will include the ith basis vector and the closest vector in this lattice mirrors the shortest vector in $\mathcal{L}(\mathbf{B})$ that has a nonzero ith component. Formally, we have the following result.

Theorem 4.11. *For any l_p norm, there is a polynomial time reduction from SVP′ to CVP that preserves the rank and dimension of the input lattice.*

Proof. Let $(\mathbf{B}, i)$ be a given SVP′ instance. Let α be such that

$$\alpha \leq \text{dist}(\mathbf{b}_i, \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n)),$$

so α is a lower bound of the distance between $\mathbf{b}_i$ and the linear subspace spanned by the other basis vectors. Query the CVP oracle on $(\mathbf{B}_j, \mathbf{t}_j)$ for

$$j = 0, \ldots, \lfloor \log_2 \|\mathbf{b}_i\|/\alpha \rfloor ,$$

where

$$\mathbf{B}_j = [\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, 2^{j+1}\mathbf{b}_i, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n] ,$$

and

$$\mathbf{t}_j = 2^j \mathbf{b}_i .$$

So, $\mathbf{B}_j$ is given by taking $\mathbf{B}$ and multiplying the ith basis vector by 2^{j+1} and $\mathbf{t}_j$ is the ith basis vector multiplied by 2^j. Let $\mathbf{w}_j$ denote the vector returned by the CVP oracle to the query $(\mathbf{B}_j, \mathbf{t}_j)$. Compute $\mathbf{w}_j - \mathbf{t}_j$ for all j and return the vector with the smallest norm as the answer to the SVP′ query $(\mathbf{B}, i)$.

We now show that the vector returned by this procedure is indeed the shortest vector in $\mathcal{L}(\mathbf{B})$ where the ith component is nonzero. First, we note that any vector returned by our procedure will be of the form $\mathbf{B}_j\mathbf{x}_j - \mathbf{t}_j$ for some integer vector $\mathbf{x}_j$, and so it follows that the component of $\mathbf{b}_i$ will be $2^{j+1}x_{j,i} - 2^j \neq 0$ since $x_{j,i} \in \mathbb{Z}$. Therefore, the returned vector satisfies the required property to be a solution to the SVP′ instance $(\mathbf{B}, i)$. Let $\mathbf{Bx}$ denote the optimal solution to the SVP′ instance $(\mathbf{B}, i)$. Let j denote the largest power of 2 that divides x_i. So, $2^j | x_i$, but $2^{j+1} \nmid x_i$. Let y be the integer such that

$$x_i = 2^j(2y - 1) .$$

Then, if we let

$$\mathbf{z} = [x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n] ,$$

we see that

$$\mathbf{B}_j\mathbf{z} - \mathbf{t}_j = \mathbf{Bx} ,$$

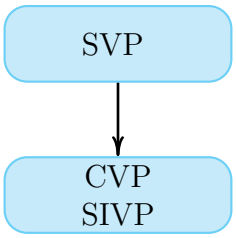

Figure 4: Known reductions between exact versions of SVP, CVP, and SIVP.

since the ith component has coefficient

$$2^{j+1}y - 2^j = 2^j(2y-1) = x_i\,,$$

and the other components have the same coefficients. Therefore, the reduction will output the shortest vector $\mathbf{Bx}$ with $x_i \neq 0$ provided that the value of j above is one of the j's we use to query the CVP oracle. We need to show that $\|\mathbf{b}_i\|/\alpha \geq 2^j$. We will show that $\|\mathbf{b}_i\|/\alpha \geq |x_i| \geq 2^j$. This follows since any vector of the form $\mathbf{Bx}$ has norm at least $\alpha|x_i|$ since α is the distance between $\mathbf{b}_i$ and the linear subspace spanned by the other basis vectors. Additionally, since $\mathbf{Bx}$ was the shortest vector with $x_i \neq 0$, it must be that $\|\mathbf{Bx}\| \leq \|\mathbf{b}_i\|$, and so it follows that $\alpha|x_i| \leq \|\mathbf{b}_i\|$ and so $\|\mathbf{b}_i\|/\alpha \geq 2^j$ showing that the reduction will indeed find the shortest vector $\mathbf{Bx}$ with $x_i \neq 0$. □

Combining these reductions, we arrive at the following theorem.

Theorem 4.12. *For any l_p norm, there is a polynomial time reduction from SIVP to CVP that preserves the rank and dimension of the input lattice.*

Proof. Simply use the identity function to map an SIVP instance to an SMP instance. Then, use the reductions in Theorems 4.10 and 4.11 to arrive at a set of CVP queries that can be used to solve the given SIVP instance. □

To finish showing the equivalence between SIVP and CVP, we must present a rank and dimension preserving reduction in the other direction. In §4.2, we showed a reduction from CVP to SIVP that mapped rank n CVP instances to rank $n+1$ SIVP instances. [Mic08] presents a modification of the above reduction that succeeds in mapping rank n CVP instances to rank n SIVP instances. The idea behind this reduction is to make use of the dual lattice. If we query our SIVP oracle on our original lattice Λ, we can find a nonzero lattice vector $\mathbf{v} \in \Lambda$ with $\|\mathbf{v}\| \leq \lambda_n(\Lambda)$. Using $\mathbf{v}$ and Banaszczyk's Transference Theorem (Thm. 2.19), it is then possible to find $\mathbf{v}^*$, the shortest vector in Λ^*. Using $\mathbf{v}^*$, it is possible to solve CVP in the original lattice by solving $O(n)$ CVP problems of rank $n-1$, each of which can be solved by querying the rank n SIVP oracle using the reduction shown in §4.2. We omit the details here, but the full proof can be found in [Mic08]. Having found polynomial time rank-preserving reductions in both directions, we conclude that

Theorem 4.13. *CVP and SIVP are equivalent under rank-preserving reductions.*

5 Exponential Time Hardness

Although showing that SVP, CVP, and SIVP are NP-complete is useful because it shows that these problems cannot be solved in polynomial time unless P = NP, this is generally not a strong enough result for the security of a cryptosystem to rely on. This is because there is a vast difference in running times of non-polynomial algorithms. For example, a quasi-polynomial time algorithm for a problem might be able to run fast enough that a cryptosystem whose security relies on the fact that no polynomial time algorithm can solve that problem is insecure. Rather, we would prefer to be able to show that the hard problem on which the security of our cryptosystem is based cannot be solved in subexponential time.

In this section, I present my own exponential time hardness results for CVP and SIVP in all l_p norms and for SVP in the l_∞ norm. When we say that a problem can be solved in subexponential time, we mean that there is an $O(2^{\varepsilon n})$ time algorithm for the problem for all $\varepsilon > 0$. In order to show exponential time hardness of various lattice problems, we will assume the exponential time hypothesis (ETH), the statement that 3SAT cannot be solved in subexponential time [IP01]. In the following sections, we will show that various lattice problems are ETH-hard, meaning that they cannot be solved in subexponential time unless the ETH is false. We will do this by reducing 3SAT (an ETH-hard problem by definition) to these problems. One thing to note is that for our reductions to be valid ETH-hardness reductions, a 3SAT instance of size n must map to an instance of the lattice problem of size $O(n)$ or subexponentially many such instances. Otherwise, the reduction will not show ETH-hardness. For example, if we map a size n 3SAT instance to a size n^2 instance of another problem, we could have a $2^{O(n)}$ time algorithm for that problem, which would be subexponential in the size of the instance of that problem, but would not allow us to solve 3SAT in subexponential time. One last point is that it is not entirely clear how to determine the size of a 3SAT instance. Should it be based on n (the number of variables), m (the number of clauses), or some linear combination of the two? Fortunately, this question has been resolved by the sparsification lemma shown in [IPZ01], which shows that any 3SAT formula can be solved by solving $O(2^{\varepsilon n})$ 3SAT formulas for which $m = O(n)$ for any $\varepsilon > 0$. Therefore, it follows that a reduction from 3SAT is valid for showing ETH-hardness if it maps 3SAT instances to instances of our problem of size $O(n + m)$. In this section, we present our results for ETH-hardness of various lattice problems.

5.1 ETH-hardness of CVP

Based on what we have previously shown, we can now prove ETH-hardness of CVP. In [Sip12], the reduction from 3SAT to subset sum to show that subset sum is NP-complete mapped a 3SAT instance to a subset sum instance of size $O(n+m)$. Similarly, the reduction from subset sum to CVP shown in the proof of Thm. 4.2 mapped an n dimensional subset sum instance to an $n + 1$ dimensional CVP instance. Composing these two reductions, we have a reduction from 3SAT to CVP that maps 3SAT instances with n variables and m clauses to CVP instances of size $O(n + m)$. This is enough to show the following result.

Theorem 5.1 (ETH-hardness of CVP)**.** *For any $p \geq 1$, CVP is ETH-hard in the l_p norm.*

Proof. Suppose that CVP can be solved in subexponential time, meaning that it can be

solved in time $O(2^{\varepsilon n})$ time for all $\varepsilon > 0$. Then, to solve 3SAT in $O(2^{\delta n})$ time for any fixed $\delta > 0$, simply apply the sparsification lemma and the reduction to CVP to solve 3SAT in $O(2^{\varepsilon n} \cdot 2^{\varepsilon' O(n)}) = O(2^{\delta n})$ time for sufficiently small $\varepsilon, \varepsilon'$. So, this implies 3SAT is solvable in subexponential time, contradicting the exponential time hypothesis, and so we conclude that CVP cannot be solved in subexponential time under the ETH. □

5.2 ETH-hardness of SIVP

From the ETH-hardness of CVP and our reduction from CVP to SIVP that maps an n dimensional CVP instance to an $n+1$ dimensional SIVP instance, we can easily show the ETH-hardness of SIVP.

Theorem 5.2 (ETH-hardness of SIVP)**.** *For any $p \geq 1$, SIVP is ETH-hard in the l_p norm.*

Proof. Composing the reduction from 3SAT to subset sum, the reduction from Thm. 4.2, and the reduction from Thm. 4.3, we have arrived at a reduction from 3SAT to SIVP that maps 3SAT instances of n variables and m clauses to SIVP instances of size $O(n+m)$. Therefore, by the same reasoning as in the proof of Thm. 5.1, it follows that SIVP is ETH-hard. □

5.3 ETH-hardness of SVP in the l_∞ norm

While showing ETH-hardness of CVP was very straightforward, showing the ETH-hardness of SVP is much harder. The general difficulty comes from the fact that our short vector must be nonzero, and it is difficult when applying these reductions to guarantee that this will be the case. So far, I have only been able to show ETH-hardness of SVP in the l_∞ norm. This is done by a series of reductions which when composed, map 3SAT instances with n variables and m clauses to SVP_{l_∞} instances of size $O(n+m)$. We now lay out these various reductions.

5.3.1 3SAT $\leq$ PARTITION

PARTITION is defined as follows.

Definition 5.3 (PARTITION problem)**.** Given a finite multiset S of positive integers, it is a YES instance of PARTITION if it is possible to find a subset $S' \subseteq S$ such that $\sum_{a \in S'} a = \sum_{a \in S \setminus S'} a$ and a NO instance otherwise.

The reduction we use here is well known, and I arrived at it by simply adapting the reduction from 3SAT to subset sum in [Sip12]. The general idea is that we view the 3SAT instance as a bunch of different clauses of 3 variables that all must be satisfied. To map this to a partition instance, we consider numbers base 10 and have one digit of our base 10 numbers correspond to each of the clauses in the 3SAT instance. Additionally, we have a digit in our base 10 numbers that correspond to each variable in the 3SAT instance. The numbers in S are then base 10 numbers whose digits are all either 1's or 0's, and the 1's are placed in the indices corresponding to a particular variable and the clauses it occurs in

(or its negation depending on whether we are setting the variable to true or false). We then have the constraint that each variable can only be assigned one value (true or false) and that each clause must contain at least one true value (out of its 3 values). Obviously, this must be made more precise, and we present the formal reduction and proof below.

Proposition 5.4. *There is a polynomial time reduction from 3SAT to PARTITION that maps a 3SAT instance with n variables and m clauses to a PARTITION instance of size $2(n+m)+1 = O(n+m)$.*

Proof. Given a 3SAT formula ϕ with n variables and m clauses, we map ϕ to a PARTITION instance as follows. Every number we add to our partition instance can be expressed using $n+m$ digits in base 10. We write the number in base 10 as $v_1 v_2 v_3 \dots v_n c_1 c_2 \dots c_m$ where the v_i's correspond to the variables in ϕ and the c_j's correspond to the clauses in ϕ. For each variable x_i in ϕ, we add two numbers, t_i and f_i, to our PARTITION instance. t_i is defined as follows. For $1 \leq j \leq n$, $v_j = 1$ if $j = i$ and is 0 otherwise. For $1 \leq j \leq m$, $c_j = 1$ if x_i is in the jth clause and is 0 otherwise. Similarly, f_i is defined with $v_j = 1$ if $j = i$ and 0 if $j \neq i$ and with $c_j = 1$ if $\bar{x}_i$ is in the jth clause and 0 otherwise. Additionally, for every clause w_j in ϕ, we add y_j and z_j to our PARTITION instance where $y_j = z_j$ are defined by setting $c_j = 1$ and all other indices 0. Finally, we add the number s to our PARTITION instance where s is defined by setting $v_i = 0$ for all i and $c_j = 1$ for all j.

As an example, suppose we had the 3SAT formula $\phi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3)$. Then, this would map to the PARTITION instance S where S contains the following positive integers.

t_1	10010
f_1	10001
t_2	01001
f_2	01010
t_3	00111
f_3	00100
y_1	00010
z_1	00010
y_2	00001
z_2	00001
s	00011

($\implies$) Suppose ϕ is satisfiable. Let $x_1, \dots, x_n$ be the satisfying assignment. Then, define the subset S' of S as follows. $t_i \in S'$ if x_i is true and $f_i \in S'$ if x_i is false. $y_j \in S'$ if the number of true literals in the jth clause is ≤ 2. $z_j \in S'$ if the number of true literals in the jth clause is 1. We note that $\sum_{a \in S'} a$ has $v_i = 1$ for all i and $c_j = 3$ for all j. For example, if ϕ had 3 variables and 4 clauses, we would have $\sum_{a \in S'} a = 1113333$. We see this by the following. First, we note that there are never any carries when we sum numbers in S since there are 2 elements of S with $v_i = 1$ for each i and $v_i = 0$ for all other elements of S. These elements are the t_i and f_i associated with v_i. Similarly, there are only 6 elements in S with $c_j = 1$ for each j and $c_j = 0$ for all other elements. This follows since there are 3 elements with $c_j = 1$ that correspond to the three literals in the jth clause, there are y_j and

z_j corresponding to the jth clause, and the number s, giving a total of 6 elements in S with $c_j = 1$. So, it follows that $\sum_{a \in S} a = 2226666$ ($v_i = 2$ for all i and $c_j = 6$ for all j). Now, consider S'. For each variable x_i, exactly one of t_i and f_i is in S'. This implies that $v_i = 1$ for all i in $\sum_{a \in S'} a$. Additionally, for each clause w_j, there is a 1 in c_j for each true literal in w_j given our satisfying assignment. So, at least 1 of the literals in w_j must be true and we include y_j and z_j in S' such that $c_j = 3$ for all j in $\sum_{a \in S'} a$. Therefore, it follows that $\sum_{a \in S'} a = 1113333$ ($v_i = 1$ for all i and $c_j = 3$ for all j). Since $\sum_{a \in S} a = 2226666$, it follows that $\sum_{a \in S'} a = \sum_{a \in S \setminus S'} a = 1113333$ and PARTITION is satisfied.

($\Longleftarrow$) Conversely, suppose that PARTITION is satisfied and we have some $S' \subseteq S$ such that $\sum_{a \in S'} a = \sum_{a \in S \setminus S'} a$. Since $\sum_{a \in S} = 2226666$ as shown above, it follows that $\sum_{a \in S'} a = 1113333$. WLOG, suppose that S' does not contain s. This can be done since we can simply relabel S' and $S \setminus S'$ to ensure this condition is satisfied. Now, let x_i be true if $t_i \in S'$ and x_i be false if $f_i \in S'$. Since $v_i = 1$ in $\sum_{a \in S'} a$ for all i, it follows that exactly one of t_i and f_i is in S' so the assignment is well defined. Additionally, since $c_j = 3$ in $\sum_{a \in S'} a$ for all j, it follows that every clause is satisfied since at least one of the literals must be true for $c_j = 3$ since s is not in S'. Therefore, it follows that ϕ is satisfied as desired.

So, since this mapping can be done in polynomial time, we have reduced 3SAT to PARTITION and the mapping takes a 3SAT formula with n variables and m clauses and maps it to a PARTITION problem where the size of S is $O(n + m)$. □

5.3.2 PARTITION $\leq$ WEAK-PARTITION

WEAK-PARTITION is defined as follows.

Definition 5.5 (WEAK-PARTITION)**.** Given a finite vector $(a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n$, it is a YES instance of WEAK-PARTITION if there a solution to $\sum_{i=1}^{n} x_i a_i = 0$ with $x_i \in \{-1, 0, 1\}$ and not all $x_i = 0$ and a NO instance otherwise.

As we can see, WEAK-PARTITION is a version of the PARTITION problem where we allow some (but not all) of the entries in our vector to be excluded. To show this reduction, we follow the proof of van Emde Boas in [vEB81]. The general idea is to take the PARTITION instance and put it inside a WEAK-PARTITION problem where we have added an additional system of equations that must be satisfied. In order for the WEAK-PARTITION instance to be satisfiable, the original PARTITION instance must be satisfiable in our WEAK-PARTITION setting where we also allow 0 as a coefficient, and the additional system of equations must be satisfiable. However, we will show that this system of equations holds the property that it is satisfiable if and only if the coefficients of the a_i's in our solution to the PARTITION instance are all either -1 or 1, meaning that they are a valid solution to the PARTITION problem. Formally, we have the following.

Proposition 5.6. *There is a polynomial time reduction from PARTITION to WEAK-PARTITION that maps a PARTITION instance of size n to a WEAK-PARTITION instance of size $5n - 1 = O(n)$.*

Proof. Suppose we are given a PARTITION instance S. Express S as a vector $(a_1, a_2, \ldots, a_n) \in$

$\mathbb{Z}_{>0}{}^n$ by arbitrarily ordering the elements. Then, for each a_i map it to 5 b_{ij}'s defined as follows.

$$\begin{aligned}
b_{i1} &= a_i + N(10^{4i-4} + 10^{4i-3} + 0 + 10^{4i-1} + 0) \\
b_{i2} &= 0 + N(0 + 10^{4i-3} + 0 + 0 + 10^{4i}) \\
b_{i3} &= 0 + N(10^{4i-4} + 0 + 10^{4i-2} + 0 + 0) \\
b_{i4} &= a_i + N(0 + 0 + 10^{4i-2} + 10^{4i-1} + 10^{4i}) \\
b_{i5} &= 0 + N(0 + 0 + 0 + 10^{4i-1} + 0)
\end{aligned}$$

where $N = 1 + 2\sum_{i=1}^{n} a_i$. However, define b_{n2} and b_{n4} as above, but replace the $N10^{4n}$ term with N so that

$$\begin{aligned}
b_{n2} &= N(1 + 10^{4n-3}) \\
b_{n4} &= a_i + N(1 + 10^{4n-2} + 10^{4n-1})
\end{aligned}$$

We will convert the condition that $\sum_{i,j} x_{ij} b_{ij} = 0$ for $x_{ij} \in \{-1, 0, 1\}$ into a system of equations by repeated application of the following lemma. Suppose that we can write each a_i in the form $a_i = b_i + Nc_i$ and $N > \sum_i |b_i|$. Then, $\sum_i x_i a_i = 0$ with $x_i \in \{-1, 0, 1\}$ is equivalent to the conditions $\sum_i x_i b_i = 0$ and $\sum_i x_i c_i = 0$ for $x_i \in \{-1, 0, 1\}$. To see this, simply divide $\sum_i x_i a_i$ by N and note that both the integral and fractional parts must sum to 0. Since $N > \sum_i |b_i|$, the fractional part corresponds to $\sum_i x_i b_i$ and the integral part corresponds to $\sum_i x_i c_i$.

So, since $N = 1 + 2\sum_{i=1}^{n} a_i > 2\sum_{i=1}^{n} |a_i|$ since all the a_i's are positive, by repeated application of the above lemma, we see that the condition $\sum_{i,j} x_{ij} b_{ij} = 0$ for $x_{ij} \in \{-1, 0, 1\}$ is equivalent to the following system of equations

$$\begin{aligned}
\sum_{i=1}^{n} (x_{i1} + x_{i4}) a_i &= 0 \\
x_{i1} + x_{i3} + x_{i-1,2} + x_{i-1,4} &= 0 \quad \text{for } i = 2, \ldots, n \\
x_{11} + x_{13} + x_{n2} + x_{n4} &= 0 \\
x_{i1} + x_{i2} &= 0 \quad \text{for } i = 1, 2, \ldots, n \\
x_{i3} + x_{i4} &= 0 \quad \text{for } i = 1, 2, \ldots, n \\
x_{i1} + x_{i4} + x_{i5} &= 0 \quad \text{for } i = 1, 2, \ldots, n
\end{aligned}$$

for $x_{ij} \in \{-1, 0, 1\}$. In order for this system of equations to have a solution, we must have that $x_{i1} = -x_{i2}$ and $x_{i3} = -x_{i4}$ for all i. Additionally, $x_{i1} + x_{i3} = -(x_{i-1,2} + x_{i-1,4})$ for $i = 2, \ldots, n$ and $x_{11} + x_{13} = -(x_{n2} + x_{n4})$. Combining these conditions give us the condition that $x_{i1} + x_{i3}$ is a constant independent of i. Now, suppose that we remove b_{13} from the set integers in our WEAK-PARTITION instance. This effectively adds the requirement that $x_{13} = 0$. So, this means that $x_{11} + x_{13} = x_{11} \in \{-1, 0, 1\}$ and so $x_{i1} + x_{i3} \in \{-1, 0, 1\}$ for all i. WLOG, we can assume that $x_{i1} + x_{i3} \in \{0, 1\}$ since we can multiply any solution to the system of equations by -1 and still have a valid solution.

Suppose that $x_{i1} + x_{i3} = 0$. Combining two of the equations, we note that $x_{i5} = x_{i3} - x_{i1}$ for all i. Since $x_{i5} \in \{-1, 0, 1\}$ and $x_{i1} + x_{i3} = 0$, this implies that $x_{i5} = 0$ for all i. This then gives us the following equations

$$\begin{aligned} x_{i1} + x_{i3} &= 0 \\ x_{i3} + x_{i4} &= 0 \\ x_{i1} + x_{i4} &= 0 \end{aligned}$$

which implies that $x_{i1}, x_{i3}, x_{i4} = 0$ for all i, which implies that $x_{i2} = 0$ and so the only possible solution to this system of equations is the all 0 solution, which is not a valid solution to the WEAK-PARTITION problem.

So, for a solution to exist, there must be a solution with $x_{i1}+x_{i3} = 1$. Since $x_{i5} = x_{i3}-x_{i1}$, this implies that $x_{i5} = \pm 1$. If we solve the system of equations for both possible values of x_{i5}, we find that for each i, the values of the x_{ij}'s must be one of the following

$$x_{i1} = 0,\ x_{i2} = 0,\ x_{i3} = 1,\ x_{i4} = -1,\ x_{i5} = 1$$
$$x_{i1} = 1,\ x_{i2} = -1,\ x_{i3} = 0,\ x_{i4} = 0,\ x_{i5} = -1$$

In particular, for every i, $x_{i1} + x_{i4} \in \{-1, 1\}$. So, this implies that any solution to the WEAK-PARTITION instance gives a solution to $\sum_i y_i a_i = 0$ with $y_i \in \{-1, 1\}$ by setting $y_i = x_{i1} + x_{i4}$. By taking S' to be the elements of S for which the corresponding $y_i = 1$, we arrive at a solution to PARTITION. In the other direction, if we take a solvable PARTITION instance S and map it in the above manner to a WEAK-PARTITION instance, we can find a solution to the WEAK-PARTITION instance by setting

$$x_{i1} = 1,\ x_{i2} = -1,\ x_{i3} = 0,\ x_{i4} = 0,\ x_{i5} = -1$$

if the corresponding a_i is in S' and by setting

$$x_{i1} = 0,\ x_{i2} = 0,\ x_{i3} = 1,\ x_{i4} = -1,\ x_{i5} = 1$$

otherwise. So, we have found a polynomial time reduction from PARTITION to WEAK-PARTITION that maps a PARTITION instance of size n to a WEAK-PARTITION instance of size $5n - 1 = O(n)$. □

5.3.3 WEAK-PARTITION $\leq$ SVP$_{l_\infty}$

The final reduction in our series of reductions to show ETH-hardness of SVP$_{l_\infty}$ is the reduction from WEAK-PARTITION to SVP$_{l_\infty}$. The idea behind this reduction is that in the l_∞ norm, we can view a bound on the norm of a vector as a bound on the absolute value of each of its entries. Suppose we set our distance parameter to be $r = 1$. If the basis matrix of our lattice contains the identity matrix as a submatrix, this ensures that all the coefficients of our basis vectors in any lattice vector with norm ≤ 1 are in $\{-1, 0, 1\}$. If we then add a row

to our lattice matrix that corresponds to the WEAK-PARTITION instance, but scale it by some factor $\alpha > 1$, then this ensures that the lattice can only have a vector with norm ≤ 1 if there is a solution to the WEAK-PARTITION problem since the identity matrix ensures that the x_i's are in $\{-1, 0, 1\}$ and the fact that $\alpha > 1$ ensures $\sum_i x_i a_i = 0$. Expanding on this idea, we arrive at the following proposition.

Proposition 5.7. *There is a polynomial time reduction from WEAK-PARTITION to SVP_{l_∞} that maps a WEAK-PARTITION instance of size n to a SVP_{l_∞} instance of size $n+1 = O(n)$.*

Proof. Suppose we are given $(a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n$ as a WEAK-PARTITION instance. We map this to the SVP_{l_∞} instance given by the lattice basis

$$\mathbf{B} = \begin{pmatrix} \alpha a_1 & \alpha a_2 & \ldots & \alpha a_n \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix}$$

and distance $r = 1$ where $\alpha > 1$.

($\Longrightarrow$) Now, suppose that $(a_1, a_2, \ldots, a_n) \in \mathbb{Z}^n$ is a YES instance of WEAK-PARTITION, so there exists $x_i \in \{-1, 0, 1\}$ such that $\sum x_i a_i = 0$ and not all $x_i = 0$. Then, if we take $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, we have that $\mathbf{Bx} = (0, \mathbf{x})^T$ and so $\|\mathbf{Bx}\|_\infty = 1$ implying that $(\mathbf{B}, 1)$ is a YES instance of SVP_{l_∞}.

($\Longleftarrow$) Conversely, suppose that $(\mathbf{B}, 1)$ is a YES instance of SVP_{l_∞}. Then, there exists some $\mathbf{x} \neq \mathbf{0} \in \mathbb{Z}^n$ such that $\|\mathbf{Bx}\|_\infty \leq 1$. This implies that $|x_i| \leq 1$ for all x_i and since each $x_i \in \mathbb{Z}$, this implies that $x_i \in \{-1, 0, 1\}$. Additionally, we have that $|\alpha(\sum x_i a_i)| \leq 1$ but since $\alpha > 1$, this is only possible if $\sum x_i a_i = 0$, implying that $(a_1, a_2, \ldots, a_n)$ is a YES instance of WEAK-PARTITION.

So, we have found a polynomial time reduction from WEAK-PARTITION to SVP_{l_∞} that maps a WEAK-PARTITION instance of size n to an SVP_{l_∞} instance of size $O(n)$. □

From the above reductions, we arrive at the following conclusion.

Theorem 5.8 (SVP_{l_∞} **hardness of ETH).** *SVP is ETH-hard in the l_∞ norm.*

Proof. Composing the reductions, we have found a polynomial time reduction from 3SAT to SVP_{l_∞} that maps a 3SAT instance of n variables and m clauses to an SVP_{l_∞} instance of size $O(n + m)$. Following the same reasoning as in the proof of Thm. 5.1, we conclude that SVP_{l_∞} is ETH-hard. □

6 Lattice Based Cryptography

Now that we have established some hardness results for lattice problems, we will examine their application to cryptography. Lattice based cryptography is particularly attractive because it is conjectured to be resistant to quantum attacks. Additionally, lattice based cryptography has the remarkable property that its security is able to be based on the worst-case hardness of lattice problems. In cryptography, we sample cryptographic objects from some random distribution, so in order for a scheme to be secure, it is necessary for the scheme to be hard to break in the average-case. However, in [Ajt96], Ajtai was able to show that certain problems are average-case hard provided that other lattice problems are worst-case hard. All the NP and ETH-hardness results we showed in the previous sections are worst-case hardness results for lattice problems, but using Ajtai's result, we will be able to use these worst-case hardness results to reason about the average-case hardness of other problems upon which lattice based cryptosystems are based. Finally, lattice based cryptosystems have some amazing properties. For example, it is possible to construct fully homomorphic encryption schemes over lattices, which are encryption schemes that support arbitrary computation on the ciphertexts (addition and multiplication). In fact, the first candidate construction for fully homomorphic encryption due to Gentry was based on ideal lattices [Gen09]. As such, lattices are fundamental to modern cryptography, and with the possibility of quantum computers, they may one day become the gold standard and be used on many devices all over the internet. In this section, we will merely skim the surface of lattice based cryptography by detailing one encryption scheme based on lattices due to Regev that is based on the learning with errors problem [Reg05].

6.1 Cryptosystems

In order to describe a cryptosystem over lattices, we must first formalize what we mean by an encryption scheme. There are two main types of encryption: private key and public key encryption. In private key encryption, there is a secret key that is shared between two parties that wish to communicate privately. A new secret key must be generated and shared between any pair of parties that wish to communicate. In contrast, in a public key encryption system, there are two components of the key, the public key and the secret key. The main point here is that each individual has their own public and secret key pair and the public key is available to anyone while the secret key is kept hidden by the individual. Then, if party A wants to send a message to party B, A looks up B's public key and encrypts the message under B's public key. In a public key encryption scheme, this encrypted message can only be decrypted if one possesses B's secret key. A main advantage of public key encryption is that any two parties can send messages to each other without having to generate a secret key that is shared between them. Instead, each individual has their own secret key, but anybody can send them a message by encrypting using their public key. Therefore, public key encryption schemes are much more desirable and practical, and the encryption scheme we will study in this section is a public key encryption scheme.

Formally, according to [KL07], a public key encryption scheme is a tuple of probabilistic polynomial time algorithms $(\text{Gen}, \text{Enc}, \text{Dec})$ that are defined as follows. Gen is the key generation algorithm and takes in a security parameter 1^n and outputs a tuple (pk, sk)

where pk is the public key and sk is the secret key. Enc is the encryption algorithm and takes as input a public key pk and a message m from some message space $\mathcal{M}$ and outputs a ciphertext c in some ciphertext space $\mathcal{C}$. In order to obtain the correct definition of security, which we will discuss later, it is required that Enc is probabilistic. Dec is the decryption algorithm and it takes a ciphertext $c \in \mathcal{C}$ along with a secret key sk and outputs a message $m \in \mathcal{M}$ or a symbol $\perp$ that signals that decryption failed. Usually, Dec is deterministic and for the purposes of this section, we will assume that it is. Finally, for the correctness of the encryption scheme, there must exist a negligible function f such that for all n, all (pk, sk) output by $\text{Gen}(1^n)$, and all $m \in \mathcal{M}$,

$$\Pr[\text{Dec}(sk, \text{Enc}(pk, m)) \neq m] \leq f(n)\,,$$

where a negligible function is a function such that for every polynomial p, there exists an N such that for all $n > N$, $f(n) < 1/p(n)$. Intuitively, we can think of a negligible function as a function that for sufficiently large n is less than one over any polynomial. The above correctness statement is simply that any encryption of a message will successfully decrypt to that message with all but negligible probability.

Now, we need to come up with some notion of security for a public key encryption system. Informally, we want to say that no probabilistic polynomial time algorithm given a ciphertext c and a public key pk can successfully determine the message m such that $\text{Dec}(sk, c) = m$ where sk is the secret key associated with the public key pk. We formalize this notion via an indistinguishability game that the adversary plays. First, for a given security parameter n, $\text{Gen}(1^n)$ is run to obtain a public key, secret key pair (pk, sk). We then give the adversary A the public key pk and A outputs two messages m_0 and m_1. A random bit b is then chosen uniformly at random and the ciphertext $\text{Enc}(pk, m_b) = c$ is given to A. A computes with pk and c and eventually outputs an answer bit b_a. If $b_a = b$, then A has succeeded and otherwise A has failed the test. We now say that a public key encryption scheme is secure against a chosen plaintext attack if for all probabilistic polynomial time algorithms A, there is some negligible function f such that the probability that A succeeds at the above game is $\leq \frac{1}{2} + f(n)$. The reason we use the terminology *chosen plaintext attack* comes from the fact that since A has the public key pk, it can choose any message and compute as many encryptions of it as it wants in polynomial time. Additionally, because A is able to compute the encryption of any message, we note that Enc must be probabilistic for our public key encryption scheme to have any chance of satisfying this notion of security since otherwise A could simply compute $\text{Enc}(pk, m_0)$ and $\text{Enc}(pk, m_1)$ and see which one evaluated to c.

6.2 The Learning with Errors Problem

The security of the cryptosystem presented in [Reg05] is based off the learning with errors problem, which is defined as follows.

Definition 6.1 (Learning With Errors Problem)**.** Let $q \geq 2$ be some integer and let χ be a probability distribution over $\mathbb{Z}_q$, the integers modulo q. For some fixed vector $\mathbf{s} \in \mathbb{Z}_q^n$, let $A_{\mathbf{s},\chi}$ be the distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ given by $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ where $\mathbf{a}$ is sampled from $\mathbb{Z}_q^n$ uniformly at random and e is sampled according to χ and $\langle \mathbf{a}, \mathbf{s} \rangle$ denotes the standard inner product of $\mathbf{a}$ and $\mathbf{s}$. The learning with errors problem is the problem of distinguishing $A_{\mathbf{s},\chi}$

for some $\mathbf{s}$ sampled uniformly at random from $\mathbb{Z}_q^n$ from the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ with nonnegligible probability.

The learning with errors problem (LWE) is used frequently in cryptographic constructions based on lattices. The reason for this is that LWE is hard given that GAPSIVP$_\gamma$ or GAPSVP$_\gamma$ is hard. Here, GAPSVP$_\gamma$ denotes the promise version of SVP where YES instances are lattices Λ where $\lambda_1(\Lambda) \leq r$ for some distance parameter $r > 0$ and NO instances are lattices Λ where $\lambda_1(\Lambda) > \gamma r$. GAPSIVP$_\gamma$ is defined analogously. Additionally, the LWE problem naturally lends itself to cryptographic applications since it has a "secret" vector, $\mathbf{s}$ embedded in the problem definition.

The learning with errors problem is parametrized by q and χ. q is the integer we use in $\mathbb{Z}_q$ in the construction and χ is the error distribution. Typically, the error distribution χ is what is known as a discrete Gaussian distribution. This can be briefly defined as follows [GPV07].

Definition 6.2 (Discrete Gaussian Distribution)**.** Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ denote the reals modulo 1 on the interval $[0, 1)$ with addition modulo 1. Let Ψ_α denote the distribution on $\mathbb{T}$ of a Gaussian random variable with mean 0 and variance $\frac{\alpha^2}{2\pi}$. Given an integer q, let its discretization $\bar{\Psi}_\alpha$ be defined as the distribution of

$$\lfloor (q \cdot X_{\Psi_\alpha}) + 1/2 \rfloor \bmod q\,,$$

where X_{Ψ_α} is a random variable with distribution Ψ_α.

Examining the definition above, we see that Ψ_α can be thought of as the distribution we arrive at when we sample a value from a Gaussian random variable with mean 0 and variance $\frac{\alpha^2}{2\pi}$ and then reduce it modulo 1 so that we arrive at a value in $[0, 1)$. To obtain the discretization $\bar{\Psi}_\alpha$, we take the original continuous distribution defined on $[0, 1)$ and scale it by q so that it lies in $[0, q)$. We then map every value in this continuous distribution on $[0, q)$ to the integer in $\{0, 1, 2, \ldots, q-1\}$ that it is closest to in $\mathbb{Z}_q$.

We are now ready to state the main result of Regev in [Reg05] that established the hardness connection between LWE and GAPSVP$_\gamma$ and GAPSIVP$_\gamma$.

Theorem 6.3 (Hardness of LWE)**.** *Let $n, q \in \mathbb{Z}$ and let $\alpha \in (0, 1)$ be such that $\alpha q > 2\sqrt{n}$. If there exists an efficient algorithm that solves LWE with parameters $q, \bar{\Psi}_\alpha$, then there exists an efficient quantum algorithm that solves GAPSVP$_\gamma$ and GAPSIVP$_\gamma$ in the worst case with approximation factor $\gamma = \tilde{O}(n/\alpha)$.*

The full proof is extremely lengthy and is beyond the scope of this thesis, but the details can be found in [Reg05]. The most important takeaway from this theorem is that cryptosystems based on LWE are secure provided that there are no polynomial time quantum algorithms for GAPSVP$_\gamma$ or GAPSIVP$_\gamma$ with approximation factor $\gamma = \tilde{O}(n/\alpha)$.

Even though there are no known quantum algorithms for these lattice problems that perform significantly better than classical algorithms [Pei16], it would be more comfortable for the reduction to be a classical one so that the security of LWE is implied by the nonexistence of classical algorithms for these lattice problems. Fortunately, Peikert gave a classical version of Regev's LWE hardness result in [Pei09] and so the concern of using a quantum reduction has been eliminated.

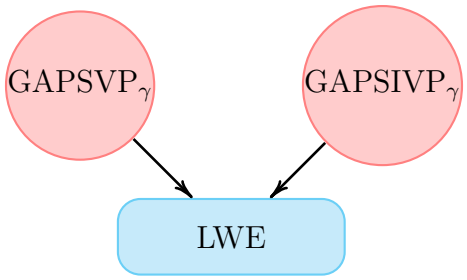

Figure 5: Known reductions between lattice problems and LWE for $\gamma = \tilde{O}(n/\alpha)$.

6.3 A Public Key Cryptosystem on Lattices

We are now ready to go over our first public key cryptosystem based on lattices, which is presented in [Reg05]. This cryptosystem makes use of a discrete Gaussian distribution that was defined in Def. 6.2. Let n be the security parameter of the cryptosystem. Then, pick p to be some prime with $n^2 \leq p \leq 2n^2$ and set

$$m = (1+\varepsilon)(n+1)\log p$$

for some small and arbitrary constant $\varepsilon > 0$. Next, let χ be the probability distribution given by the discretization of Ψ_α over $\mathbb{Z}_p$ for $\alpha = o(1/(\sqrt{n}\log n))$. Our message space $\mathcal{M} = \{0,1\}$ and our ciphertext space $\mathcal{C} = \mathbb{Z}_p^{n+1}$. The tuple of probabilistic polynomial time algorithms $(\text{Gen}, \text{Enc}, \text{Dec})$ are defined as follows:

Gen

For the secret key sk, Gen chooses a vector $\mathbf{s}$ uniformly at random over $\mathbb{Z}_p^n$ and makes that the secret key.

For the public key pk, Gen generates m vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ independently and uniformly at random over $\mathbb{Z}_p^n$. It then generates m error values $e_1, \ldots, e_m$ each in $\mathbb{Z}_p$ independently according to χ. Using the secret key, it computes m values $b_1, \ldots, b_m \in \mathbb{Z}_p$ by setting $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i$. The public key is then the matrix

$$\begin{pmatrix} \mathbf{a}_1 & b_1 \\ \mathbf{a}_2 & b_2 \\ \vdots & \vdots \\ \mathbf{a}_m & b_m \end{pmatrix}.$$

Enc

To encrypt a message $m' \in \{0,1\}$, we first select a subset $S \subseteq \{1, 2, \ldots, m\}$ uniformly at random over the 2^m possible subsets. If $m' = 0$, its encryption is then

$$\left(\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i \right),$$

and if $m' = 1$, its encryption is

$$\left(\sum_{i \in S} \mathbf{a}_i, \lfloor \frac{p}{2} \rfloor + \sum_{i \in S} b_i \right) ,$$

where the arithmetic is carried out over the group $\mathbb{Z}_p$.

Dec

To decrypt a ciphertext $c \in \mathbb{Z}_p^{n+1}$, we let $c = (\mathbf{a}, b)$ with $\mathbf{a} \in \mathbb{Z}_p^n$ and $b \in \mathbb{Z}_p$ and then compute $b' = b - \langle \mathbf{a}, \mathbf{s} \rangle$. If b' is closer to 0 than $\lfloor \frac{p}{2} \rfloor$, we decrypt c to 0 and if it is closer to $\lfloor \frac{p}{2} \rfloor$ than to 0, we decrypt c to 1. If b' is the same distance from both values, we output the failure symbol $\perp$.

One thing to immediately note about this cryptosystem is that its security must be based on the LWE assumption since the public key is m independent copies of $A_{\mathbf{s},\chi}$ and the way a bit is encrypted is the same up to adding $\lfloor \frac{p}{2} \rfloor$ to $\sum_{i \in S} b_i$. Essentially, the LWE assumption states that it should be difficult to distinguish the b_i's from uniformly random elements of $\mathbb{Z}_p$, so it makes sense that one would not be able to distinguish $\lfloor \frac{p}{2} \rfloor + \sum_{i \in S} b_i$ from $\sum_{i \in S} b_i$.

Now, we will sketch the proofs of correctness and security for this cryptosystem as detailed in [Reg05]. We use the notation χ^k to represent the sum of k independent samples from χ. We will show for our choice of parameters m, p, α that there exists some negligible function $\delta(n)$ such that for all $k \in \{0, 1, \ldots, m\}$,

$$\Pr[|e| < \lfloor \frac{p}{2} \rfloor / 2] > 1 - \delta(n) ,$$

where the probability is taken over e sampled from $\bar{\Psi}_\alpha^k$ and $|e|$ is defined as $\min\{e, p - e\}$, the distance of e from 0.

The proof of this statement follows by noting that a sample from $\bar{\Psi}_\alpha^k$ can be obtained by sampling from Ψ_α k times independently to obtain $e_1, \ldots, e_k$ and outputting $\sum_{i=1}^k \lfloor pe_i + 1/2 \rfloor \bmod p$. Based on the choice of parameters, one can show that this output distribution varies from $\sum_{i=1}^k pe_i \bmod p$ by at most $p/32$. Therefore, if we can show that

$$\left| \sum_{i=1}^k pe_i \bmod p \right| < \frac{p}{16} ,$$

with high probability then the statement will follow since we will have that

$$|e| < \frac{p}{16} + \frac{p}{32} < \frac{p}{8} < \frac{\lfloor p/2 \rfloor}{2} ,$$

as desired. We note that $|\sum_{i=1}^k pe_i \bmod p| < \frac{p}{16} \iff |\sum_{i=1}^k e_i \bmod 1| < \frac{1}{16}$ and that $|\sum_{i=1}^k e_i \bmod 1|$ is distributed the same as a random sample from $\Psi_{\sqrt{k}\alpha}$ since it is the sum of k independent Gaussian random variables. From our choice of parameters, we have that $\sqrt{k} \cdot \alpha = o(\frac{1}{\sqrt{\log n}})$ since $k \leq m$ and so as n becomes large, the probability that a random sample from $\Psi_{\sqrt{k}\alpha}$ is less than some fixed constant ($\frac{1}{16}$ in our case) is $> 1 - \delta(n)$ for some negligible function $\delta(n)$.

Using the above lemma, we can now show the correctness of the cryptosystem, namely the fact that

$$\Pr[\mathrm{Dec}(sk, \mathrm{Enc}(pk, m)) \neq m] \leq f(n)\,,$$

for some negligible function $f(n)$. Suppose we are given the encryption of a message $m' \in \{0,1\}$. Then, the ciphertext is of the form $(\mathbf{a}, \lfloor \frac{p}{2} \rfloor \cdot m' + b)$ where $\mathbf{a} = \sum_{i \in S} \mathbf{a}_i$ and $b = \sum_{i \in S} b_i$ for some $S \subseteq \{1, 2, \ldots, m\}$. Now, we have that

$$\begin{aligned} b - \langle \mathbf{a}, \mathbf{s} \rangle &= \sum_{i \in S} b_i - \langle \sum_{i \in S} \mathbf{a}_i, \mathbf{s} \rangle \\ &= \sum_{i \in S} b_i - \sum_{i \in S} \langle \mathbf{a}_i, \mathbf{s} \rangle \\ &= \sum_{i \in S} e_i\,. \end{aligned}$$

Then, by the above lemma $|\sum_{i \in S} e_i| < \lfloor \frac{p}{2} \rfloor / 2$ with all but negligible probability, and so if $m' = 0$, $|\sum_{i \in S} e_i|$ will be closer to 0 than $\lfloor \frac{p}{2} \rfloor$ with all but negligible probability and we decrypt to 0, and by symmetry if $m' = 1$, $|\lfloor \frac{p}{2} \rfloor + \sum_{i \in S} e_i|$ will be closer to $\lfloor \frac{p}{2} \rfloor$ than 0 with all but negligible probability and we decrypt to 1. Therefore, it follows that Dec decrypts successfully with all but negligible probability.

For the proof of security, the general idea is to show that if there is a distinguisher that can tell the difference between encryptions of 0 and encryptions of 1, then there exists a distinguisher that can tell the difference between $A_{\mathbf{s},\chi}$ and $U_{\mathbb{Z}_p^{n+1}}$ for a nonnegligible fraction of the $\mathbf{s}$'s. If we let D be the algorithm that can distinguish between encryptions of 0 and encryptions of 1 then by assumption, we have that

$$|\Pr[D(0) = 1] - \Pr[D(1) = 1]| \geq \frac{1}{n^c}\,,$$

for some constant c where $\Pr[D(0) = 1]$ is the probability that D accepts on an encryption of 0, where the probability is taken over all possible public, secret key pairs and possible encryptions of 0. $\Pr[D(1) = 1]$ is defined analogously for encryptions of 1 and $\Pr[D(u) = 1]$ denotes the probability that D accepts on a uniform string in $\mathbb{Z}_p^{n+1}$. By the triangle inequality, it follows that either

$$|\Pr[D(0) = 1] - \Pr[D(u) = 1]| \geq \frac{1}{2n^c}\,,$$

or

$$|\Pr[D(1) = 1] - \Pr[D(u) = 1]| \geq \frac{1}{2n^c}\,.$$

From the above, we have shown that D is a distinguisher whose acceptance probability on encryptions of either 0 or 1 and the uniform distribution on $\mathbb{Z}_p^{n+1}$ differs by at least $1/2n^c$. WLOG, suppose that it is encryptions of 0. Since these acceptances probabilities were taken over all secret keys $\mathbf{s} \in \mathbb{Z}_p^n$, it can be shown by manipulation of the expectations that for at least $1/4n^c$ fraction of the $\mathbf{s}$'s that

$$|\Pr[D(0, \mathbf{s}) = 1] - \Pr[D(u, \mathbf{s}) = 1]| \geq \frac{1}{4n^c}$$

where $\Pr[D(0, \mathbf{s}) = 1]$ is the same thing as $\Pr[D(0) = 1]$ except $\mathbf{s}$ is now fixed and the probability is taken over the randomness of the public key and the encryption of 0. For any $\mathbf{s}$ satisfying the above property, an algorithm D' that distinguishes between $A_{\mathbf{s},\chi}$ and $U_{\mathbb{Z}_p^{n+1}}$ is given by first constructing a public key from the unknown distribution (either $A_{\mathbf{s},\chi}$ or $U_{\mathbb{Z}_p^{n+1}}$) by taking m samples. D' then estimates $\Pr[D(0, \mathbf{s}) = 1]$ by encrypting 0 a polynomial number of times and computing the fraction of acceptances. D' does the same thing to estimate $\Pr[D(u, \mathbf{s}) = 1]$ by running D a polynomial number of times on uniformly random strings in $\mathbb{Z}_p^{n+1}$. If the difference in the probability of acceptance between these two procedures differs by more than some polynomial fraction, say $1/16n^c$, then D' concludes that the distribution was $A_{\mathbf{s},\chi}$ and otherwise concludes that it was $U_{\mathbb{Z}_p^{n+1}}$. We omit the details here, but the full analysis can be found in [Reg05]. Essentially what the above has shown is that there is some good set of secret keys of nonnegligible size on which D' can distinguish $A_{\mathbf{s},\chi}$ from $U_{\mathbb{Z}_p^{n+1}}$, which contradicts the hardness of the learning with errors problem that we described earlier. Therefore, it follows that the encryption scheme presented is secure.

6.4 Other Cryptographic Constructions

The basic encryption scheme presented in the previous section is just the beginning of the cryptographic applications of lattices. As mentioned earlier, the first candidate fully homomorphic encryption scheme (an encryption scheme that allows addition and multiplication of ciphertexts) was constructed using lattices [Gen09]. This construction has since been refined and [BV13] presents an improved fully homomorphic encryption scheme based on lattices. Furthermore, other encryption schemes with desirable properties have been constructed using lattices. [GVW15] presents an attribute based encryption scheme based on lattices, which is an encryption scheme where only people who satisfy a particular property are capable of decrypting.

In addition to encryption schemes, there are other useful cryptographic primitives that have been constructed using lattices. In [GPV07], they show a construction of trapdoor functions based on lattices. Trapdoor functions are one-way functions (functions that can be computed in polynomial time, but cannot be inverted in polynomial time except with negligible probability) that possess the additional property that there is some special information called a trapdoor, which if known, makes the function invertible in polynomial time. [GPV07] goes on to further show how to use these trapdoor functions to construct digital signatures based on lattices, which are a method of message authentication for public key cryptography. Essentially, digital signatures provide a guarantee that the message sender is who they say they are, which is important in the public key setting since anybody is capable of encrypting and sending messages using a person's public key. Additionally, [BPR11] presents a construction of pseudorandom functions from lattices, which are functions that are indistinguishable to a polynomial time adversary from truly random functions.

The above cryptographic constructions based on lattices suggest that lattices are invaluable to cryptography. Moreover, the supposed resistance of lattice problems to quantum attacks suggests that lattice based cryptography may become the standard in a post-quantum world. For the interested reader, [Pei16] presents a beautiful survey of the current state of lattice based cryptography, detailing the many breakthroughs in the past few years.

7 Conclusions and Future Work

Currently, my ETH-hard result for SVP only applies in the l_∞ norm. A natural next step is to try to improve this result for the general l_p norm. I tried to use the reductions in [Mic01b] and [HR07] that I studied to see if something similar could be used to show exponential time hardness. Unfortunately, I was unsuccessful, and the main difficulty I encountered was that none of the reductions were linear in the dimension size. Instead, they all mapped 3SAT instances of dimension n to GAPSVP$_\gamma$ or SVP instances of dimension n^c for some constant $c > 1$. The way the reductions were constructed made the polynomial blowup seem inevitable, and it is quite possible that a different approach will be needed to show ETH-hardness of SVP in any l_p norm.

Additionally, the ETH-hardness results I showed for these lattice problems were only for the exact versions of the problems. In cryptography, however, we are interested in the approximation versions of these problems, where the approximation factor, γ, is some polynomial in n. If I could show ETH-hardness of GAPSIVP$_\gamma$ for $\gamma = \text{poly}(n)$, this would greatly increase our confidence in the security of lattice based cryptosystems.

References

[ABSS97] Sanjeev Arora, Lazlo Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[ADRSD15] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time using discrete gaussian sampling: Extended abstract. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 733–742, New York, NY, USA, 2015. ACM.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM.

[Ajt98] Miklós Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 10–19, New York, NY, USA, 1998. ACM.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 601–610, New York, NY, USA, 2001. ACM.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP $\cap$ coNP. *J. ACM*, 52(5):749–765, September 2005.

[Ban93] W. Banaszczyzyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

[BPR11] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. Cryptology ePrint Archive, Report 2011/401, 2011. `http://eprint.iacr.org/`.

[BS99] Johannes Blömer and Jean-Pierre Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, STOC '99, pages 711–720, New York, NY, USA, 1999. ACM.

[BV13] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. Cryptology ePrint Archive, Report 2013/541, 2013. `http://eprint.iacr.org/`.

[Gen09] Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, 2009.

[GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

[GPV07] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. Cryptology ePrint Archive, Report 2007/432, 2007. `http://eprint.iacr.org/`.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *J. ACM*, 62(6):45:1–45:33, December 2015.

[HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 469–477, New York, NY, USA, 2007. ACM.

[IP01] Russel Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, March 2001.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, December 2001.

[Kho05] S. Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, September 2005.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

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

[MG02] Daniele Micciancio and Shafi 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, March 2002.

[Mic01a] Daniele Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Transactions on Information Theory*, 47(3):1212–1215, 2001.

[Mic01b] Daniele Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, March 2001. Preliminary version in FOCS 1998.

[Mic08] Daniele Micciancio. Efficient reductions among lattice problems. In *Proceedings of SODA 2008*, San Francisco, CA, USA, January 2008. ACM-SIAM. To appear.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 333–342, New York, NY, USA, 2009. ACM.

[Pei16] Chris Peikert. A decade of lattice cryptography, February 2016. `https://web.eecs.umich.edu/~cpeikert/pubs/lattice-survey.pdf`.

[Reg04] Oded Regev. Lecture notes for lattices in computer science, September 2004. `http://www.cims.nyu.edu/~regev/teaching/lattices_fall_2004/index.html`.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[Sch87] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53(2-3):201–224, August 1987.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.

[Sip12] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 3rd edition, 2012.

[Vai15] Vinod Vaikuntanathan. Lecture notes for 6.876j advanced topics in cryptography: Lattices, September 2015. `http://people.csail.mit.edu/vinodv/6876-Fall2015/index.html`.

[vEB81] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. *Technical Report 81-04*, 1981.