Lattice Basis Reduction and Public-Key Cryptography

A Thesis Presented

by

Alexander D. Healy
to

Mathematics and Computer Science

in partial fulfillment of the honors requirements

for the degree of

Bachelor of Arts

Harvard College

Cambridge, Massachusetts

April 1, 2002

Name: Alexander D. Healy
Email: ahealy@fas.harvard.edu
Phone: (617) 493-2994
Advisor: Professor Michael Rabin (DEAS)

Contents

Introduction **2**

1 Lattices and Lattice Reduction **4**

1.1 Definitions and Properties of Lattices 4

1.2 Lattice Reduction and the LLL Algorithm 7

1.3 Provably Hard Lattice Problems 13

2 Lattice Cryptosystems **17**

2.1 The Goldreich-Goldwasser-Halevi Cryptosystem 17

2.1.1 Generating Keys 18

2.1.2 Encryption/Decryption 19

2.1.3 Cryptanalysis 20

2.2 The Ajtai-Dwork Cryptosystem 21

2.2.1 Generating Keys 21

2.2.2 Encryption/Decryption 22

2.2.3 Cryptanalysis 23

2.3 The NTRU Cryptosystem 25

2.3.1 Generating Keys 25

2.3.2 Encryption/Decryption 26

2.3.3 Cryptanalysis 26

Conclusion **29**

Acknowledgements **30**

Bibliography **30**

Introduction

Toward the end of the 1970's, the field of cryptography underwent a significant transformation with the introduction of public-key cryptosystems. Previous cryptosystems required that any two users share a secret key if they wished to communicate securely, whereas in the new public-key paradigm encryption and decryption used different keys: one encryption key which is made public, and one decryption key which is kept secret. Every user would produce such a pair of keys, and provided that there was some central repository of public keys, anyone could send encrypted messages to anyone else without ever having to exchange a secret key in advance. Naturally, if public-key cryptography is to work, it is essential that the (public) encryption key provide a wouldbe adversary with little or no information about how to decrypt encrypted messages. Information-theoretically, however, this is impossible. Nonetheless, if one is willing to assume widely believed conjectures about the complexity of certain computational problems (such as factoring large integers), then it is feasible to construct public-key cryptosystems that can be implemented efficiently, but that cannot be cracked efficiently. For instance, Rabin constructed a cryptosystem in [Rab79] where deciphering messages is provably as hard as factoring large integers of the form $N = pq$, where p and q are primes, and in [ElG85], ElGamal gave a cryptosystem which is secure provided that the so-called "discrete logarithm" problem is computationally intractable. Other cryptosystems, such as the well-known RSA cryptosystem and the Diffie-Hellman key-exchange protocol rely on very similar (albeit, possibly stronger) assumptions than the Rabin and ElGamal cryptosystems.

In addition to these cryptosystems, there have been various proposals that rely on the (conjectured) complexity of some other mathematical problems. For example, the Merkle-Hellman [MH78] and Chor-Rivest [CR88] cryptosystems rely on the intractability of a combinatorial problem known as the "knapsack problem", and the McEliece [McE78] cryptosystem relies on the intractability of decoding certain linear error-correcting codes. Unfortunately, these cryptosystems, and many others, have either been shown to be impractical (e.g. the public key may need to be very large in order to achieve a reasonable level of security) or to be insecure. Hence, we are left in a position where the remaining cryptosystems — those that are practical and not known to be insecure — are based on the assumptions that integer factorization and the discrete logarithm problem are intractable. This could be problematic, however, if it were ever discovered that either (or both) of these problems is not computationally infeasible. Indeed, Shor has shown (see [Sho94]) that efficient algorithms exist for integer factorization and certain versions of the discrete logarithm problem in the quantum computation model, which has raised awareness about the potential problem of having only public-key cryptosystems that rely on these assumptions.

One of the most recent classes of (conjecturally) intractable problems that has been considered for use in cryptography is that of lattice reduction problems. A *lattice* is typically represented by a collection of vectors that are said to form a *basis* for the lattice. Given a lattice, there are many possible bases, and the task of lattice reduction (more precisely, lattice basis reduction) is to find a basis where the vectors are as short as possible. Remarkable progress has been made in giving fast algorithms for sub-optimal lattice reduction, most notably the *LLL* algorithm and its variants which have a wide range of applications in computational mathematics and cryptanalysis. Even so, the problem of finding the shortest vectors in a lattice remains, in many cases, an intractable problem. This observation, together with a result due to Ajtai ([Ajt96]) which gives a class of lattice problems that are as hard on average as in the worst case, sparked an interest in using lattice problems in public-key cryptography.

The goal of this thesis is to present an introduction to the computational aspects of the theory of lattices (Chapter 1), with the ultimate goal of understanding the constructions underlying these "lattice cryptosystems" as well as the attacks that thwart them (Chapter 2). We assume familiarity with the basics of public key cryptography, elementary complexity theory (e.g. $\mathcal{NP}$-completeness and polynomial-time reductions) and linear algebra. The remainder of this thesis is organized as follows. Chapter 1 treats the basic definitions and properties of lattices, definitions of reduced bases and the *LLL* algorithm, and a brief discussion of lattice problems that have been proved to be hard in a well-defined sense. Chapter 2 describes the three most notable "lattice cryptosystems": the Goldreich-Goldwasser-Halevi, Ajtai-Dwork and NTRU cryptosystems. For each cryptosystem, we will describe the protocol, the justifications for security and the most effective cryptanalytic attacks against it. We will conclude with a discussion of the current state of lattice cryptosystems and some possible directions in which the field may go, including some open problems.

Chapter 1

Lattices and Lattice Reduction

1.1 Definitions and Properties of Lattices

The *lattices* that we will be studying are defined to be discrete subgroups $\mathcal{L}$ of n-dimensional Euclidean space $\mathbb{R}^n$ given by $\mathcal{L} \stackrel{\text{def}}{=} \mathbb{Z}\vec{b}_1 + \cdots + \mathbb{Z}\vec{b}_m$, where the $\{\vec{b}_i\}_{i=1}^m$ are linearly independent (i.e. they form a basis for an m-dimensional linear subspace of $\mathbb{R}^n$). In this case, the $\{\vec{b}_i\}_{i=1}^m$ are said to form a *basis* for the lattice $\mathcal{L}$ and typically a lattice will be given by the $n \times m$ matrix of the form:

$$B = \left(\begin{bmatrix} | \\ \vec{b}_1 \\ | \end{bmatrix} \quad \begin{bmatrix} | \\ \vec{b}_2 \\ | \end{bmatrix} \quad \cdots \quad \begin{bmatrix} | \\ \vec{b}_m \\ | \end{bmatrix} \right)$$

This allows us to think of $\mathcal{L}$ as the image of $\mathbb{Z}^m$ under the linear map defined by B, i.e. $\mathcal{L} = B\mathbb{Z}^m$. However, the choice of a basis B for a lattice $\mathcal{L}$ is by no means unique. For instance, if we let M be an $m \times m$ integer matrix, then $BM\mathbb{Z}^m$ is necessarily a sub-lattice of $B\mathbb{Z}^m$ because $M\mathbb{Z}^m \subseteq \mathbb{Z}^m$, and therefore $BM\mathbb{Z}^m \subseteq B\mathbb{Z}^m$. Furthermore, if we require that M have determinant equal to 1 or -1, then it is a basic fact from linear algebra that M^{-1} will also have integer entries, so $M^{-1}\mathbb{Z}^m \subseteq \mathbb{Z}^m$ or alternatively $\mathbb{Z}^m \subseteq M\mathbb{Z}^m$, and hence we have $B\mathbb{Z}^m \subseteq BM\mathbb{Z}^m$. Therefore we can conclude that $BM\mathbb{Z}^m = B\mathbb{Z}^m$, i.e. BM is also a basis for $\mathcal{L} = B\mathbb{Z}^m$.

Next we will define the fundamental parallelepiped, determinant and dual of a lattice.

Definition 1.1. *Let $\mathcal{L} = B\mathbb{Z}^m$ be a lattice generated by basis vectors $\{\vec{b}_i\}_{i=1}^m$. Then the* fundamental parallelepiped *of $\mathcal{L}$ with respect to B is defined by*

$$P(B) = \{c_1\vec{b}_1 + \cdots + c_m\vec{b}_m \in \mathbb{R}^n \mid 0 \leq c_i < 1\}$$

Also, we denote by $\text{width}(P(B))$ *the* width *of the parallelepiped $P(B)$, i.e. the maximum, over i, of the length of the projection of $\vec{b}_i$ onto the orthogonal complement of* $\text{span}(\vec{b}_1, \ldots, \vec{b}_{i-1}, \vec{b}_{i+1}, \ldots, \vec{b}_m)$.

Definition 1.2. *Let $\mathcal{L} = B\mathbb{Z}^m$ be a lattice. Then the* determinant *of $\mathcal{L}$,* $\det(\mathcal{L})$, *is defined to be the m-dimensional volume of $P(\mathcal{L})$. In particular, if $\mathcal{L}$ is full-dimensional (i.e. $m = n$), then* $\det(\mathcal{L}) = |\det(B)|$.

The following proposition shows that the determinant of a lattice is in fact well-defined.

Proposition 1.3. *The determinant of $\mathcal{L} = B\mathbb{Z}^m$ does not depend on the choice of basis B, and is given by* $\det(\mathcal{L}) = \sqrt{\det(B^T B)}$.

Proof. First we will prove the result when $\mathcal{L}$ is a full-dimensional lattice. Suppose that $\mathcal{L} = B\mathbb{Z}^n = B'\mathbb{Z}^n$ where B and B' are two different bases of $\mathcal{L}$. Then there must be integer matrices M, M' such that $B'M' = B$ and $BM = B'$ which implies that $B = BMM'$. By the multiplicativity of the determinant, this means that $\det(MM') = 1$, and hence that $\det(M), \det(M') \in \{-1, 1\}$, since M and M' are integer matrices. We know that $B' = BM$, and therefore $|\det(B)| = |\det(B')|$, concluding the proof in the case where $\mathcal{L}$ is full-dimensional.

Now suppose that $\mathcal{L} = B\mathbb{Z}^m = B'\mathbb{Z}^m$ is an m-dimensional lattice where $m < n$. Let U be an orthogonal linear transformation that maps the m-dimensional subspace $\mathbb{R}\vec{b}_1 + \cdots + \mathbb{R}\vec{b}_m = \mathbb{R}\vec{b}'_1 + \cdots + \mathbb{R}\vec{b}'_m$ to $\{(v_1, \ldots, v_n) \in \mathbb{R}^n \mid v_{m+1} = \cdots = v_n = 0\}$. (Such a map can be constructed by composing $(n-m)$ rotations that eliminate coordinates $m+1$ though n one at a time.) Since U is orthogonal, it is area-preserving and therefore the m-dimensional area of the fundamental parallelepiped of $U\mathcal{L}$ (with respect to the basis UB) is the same as the m-dimensional area of the fundamental parallelepiped of $\mathcal{L}$ (with respect to B). By construction UB (respectively UB') has m non-zero rows and $n-m$ rows of zeros, so let B_m (resp. B'_m) be the $m \times m$ matrix consisting of the non-zero rows of UB (resp. UB'). It is not difficult to see that $|\det(B_m)|$ is equal to the m-dimensional area of the fundamental parallelepiped of $\mathcal{L} = B\mathbb{Z}^m$ and that $|\det(B'_m)|$ is equal to the m-dimensional area of the fundamental parallelepiped of $\mathcal{L} = B'\mathbb{Z}^m$. Hence, by the proof in the case where $\mathcal{L}$ was full-dimensional, it follows that these two volumes are the same. Furthermore, $\det(B^T B) = \det((UB)^T UB) = \det(B_m)^2 = \det(L)^2$, yielding an efficient way to compute $\det(\mathcal{L})$ as $\sqrt{\det(B^T B)}$. $\square$

Definition 1.4. *The* dual lattice *of a full-dimensional lattice $\mathcal{L}$, denoted $\mathcal{L}^*$, consists of all vectors whose inner-product with all vectors in $\mathcal{L}$ is an integer, i.e.*

$$\mathcal{L}^* = \{\vec{v} \in \mathbb{R}^n \mid \langle \vec{v}, \vec{z} \rangle \in \mathbb{Z} \text{ for all } \vec{z} \in \mathbb{Z}^n\}$$

Hence if B is a basis for $\mathcal{L}$, then $B^{T^{-1}}$ is a basis for $\mathcal{L}^*$.

In the next section, we will be concerned with the problem of finding a basis B' for a lattice $\mathcal{L} = B\mathbb{Z}^m$, such that the $\vec{b}'_i$ are "short" in some well-defined sense. In preparation for this, we will define a Minkowski reduced basis, and an LLL-reduced basis.

Definition 1.5. *Let $\mathcal{L} = B\mathbb{Z}^m$ be a lattice. Then the basis B for $\mathcal{L}$ is said to be* Minkowski reduced *if there is no non-zero vector in $\mathcal{L}$ that is shorter than $\vec{b}_1$, and in general there is no non-zero vector in $\mathcal{L}$ that is shorter than $\vec{b}_i$ and not contained in* $\operatorname{span}(\vec{b}_1, \ldots, \vec{b}_{i-1})$ *such that $\vec{b}_1, \ldots, \vec{b}_i$ can be extended to a basis for $\mathcal{L}$.*

Before defining the notion of an *LLL* reduced basis, we recall the Gram-Schmidt orthogonalization method where, given a basis $\vec{b}_1, \ldots, \vec{b}_n$, we construct an orthogonal basis $\vec{b}^*_1, \ldots, \vec{b}^*_n$, by setting

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

where $\mu_{i,j} \stackrel{\text{def}}{=} \langle \vec{b}_i, \vec{b}_j^* \rangle / \langle \vec{b}_j^*, \vec{b}_j^* \rangle = \langle \vec{b}_i, \vec{b}_j^* \rangle / \|\vec{b}_j^*\|^2$.

Definition 1.6. *Let $\mathcal{L} = B\mathbb{Z}^m$ be a lattice. Then the basis B for $\mathcal{L}$ is said to be* LLL reduced *if*

$$|\mu_{i,j}| \le \frac{1}{2} \text{ for } 1 \le j < i \le m$$

and

$$\|\vec{b}_i^* + \mu_{i,i-1}\vec{b}_{i-1}^*\|^2 \ge \frac{3}{4}\|\vec{b}_{i-1}^*\|^2 \text{ for } 1 < i \le m$$

The constant $\frac{3}{4}$ is arbitrary. It may be replaced with any δ satisfying $\frac{1}{4} < \delta < 1$, in which case the basis is said to be *LLL* reduced with respect to the parameter δ. We will complete the analysis using the constant $\frac{3}{4}$ for notational simplicity, noting that several occurrences of the constant "2" (such as in 2^{m-1} in the following result) would need to be replaced with $\frac{1}{\delta - \frac{1}{4}}$ for the more general result.

The following proposition, due to Lenstra *et al* in [LLL82], gives an explicit guarantee on the length on of the vectors in an *LLL* reduced lattice.

Proposition 1.7. *Let $\vec{b}_1, \dots, \vec{b}_m$ be an LLL reduced basis for a lattice $\mathcal{L} \subseteq \mathbb{R}^n$, and let $\vec{x}_1, \dots, \vec{x}_t \in \mathcal{L}$ be linearly independent lattice points. Then*

$$\|\vec{b}_i\|^2 \le 2^{m-1} \cdot \max\{\|\vec{x}_1\|^2, \dots, \|\vec{x}_t\|^2\}$$

for $1 \le i \le t$.

Proof. First we will show that $\|\vec{b}_j^*\|^2 \le 2^{i-j}\|\vec{b}_i^*\|^2$ for $1 \le j \le i \le m$. By the definition of an *LLL* reduced basis, we have that $\|\vec{b}_i^* + \mu_{i,i-1}\vec{b}_{i-1}^*\|^2 \ge \frac{3}{4}\|\vec{b}_{i-1}^*\|^2$, or equivalently, $\|\vec{b}_i^*\|^2 \ge \left(\frac{3}{4} - \mu_{i,i-1}^2\right)\|\vec{b}_{i-1}^*\|^2$ for $1 < i \le m$. Since $|\mu_{i,i-1}| \le \frac{1}{2}$ by assumption, this gives us that $\|\vec{b}_i^*\|^2 \ge \frac{1}{2}\|\vec{b}_{i-1}^*\|^2$, and so it follows by induction that $\|\vec{b}_j^*\|^2 \le 2^{i-j}\|\vec{b}_i^*\|^2$ for $1 \le j \le i \le m$. We can use this to show that $\|\vec{b}_i\|^2 \le 2^{i-1}\|\vec{b}_i^*\|^2$ as follows. By the definition of the Gram-Schmidt vectors $\vec{b}_i^*$, we have $\vec{b}_i = \vec{b}_i^* + \sum_{j=1}^{i-1} \mu_{i,j}\vec{b}_j^*$, where the vectors $\vec{b}_i^*$ are orthogonal by construction, and thus

$$\begin{aligned}
\|\vec{b}_i\|^2 &= \|\vec{b}_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \|\vec{b}_j^*\|^2 \\
&\le \|\vec{b}_i^*\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} \cdot 2^{i-j} \|\vec{b}_i^*\|^2 \\
&= \|\vec{b}_i^*\|^2 \left(1 + \frac{1}{8}\left(2^{i-1} - 1\right)\right) \\
&\le 2^{i-1}\|\vec{b}_i^*\|^2
\end{aligned}$$

Now let $r_{i,j} \in \mathbb{Z}$ be such that $\vec{x}_j = \sum_{i=1}^m r_{i,j}\vec{b}_i$, and define, for each j, $\ell(j)$ to be the largest i such that $r_{i,j} \ne 0$. In particular, if we write $\vec{x}_j = \sum_{i=1}^m r'_{i,j}\vec{b}_i^*$, where $r'_{i,j} \in \mathbb{R}$, then we have that

$r'_{\ell(j),j} = r_{\ell(j),j} \in \mathbb{Z}$ by the definition of the Gram-Schmidt procedure, and hence we have that $\|\vec{x}_j\|^2 \geq r'^2_{\ell(j),j}\|\vec{b}_{\ell(j)}\|^2 \geq \|\vec{b}_{\ell(j)}\|^2$. Without loss of generality, we may assume that the $\vec{x}_j$ are such that $\ell(1) \leq \cdots \leq \ell(t)$, and this forces $j \leq \ell(j)$, for if this were not the case, we would have $\vec{x}_1, \ldots, \vec{x}_j \in \operatorname{span}(\vec{b}_1, \ldots, \vec{b}_{j-1})$, violating the assumption that $\vec{x}_1, \ldots, \vec{x}_t$ are linearly independent. Combining this with the previous observations that $\|\vec{b}_j^*\|^2 \leq 2^{i-j}\|\vec{b}_i^*\|^2$ for $1 \leq i \leq j \leq m$ and that $\|\vec{x}_j\|^2 \geq \|\vec{b}_{\ell(j)}\|^2$, we have

$$\|\vec{b}_i\|^2 \leq 2^{i-1}\|\vec{b}_i^*\|^2 \leq 2^{\ell(i)-1}\|\vec{b}_{\ell(i)}^*\|^2 \leq 2^{m-1}\|\vec{x}_i\|$$

□

1.2 Lattice Reduction and the LLL Algorithm

In general, it is difficult to find a Minkowski reduced basis for a lattice. For small dimensions this is a tractable problem (e.g. see [CS93]), and we give an algorithm for 2-dimensional lattices below. However, the complexity of the best known methods grows exponentially in the dimension of the lattice, and they quickly become impractical. Even so, in [LLL82] an approximation algorithm is given that finds a reasonably reduced basis in polynomial time. This algorithm has come to be known as the *LLL* (or L^3) algorithm (after its authors, Lenstra, Lenstra and Lovász), and the basis returned by this algorithm is guaranteed to be *LLL reduced*.

First, however, we consider the problem of finding a Minkowski reduced basis of a 2-dimensional lattice.

Proposition 1.8. *A basis $B = [\vec{b}_1, \vec{b}_2]$ for a 2-dimensional lattice $\mathcal{L}$ is Minkowski reduced if and only if $\|\vec{b}_1\| \leq \|\vec{b}_2\| \leq \|\vec{b}_1 + \vec{b}_2\|, \|\vec{b}_1 - \vec{b}_2\|$.*

Proof. If B is Minkowski reduced, then by definition, $\|\vec{b}_1\| \leq \|\vec{b}_2\|$. Furthermore, if it were not the case that $\|\vec{b}_2\| \leq \|\vec{b}_1 + \vec{b}_2\|$ (resp. $\|\vec{b}_2\| \leq \|\vec{b}_1 - \vec{b}_2\|$), then $\vec{b}_1 + \vec{b}_2$ (resp. $\vec{b}_1 - \vec{b}_2$) would be shorter than $\vec{b}_2$, and since it is clearly linearly independent of $\vec{b}_1$ this would contradict the fact that $\vec{b}_2$ is the shortest non-zero vector that is linearly independent of $\vec{b}_1$.

Now suppose that B satisfies $\|\vec{b}_1\| \leq \|\vec{b}_2\| \leq \|\vec{b}_1 + \vec{b}_2\|, \|\vec{b}_1 - \vec{b}_2\|$, and consider an arbitrary non-zero lattice vector $\vec{v} = p\vec{b}_1 + q\vec{b}_2$ where $(p, q) \in \mathbb{Z}^2 \setminus \{\vec{0}\}$. Without loss of generality we may assume that p and q are both non-negative since we may freely replace $[\vec{b}_1, \vec{b}_2]$ with $[\pm\vec{b}_1, \pm\vec{b}_2]$ while preserving the fact that $\|\vec{b}_1\| \leq \|\vec{b}_2\| \leq \|\vec{b}_1 + \vec{b}_2\|, \|\vec{b}_1 - \vec{b}_2\|$.

If either q or p is zero, then $\vec{v}$ is simply a multiple of either $\vec{b}_1$ or $\vec{b}_2$, and so we have that $\|\vec{v}\| \geq \|\vec{b}_1\|$.

Before continuing, we recall a basic geometric fact which we will use several times in the remainder of the proof: If $\vec{a}, \vec{b}$ and $\vec{c}$ lie on a line (in this order, i.e. $\vec{b}$ lies between $\vec{a}$ and $\vec{c}$) then if $\|\vec{b}\| > \|\vec{a}\|$ it must be the case that $\|\vec{c}\| > \|\vec{b}\|$.

Next we consider the case where both p and q are non-zero, i.e. p and q are both strictly positive. If $q \geq p$ we have $\|\vec{v}\| = \|p\vec{b}_1 + q\vec{b}_2\| \geq \|\vec{b}_1 + \frac{q}{p}\vec{b}_2\|$, where $\frac{q}{p} \geq 1$. However, $\vec{b}_1, \vec{b}_1 + \vec{b}_2$ and $\vec{b}_1 + \frac{q}{p}\vec{b}_2$

lie on a line (in this order), and hence if it were the case that $\|\vec{v}\| < \|\vec{b}_1\|$, we would have that $\|\vec{b}_1 + \frac{q}{p}\vec{b}_2\| < \|\vec{b}_1\|$ which would imply that $\|\vec{b}_1 + \vec{b}_2\| < \|\vec{b}_1\|$, but this is a contradiction. On the other hand, if $p > q$ we have $\|\vec{v}\| = \|p\vec{b}_1 + q\vec{b}_2\| \geq \|\frac{p}{q}\vec{b}_1 + \vec{b}_2\|$, where $\frac{p}{q} > 1$. By the same reasoning as before, since $\vec{b}_1, \vec{b}_1 + \vec{b}_2$ and $\frac{p}{q}\vec{b}_1 + \vec{b}_2$ lie on a line (in this order), if it were the case that $\|\vec{v}\| < \|\vec{b}_1\|$, we would have that $\|\frac{q}{p}\vec{b}_1 + \vec{b}_2\| < \|\vec{b}_1\|$ which would imply that $\|\vec{b}_1 + \vec{b}_2\| < \|\vec{b}_1\|$, which is again a contradiction.

Thus, we have that $\|\vec{v}\| \geq \|\vec{b}_1\|$, which proves that $\vec{b}_1$ is a shortest vector in the lattice. It remains to be shown that $\vec{b}_2$ is one of the shortest vectors which are linearly independent of $\vec{b}_1$. Let $\vec{v} = p\vec{b}_1 + q\vec{b}_2$ where $q \neq 0$, and suppose that $\|\vec{v}\| < \|\vec{b}_2\|$. Consider the three collinear points $\vec{b}_2, \vec{b}_2 + \vec{b}_1$ and $\vec{b}_2 + \frac{p}{q}\vec{b}_1$. If $p \geq q$ these points lie in the above order, and thus if it were the case that $\|\vec{b}_2 + \frac{p}{q}\vec{b}_1\| \leq \|\vec{v}\| < \|\vec{b}_2\|$, this would mean that $\|\vec{b}_2 + \vec{b}_1\| < \|\vec{b}_2\|$ which is a contradiction. If $p < q$ and $\|\vec{v}\| = \|p\vec{b}_1 + q\vec{b}_2\| < \|\vec{b}_2\|$, then we have that $\|\frac{p}{q}\vec{b}_1 + \vec{b}_2\| < \frac{1}{q}\|\vec{b}_2\|$. By the triangle inequality, we also have $\|\frac{p}{q}\vec{b}_1 + \vec{b}_2\| \geq \left|\|\frac{p}{q}\vec{b}_1\| - \|\vec{b}_2\|\right|$. Since $p < q$, $\|\frac{p}{q}\vec{b}_1\| \leq \frac{q-1}{q}\|\vec{b}_1\| \leq \frac{q-1}{q}\|\vec{b}_2\|$ implies that $\|\frac{p}{q}\vec{b}_1 + \vec{b}_2\| \geq \left|\|\frac{p}{q}\vec{b}_1\| - \|\vec{b}_2\|\right| \geq \frac{1}{q}\|\vec{b}_2\|$, contradicting the previous observation that $\|\frac{p}{q}\vec{b}_1 + \vec{b}_2\| < \frac{1}{q}\|\vec{b}_2\|$. □

Algorithm 1.9 (Gauss). *Let $[\vec{b}_1, \vec{b}_2]$ be a basis for a two dimensional lattice $\mathcal{L}$. Then the following algorithm returns a Minkowski reduced lattice within polynomially many iterations.*

IF $\|\vec{b}_1\| > \|\vec{b}_2\|$ THEN swap$(\vec{b}_1, \vec{b}_2)$;
WHILE $\|\vec{b}_2\| > \|\vec{b}_1 + \vec{b}_2\|$ *or* $\|\vec{b}_2\| > \|\vec{b}_1 - \vec{b}_2\|$ DO
    $\vec{b}_2 = \vec{b}_2 - \left\lceil \frac{\langle \vec{b}_1, \vec{b}_2 \rangle}{\|\vec{b}_1\|^2} \right\rfloor \vec{b}_1$;
    IF $\|\vec{b}_1\| > \|\vec{b}_2\|$ THEN swap$(\vec{b}_1, \vec{b}_2)$;
END WHILE
RETURN$[\vec{b}_1, \vec{b}_2]$

Proof. Clearly the algorithm will only return a Minkowski reduced basis by Proposition 1.8 (observe that it is always the case that $\|\vec{b}_1\| \leq \|\vec{b}_2\|$ at the beginning and end of the WHILE loop), so we need only show that the algorithm halts in polynomial time. First we note that $\frac{\langle \vec{b}_1, \vec{b}_2 \rangle}{\|\vec{b}_1\|^2}$ is equal to the Gram-Schmidt coefficient $\mu_{2,1}$. Furthermore, we note that the condition "$\|\vec{b}_2\| > \|\vec{b}_1 + \vec{b}_2\|$ or $\|\vec{b}_2\| > \|\vec{b}_1 - \vec{b}_2\|$" is equivalent to $|\mu_{2,1}| > \frac{1}{2}$: Indeed, $\|\vec{b}_2\| > \|\vec{b}_1 + \vec{b}_2\|$ if and only if $\|\vec{b}_2\|^2 > \|\vec{b}_1 + \vec{b}_2\|^2$, or equivalently $\langle \vec{b}_2, \vec{b}_2 \rangle > \langle \vec{b}_1 + \vec{b}_2, \vec{b}_1 + \vec{b}_2 \rangle = \langle \vec{b}_1, \vec{b}_1 \rangle + 2\langle \vec{b}_1, \vec{b}_2 \rangle + \langle \vec{b}_2, \vec{b}_2 \rangle$, i.e. $\mu_{2,1} = \frac{\langle \vec{b}_1, \vec{b}_2 \rangle}{\langle \vec{b}_1, \vec{b}_1 \rangle} < -\frac{1}{2}$. By the same reasoning, $\|\vec{b}_2\| > \|\vec{b}_1 - \vec{b}_2\|$ if and only if $\langle \vec{b}_2, \vec{b}_2 \rangle > \langle \vec{b}_1 - \vec{b}_2, \vec{b}_1 - \vec{b}_2 \rangle = \langle \vec{b}_1, \vec{b}_1 \rangle - 2\langle \vec{b}_1, \vec{b}_2 \rangle + \langle \vec{b}_2, \vec{b}_2 \rangle$, i.e. $\mu_{2,1} = \frac{\langle \vec{b}_1, \vec{b}_2 \rangle}{\langle \vec{b}_1, \vec{b}_1 \rangle} > \frac{1}{2}$. Thus, in the case of a two-dimensional lattice, the notions of Minkowski reduced and LLL reduced are equivalent. With this observation, we will suppress the remainder of the proof, noting simply that Algorithm 1.9 is a special case of Algorithm 1.12 (the LLL algorithm) which we prove returns an LLL reduced basis within polynomially many iterations. □

Next we will describe the LLL basis reduction algorithm. Let $\vec{b}_1, \ldots, \vec{b}_m$ be a basis for an m-dimensional lattice $\mathcal{L} \subseteq \mathbb{Z}^n$. We assume that the Gram-Schmidt coefficients $\mu_{i,j} = \langle \vec{b}_i, \vec{b}_j^* \rangle / \langle \vec{b}_j^*, \vec{b}_j^* \rangle$, and the squared lengths of the Gram-Schmidt basis, $B_i \stackrel{\text{def}}{=} \|\vec{b}_i^*\|^2$ are always available (and up-to-date for the current basis) at every stage of the algorithm. We will now describe the two major subroutines used in the LLL algorithm.

Subroutine 1.10 (Size reduction).
SIZEREDUCE(k)
    $i = 1$;
    WHILE $i < k$ DO
        $\vec{b}_k = \vec{b}_k - \lceil \mu_{k,k-i} \rfloor \vec{b}_{k-i}$;
        $\mu_{k,k-i} = \mu_{k,k-i} - \lceil \mu_{k,k-i} \rfloor$;
        $i = i + 1$;
    END WHILE
END SIZEREDUCE

First, we note that the Gram-Schmidt vectors $\vec{b}_i^*$ are unchanged, since the only change that is made to $\vec{b}_k$ is to subtract a vector that is in $\text{span}(\vec{b}_1^*, \ldots, \vec{b}_{k-1}^*)$. Therefore the B_i are unaffected by a call to SIZEREDUCE(k).

Second, after calling SIZEREDUCE(k), all the Gram-Schmidt coefficients $\mu_{s,t}$ $(1 \le t < s \le m)$ are accurate for the updated basis: This is because every time the basis vector $\vec{b}_k$ is modified by subtracting a multiple of $\vec{b}_{k-i}$, the appropriate change is made to $\mu_{k,k-i}$ and no other $\mu_{s,t}$ $(1 \le t < s \le m)$ are affected since, in general, $\mu_{s,t}$ only changes if $\vec{b}_s$ changes, and we have accounted for the changes to all $\mu_{k,t}$ where $1 \le t < k$.

Finally, we observe that the assignment $\mu_{k,k-i} = \mu_{k,k-i} - \lceil \mu_{k,k-i} \rfloor$ ensures that $|\mu_{k,s}| \le 1/2$ for all $s < k$. In particular, if we know that the partial basis $\vec{b}_1, \ldots, \vec{b}_{k-1}$ is LLL reduced, then the partial basis $\vec{b}_1, \ldots, \vec{b}_k$ that is obtained after running SIZEREDUCE(k) will satisfy the first condition of being LLL reduced, i.e. $|\mu_{i,j}| \le \frac{1}{2}$ for $1 \le j < i \le k$.

Next we will examine the SWAP subroutine.

Subroutine 1.11 (Swap). SWAP(k)

Exchange b_{k-1} *and* b_k*;*
$T = B_k$*;*
$\mu = \mu_{k,k-1}$*;*
$\mu_{k,k-1} = \frac{\mu_{k,k-1}B_{k-1}}{B_k+\mu_{k,k-1}^2 B_{k-1}}$*;*
Simultaneously perform the following assignments:
  $B_{k-1} = B_k + \mu^2 B_{k-1}$*;*
  $B_k = \frac{B_{k-1}B_k}{B_k+\mu^2 B_{k-1}}$*;*
\\\\ *Update the Gram-Schmidt coefficients:*
$i = 1$*;*
WHILE $i \leq k-2$ DO
  Exchange $\mu_{k-1,i}$ *and* $\mu_{k,i}$*;*
END WHILE
$i = k+1$*;*
WHILE $i \leq n$ DO
  Simultaneously perform the following assignments:
  $\mu_{i,k-1} = \mu_{i,k-1}\mu + \mu_{i,k}T/B_{k-1}$*;*
  $\mu_{i,k} = \mu_{i,k-1} - \mu_{i,k}\mu_{k,k-1}$*;*
END WHILE
END SWAP

First we will show that the assignments to B_{k-1} and B_k are correct (i.e. these are the squared lengths of the $(k-1)$-st and k-th Gram-Schmidt vectors of the new basis, after $\vec{b}_{k-1}$ and $\vec{b}_k$ are swapped). This is sufficient to show that all the B_i are correct, as $\vec{b}^*_{k-1}$ and $\vec{b}^*_k$ are the only Gram-Schmidt vectors that are affected by swapping $\vec{b}_{k-1}$ and $\vec{b}_k$. Denote the original basis (before the swap) by vectors $\vec{b}_i$ and denote the new basis (after the swap) by vectors $\vec{b}'_i$. By the definition of the Gram-Schmidt procedure, $\vec{b}'^*_{k-1}$ is the projection of $\vec{b}'_{k-1} = \vec{b}_k$ onto the orthogonal complement of $\text{span}(\vec{b}'_1, \ldots, \vec{b}'_{k-2}) = \text{span}(\vec{b}_1, \ldots, \vec{b}_{k-2})$. That is, $\vec{b}'^*_{k-1} = \vec{b}^*_k + \mu_{k,k-1}\vec{b}^*_{k-1}$, and so $\|\vec{b}'^*_{k-1}\|^2 = B_k + \mu^2_{k,k-1}B_{k-1}$, which is exactly the value that is assigned to B_{k-1}. Similarly, $\vec{b}'^*_k$ is the projection of $\vec{b}'_k = \vec{b}_{k-1}$ onto the orthogonal complement of $\text{span}(\vec{b}'_1, \ldots, \vec{b}'_{k-1}) = \text{span}(\vec{b}_1, \ldots, \vec{b}_{k-2}, \vec{b}_k)$. Hence, $\vec{b}'^*_k$ can be obtained by taking the projection of $\vec{b}_{k-1}$ onto the orthogonal complement of $\text{span}(\vec{b}_1, \ldots, \vec{b}_{k-2})$ (i.e. $\vec{b}'^*_k$) and then subtracting the projection of $\vec{b}^*_{k-1}$ onto the orthogonal complement of the projection of $\vec{b}_k$ onto the orthogonal complement of $\text{span}(\vec{b}_1, \ldots, \vec{b}_{k-2})$ (i.e. the projection of $\vec{b}^*_{k-1}$ onto the orthogonal complement of $\vec{b}'^*_{k-1}$, or $\frac{\langle \vec{b}^*_{k-1}, \vec{b}'^*_{k-1}\rangle}{\langle \vec{b}'^*_{k-1}, \vec{b}'^*_{k-1}\rangle} \cdot \vec{b}'^*_{k-1}$). Thus,

$$\vec{b}'^*_k = \vec{b}^*_k - \frac{\langle \vec{b}^*_{k-1}, \vec{b}'^*_{k-1}\rangle}{\langle \vec{b}'^*_{k-1}, \vec{b}'^*_{k-1}\rangle} \cdot \vec{b}'^*_{k-1}$$

Hence,

$$\|\vec{b}'^*_k\|^2 = \left\| \vec{b}^*_{k-1} - \frac{\langle \vec{b}^*_{k-1}, \vec{b}'^*_{k-1}\rangle}{\langle \vec{b}'^*_{k-1}, \vec{b}'^*_{k-1}\rangle} \cdot \vec{b}'^*_{k-1} \right\|^2 = \left\| \vec{b}^*_{k-1} - \frac{\mu_{k,k-1}B_{k-1}}{B_k + \mu^2_{k,k-1}B_{k-1}} \cdot \vec{b}'^*_{k-1} \right\|^2$$

$$= \left\| \vec{b}_{k-1}^* - \frac{\mu_{k,k-1}^2 B_{k-1}}{B_k + \mu_{k,k-1}^2 B_{k-1}} \cdot \vec{b}_{k-1}^* \right\|^2 + \left\| \vec{b}_{k-1}^* - \frac{\mu_{k,k-1} B_{k-1}}{B_k + \mu_{k,k-1}^2 B_{k-1}} \cdot \vec{b}_k^* \right\|^2$$

$$= \left\| \frac{B_k \vec{b}_{k-1}^*}{B_k + \mu_{k,k-1}^2 B_{k-1}} \right\|^2 + \frac{\mu_{k,k-1}^2 B_{k-1}^2 B_k}{\left(B_k + \mu_{k,k-1}^2 B_{k-1}\right)^2} = \frac{B_{k-1} B_k}{B_k + \mu_{k,k-1}^2 B_{k-1}}$$

which is exactly what is assigned to B_k.

In addition to updating B_{k-1} and B_k, we must update all the Gram-Schmidt coefficients that involve $\vec{b}_{k-1}$ or $\vec{b}_k$. We do this by computing the new coefficients in terms of the previous ones as in SIZEREDUCE(k). For $\mu_{i,j}$ where $1 \leq j < i \leq k-1$ this simply amounts to exchanging $\mu_{k,j}$ and $\mu_{k-1,j}$. To prove that the other assignments actually represent the correct values for the $\mu_{i,j}$ takes some more work however, so we leave it as an exercise, or alternatively the reader may consult [LLL82, Coh93].

A simple, yet essential observation about the SWAP subroutine, is that if the partial basis $\vec{b}_1, \ldots, \vec{b}_k$ is *LLL* reduced, then after running SWAP(k), the partial basis $\vec{b}_1, \ldots, \vec{b}_{k-1}$ is still *LLL* reduced, since none of the vectors $\vec{b}_1, \ldots, \vec{b}_{k-1}$ is affected by a call to SWAP(k).

Now we present the complete *LLL* algorithm.

Algorithm 1.12 (LLL basis reduction). *The following algorithm computes an LLL reduced basis for $\mathcal{L} = B\mathbb{Z}^n$ within polynomially many iterations.*

$LLL(\vec{b}_1, \ldots, \vec{b}_m)$
  Compute the Gram-Schmidt orthogonal basis $\vec{b}_1^*, \ldots, \vec{b}_m^*$*;*
  $B_i = \|\vec{b}_i^*\|^2$*;*
  $k = 2$*;*
  WHILE $k \leq m$ DO
    IF $|\mu_{k,k-1}| > \frac{1}{2}$ THEN SIZEREDUCE(k);
    IF $\|\vec{b}_k^* + \mu_{k,k-1}\vec{b}_{k-1}^*\|^2 < \frac{3}{4}\|\vec{b}_{k-1}^*\|^2$ THEN
      SWAP(k)*;*
      IF $k > 2$ THEN $k = k - 1$*;*
    ELSE
      $k = k + 1$*;*
    END IF
  END WHILE
END *LLL*

Proof. The correctness of the algorithm follows from previous remarks about the subroutines SIZEREDUCE and SWAP. In particular, if we consider the value of k at the beginning of each iteration though the WHILE loop, we have that $\vec{b}_1, \ldots, \vec{b}_{k-1}$ is *LLL* reduced; this is trivially the case when $k = 2$, and as we noted above, the (possible) call to SIZEREDUCE(k) will not change the fact that $\vec{b}_1, \ldots, \vec{b}_{k-1}$ is *LLL* reduced. Furthermore it will ensure that $\vec{b}_1, \ldots, \vec{b}_{k-1}$ satisfy the first condition of being *LLL* reduced. Now consider the second IF statement. If the condition,

$\|\vec{b}_k^* + \mu_{k,k-1}\vec{b}_{k-1}^*\|^2 < \frac{3}{4}\|\vec{b}_{k-1}^*\|^2$, is false then we in fact have that $\vec{b}_1, \ldots, \vec{b}_k$ is LLL reduced since we know that $\vec{b}_1, \ldots, \vec{b}_{k-1}$ is LLL reduced and $\vec{b}_1, \ldots, \vec{b}_k$ already satisfies the first condition of being LLL reduced. Therefore, $\vec{b}_1, \ldots, \vec{b}_{k-1}$ is always LLL reduced at the beginning (or equivalently, the end) of the WHILE loop. Hence, if the algorithm ever terminates (when $k = m + 1$) the basis it returns will be LLL reduced.

To show that the algorithm halts after polynomially many iterations we will consider the quantity

$$D \stackrel{\text{def}}{=} \prod_{i=1}^{m} \det(\vec{b}_1, \ldots, \vec{b}_i)^2$$

In particular, we will show that D decreases by a factor of more than $\frac{4}{3}$ each time the SWAP subroutine is called and that it is unchanged each item the SIZEREDUCE subroutine is called. By Proposition 1.3, $\det(\mathcal{L})^2 \in \mathbb{Z}$ for any integer lattice, and so we have that $D^2 \in \mathbb{N}$. Thus we can bound the number of calls to SWAP, and hence the number of iterations of the WHILE loop, by the requirement that $(3/4)^{2t}D^2 > 1$, where t denotes the number of iterations.

A call to SIZEREDUCE(k) simply acts by a series of elementary row (determinant ± 1) operations on any partial basis $\vec{b}_1, \ldots, \vec{b}_i$, and thus $\det(\vec{b}_1, \ldots, \vec{b}_i)^2$ is unchanged for all i. Specifically the value of D is unaffected by calls to the SIZEREDUCE subroutine.

A call to SWAP(k) will exchange $\vec{b}_{k-1}$ and $\vec{b}_k$, so $\det(\vec{b}_1, \ldots, \vec{b}_i)^2$ is unchanged for $1 \le i \le k-2$, and also for $k \le i \le m$ since the order of the basis vectors does not affect the determinant of a lattice. Therefore the only term of D that is affected by the swap is $\det(\vec{b}_1, \ldots, \vec{b}_{k-1})^2$. In particular, we are replacing $\det(\vec{b}_1, \ldots, \vec{b}_{k-2}, \vec{b}_{k-1})^2$ with $\det(\vec{b}_1, \ldots, \vec{b}_{k-2}, \vec{b}_k)^2$, or equivalently, replacing D with

$$D \cdot \frac{\det(\vec{b}_1, \ldots, \vec{b}_{k-2}, \vec{b}_k)^2}{\det(\vec{b}_1, \ldots, \vec{b}_{k-2}, \vec{b}_{k-1})^2}$$

Let $\pi(\vec{x})$ denote the projection of $\vec{x}$ onto the orthogonal complement of $\text{span}(\vec{b}_1, \ldots, \vec{b}_{k-2})$. Then, the above is equal to $D \cdot \frac{\|\pi(\vec{b}_k)\|^2}{\|\pi(\vec{b}_{k-1})\|^2}$. By the definition of the Gram-Schmidt procedure, $\pi(\vec{b}_{k-1}) = \vec{b}_{k-1}^*$, and similarly we have that $\pi(\vec{b}_k) = \vec{b}_k^* + \mu_{k,k-1}\vec{b}_{k-1}^*$. Therefore, D is replaced by a quantity that is no more than

$$D \cdot \frac{\|\vec{b}_k^* + \mu_{k,k-1}\vec{b}_{k-1}^*\|^2}{\|\vec{b}_{k-1}^*\|^2}$$

By assumption, $\|\vec{b}_k^* + \mu_{k,k-1}\vec{b}_{k-1}^*\|^2 < \frac{3}{4}\|\vec{b}_{k-1}^*\|^2$, and therefore D is replaced by a quantity that less than $\frac{3}{4} \cdot D$. Hence t, the number of calls to SWAP, must satisfy $(3/4)^{2t}D^2 > 1$, i.e. $-2t + 2\log_{\frac{4}{3}} D > 0$ or equivalently $t < \log_{\frac{4}{3}} D$ less than $2\log_{\frac{3}{4}} D$. Finally, we note that D can be computed in polynomial time and so $\log_{\frac{4}{3}} D$ is only polynomially large (in terms of n), and by combining this with the observation that k must increase every time that SWAP is not called, we can bound the total number of iterations by $\log_{\frac{4}{3}} D + n$ which is certainly only polynomially large. □

A more detailed analysis also reveals that each iteration only takes polynomial time if the lattice is an integer lattice, i.e. $\mathcal{L} \subseteq \mathbb{Z}^n$. Specifically, it can be shown (see [LLL82]) that the LLL

algorithm requires $O(n^4 \log K)$ arithmetic operations, or $O(n^6(\log K)^2)$ bit operations, where $K = \max\{\|\vec{b}_1\|^2, \ldots, \|\vec{b}_m\|^2, 2\}$.

In light of this, the implications of Proposition 1.7 are more significant. Most notably, the *LLL* algorithm guarantees, in polynomial time, a vector $\vec{b}_1$ (the first vector of the reduced basis) whose length is within a factor of $2^{\frac{m-1}{2}}$ of the length of the shortest vector in the lattice. Admittedly, $2^{\frac{m-1}{2}}$ grows quite rapidly as a function of the dimension, m, of the lattice. However, as noted before, we may substitute the value of $\frac{3}{4}$ in the definition of an *LLL* reduced basis with any constant $\frac{1}{4} < \delta < 1$ and this will result in an approximation of $\|\vec{b}_1\| \leq \left(\frac{1}{\delta - \frac{1}{4}}\right)^{\frac{n-1}{2}}$. Thus, by increasing δ we may improve the quality of the (guaranteed) approximation; of course, this also increases the bound on the guaranteed running time.

Furthermore, we should note that the *LLL* algorithm often performs much better in practice than the guarantee in Proposition 1.7. This has contributed to popularity of lattice basis reduction as a tool in cryptanalysis and other computational applications. For instance, standard methods for cracking weak pseudo-random number generators, and deciphering collections of messages that are encrypted via RSA using a small encryption exponent, employ lattice reduction. For a survey of applications of lattice reduction in cryptanalysis, see [JS98] or [NS00]. The original application of the *LLL* algorithm (in [LLL82]) was to finding the minimal polynomial of an algebraic real number that was given by a numerical approximation and to factoring integer polynomials. Since then, the *LLL* algorithm has found many other applications, for instance in discovering integer relations among irrational numbers of the kind used by Bailey *et al.* ([BBP97]) in finding their remarkable formula for π:

$$\pi = \sum_{i=0}^{\infty} \frac{1}{16^i}\left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6}\right)$$

which allows for the computation of an arbitrary bit of π without computing all the previous bits. More recently, Elkies has given an algorithm for finding rational points close to plane curves that uses low-dimensional lattice basis reduction and allows one to find, for instance, the following "near solution" to Fermat's equation: $283059965^3 + 2218888517^3 = 2220422932^3 - 30$, as well as many other's which are too large to write out completely here (see [Elk00]). While the applications of lattice reduction which we shall see in the next chapter are cryptanalytic, they will also demonstrate the versatility of lattice reduction as an algorithmic tool as illustrated by the above examples.

1.3 Provably Hard Lattice Problems

In the previous section, we saw how the *LLL* algorithm can efficiently give an approximation to a reduced basis or an approximation to the shortest vector of a lattice. This may seem to suggest that lattice basis reduction is not a very difficult problem, especially since the *LLL* algorithm often performs much better in practice than the guarantee in Proposition 1.7. Nonetheless, the problem of lattice basis reduction and related lattice problems remain inaccessible in general. Before discussing some results that indicate that lattice reduction is likely to be a hard problem, we will give two definitions that are basic to a discussion of hard lattice problems.

Definition 1.13. *Given a lattice $\mathcal{L} = B\mathbb{Z}^m$, the* Shortest Vector Problem (SVP) *is to find the shortest non-zero vector in $\mathcal{L}$.*

Definition 1.14. *Given a lattice $\mathcal{L} = B\mathbb{Z}^m$ and a target vector $\vec{t} \in \mathbb{R}^n$, the* Closest Vector Problem (CVP) *is to find the closest lattice point to $\vec{t}$.*

Both of these problems are known to be $\mathcal{NP}$-hard, and more recently there have been results showing that finding approximate solutions to these problems is intractable as well. For example, in [Mic98] Micciancio shows that approximating the shortest vector of a lattice to within a factor of $\sqrt{2}$ is $\mathcal{NP}$-Hard. The proof of this fact is lengthy and draws from several other complexity results concerning lattices, so we will provide only a sketch of the proof.

The following lemma is proved in [Mic98].

Lemma 1.15. *For any constant $\epsilon > 0$ there exists a probabilistic polynomial time algorithm that on input 1^k computes a lattice basis $B \in \mathbb{R}^{(m+1)\times m}$, a vector $\vec{s} \in \mathbb{R}^{m+1}$ and a matrix $C \in \mathbb{Z}^{k\times m}$ such that with probability arbitrarily close to one, the following are true:*

- *For every non-zero $\vec{z} \in \mathbb{Z}^m$, $\|B\vec{z}\|^2 > 2$.*
- *For all $\vec{x} \in \{0,1\}^k$ there exists a $\vec{z} \in \mathbb{Z}^m$ such that $C\vec{z} = \vec{x}$ and $\|B\vec{z} - \vec{s}\|^2 < 1 + \epsilon$.*

Definition 1.16. *Let g (the gap function) be a parameter that may vary as a function of the dimension of the lattice that we are considering. Then we define the promise problem $\texttt{GapSVP}_\texttt{g}$ by the following:*

- *(V, d) is a* YES *instance for $\texttt{GapSVP}_\texttt{g}$ if V is a basis for a lattice in $\mathbb{R}^n$ and $d \in \mathbb{R}$, such that $\|V\vec{z}\| \le d$ for some $\vec{z} \in \mathbb{Z}^n \setminus \{\vec{0}\}$.*
- *(V, d) is a* NO *instance for $\texttt{GapSVP}_\texttt{g}$ if V is a basis for a lattice in $\mathbb{R}^n$ and $d \in \mathbb{R}$, such that $\|V\vec{z}\| > gd$ for all $\vec{z} \in \mathbb{Z}^n \setminus \{\vec{0}\}$.*

Definition 1.17. *Let g (the gap function) be a parameter that may vary as a function of the dimension of the lattice that we are considering. Then we define the promise problem $\texttt{GapCVP}'_\texttt{g}$ by the following:*

- *$(V, \vec{y}, d)$ is a* YES *instance for $\texttt{GapCVP}'_\texttt{g}$ if $V \in \mathbb{Z}^{k\times n}$, $\vec{y} \in \mathbb{Z}^k$ and $d \in \mathbb{R}$ such that $\|V\vec{z} - \vec{y}\|^2 \le d$ for some $\vec{z} \in \{0,1\}^n$.*
- *$(V, \vec{y}, d)$ is a* NO *instance for $\texttt{GapCVP}'_\texttt{g}$ if $V \in \mathbb{Z}^{k\times n}$, $\vec{y} \in \mathbb{Z}^k$ and $d \in \mathbb{R}$ such that $\|V\vec{z} - w\vec{y}\|^2 > gd$ for all $\vec{z} \in \mathbb{Z}^n$ and for all $w \in \mathbb{Z}$.*

Theorem 1.18 (Micciancio). *Approximating the shortest vector in a lattice $\mathcal{L}$ to within any constant factor less that $\sqrt{2}$ is $\mathcal{NP}$-Hard (for randomized reductions).*

Proof. We follow the proof given in [Mic98], and give a (randomized) reduction from $\texttt{GapCVP}'_{\frac{2}{\epsilon}}$ to $\texttt{GapSVP}_{\frac{2}{1+2\epsilon}}$, using algorithm whose existence is guaranteed by Lemma 1.15. Since it is shown that $\texttt{GapCVP}'_{\texttt{c}}$ is $\mathcal{NP}$-hard (for any constant c) in [SAS93], this will prove the result.
Let $(N, \vec{y}, d)$ be either a YES instance or a NO instance of $\texttt{GapCVP}'_{\frac{2}{\epsilon}}$. We will give a transformation to an instance $(V, 1+2\epsilon)$ of $\texttt{GapSVP}_{\frac{2}{1+2\epsilon}}$ such that $(V, 1+2\epsilon)$ is a YES instance if $(N, \vec{y}, d)$ is a YES instance, and $(V, 1+2\epsilon)$ is a NO instance if $(N, \vec{y}, d)$ is a NO instance.

We use Lemma 1.15 to obtain a lattice basis B, a vector $\vec{s}$ and a matrix C, satisfying the conditions in the Lemma, and we construct the matrix V:

$$V \stackrel{\text{def}}{=} \left(\begin{array}{c|c} B & -\vec{s} \\ \hline \left(\sqrt{\frac{\epsilon}{d}}N\right)C & -\sqrt{\frac{\epsilon}{d}}\vec{y} \end{array}\right)$$

Suppose that $(N, \vec{y}, d)$ is a YES instance of $\texttt{GapCVP}'_{\frac{2}{\epsilon}}$. By definition, this means that there is an $\vec{x} \in \{0,1\}^k$ such that $\|N\vec{x} - \vec{y}\|^2 \leq d$. Let $\vec{w} \stackrel{\text{def}}{=} \begin{bmatrix} \vec{z} \\ 1 \end{bmatrix}$, where $\vec{z}$ satisfies $C\vec{z} = \vec{x}$ and $\|B\vec{z} - \vec{s}\| < 1 + \epsilon$. (We know that such a $\vec{z}$ exists by Lemma 1.15.) Using $\vec{w}$, we verify that $(V, 1+2\epsilon)$ is a YES instance of $\texttt{GapSVP}_{\frac{2}{1+2\epsilon}}$:

$$\|V\vec{w}\|^2 = \|B\vec{z} - \vec{s}\|^2 + \frac{\epsilon}{d}\|N\vec{x} - \vec{y}\|^2 \leq (1+\epsilon) + \epsilon = 1 + 2\epsilon$$

Now suppose that $(N, \vec{y}, d)$ is a NO instance of $\texttt{GapCVP}'_{\frac{2}{\epsilon}}$. In order to show that $(V, 1+2\epsilon)$ is a NO instance of $\texttt{GapSVP}_{\frac{2}{1+2\epsilon}}$, we need to show that for any non-zero $\vec{w}$, $\|V\vec{w}\|^2 \geq \frac{2}{1+2\epsilon} \cdot (1+2\epsilon) = 2$. Write $\vec{w} = \begin{bmatrix} \vec{z} \\ w \end{bmatrix} \neq \vec{0}$, yielding

$$\|V\vec{w}\|^2 = \|B\vec{z} - w\vec{s}\|^2 + \frac{\epsilon}{d}\|N\vec{x} - w\vec{y}\|^2$$

If $w = 0$, then $\vec{z}$ must be non-zero. This gives that $\|V\vec{w}\|^2 \geq \|B\vec{z}\|^2$ and we know that $\|B\vec{z}\|^2 > 2$ by construction for Lemma 1.15, so we have that $\|V\vec{w}\|^2 > 2$. If $w \neq 0$, then we have $\|V\vec{w}\|^2 \geq \frac{\epsilon}{d}\|N\vec{x} - w\vec{y}\|^2$, and by definition of $(N, \vec{y}, d)$ as a NO instance of $\texttt{GapCVP}'_{\frac{2}{\epsilon}}$, we know that $\|N\vec{x} - w\vec{y}\|^2 > \frac{2}{\epsilon}d$. Therefore $\|V\vec{w}\|^2 \geq \frac{\epsilon}{d}\|N\vec{x} - w\vec{y}\|^2 > \frac{\epsilon}{d} \cdot \frac{2}{\epsilon}d = 2$. $\square$

Another noteworthy complexity result which is responsible in part for motivating much of the material in next chapter is due to Ajtai in [Ajt96]. In this paper, Ajtai gives a randomized problem which, in essence, consists of finding a short vector in the kernel of a matrix modulo q, i.e. a matrix over $\mathbb{Z}_q$ (where "short" refers to length of the vector considered as an element of $\{0, 1, \ldots, q-1\}^n \subseteq \mathbb{Z}^n$). The main result is a proof that the ability to solve random instances of this problem with non-negligible probability implies the ability to: 1) Approximate the shortest vector in any integer lattice up to a polynomial factor, 2) Find the shortest vector in any integer lattice where the shortest vector is n^c-unique (i.e. all vectors that are less than n^c times as long as the shortest vector are simply multiples of the shortest vector, where $n =$ the dimension of the lattice, and c is a constant), and 3) Find a basis B for any integer lattice, such that $\max_i \|\vec{b_i}\|$ is within a polynomial factor of being as small as possible.

We shall discuss a similar result when examining the Ajtai-Dwork cryptosystem. In the meantime, we simply note that the above results provide strong evidence that lattice reduction problems, such as the Shortest Vector Problem, are in fact quite difficult and may be useful in constructing public-key cryptosystems as we shall see in the next chapter.

Chapter 2

Lattice Cryptosystems

In this chapter we will examine the three most notable lattice cryptosystems inspired by Ajtai's average-case/worst-case equivalence result: the Goldreich-Goldwasser-Halevi (GGH), Ajtai-Dwork and NTRU Cryptosystems. With the exception of the Ajtai-Dwork cryptosystem, where a direct connection is proved between deciphering encryptions and determining the shortest vector in a certain class of lattices, the arguments that these cryptosystems are secure are predominantly heuristic. We will occasionally summarize some of these arguments for security, but we will not focus on these as they are inherently informal and for the most part they will prove less significant when we examine the weaknesses of these cryptosystems. The attacks we shall consider also have many heuristic aspects; however, they are valuable examples of the flexibility of lattice reduction and the ingenuity that is necessary when using lattice reduction as an algorithmic tool. The discussion of these attacks will also provide a context for mentioning some standard assumptions and heuristics that are useful when applying lattice reduction in practice.

2.1 The Goldreich-Goldwasser-Halevi Cryptosystem

In [GGH96], Goldreich, Goldwasser and Halevi present a cryptosystem that has come to be known as the GGH cryptosystem. In essence, the cryptosystem works as follows: The public key is a poor basis of a lattice (i.e. a basis with long vectors), and the private key is a reduced basis of the same lattice. Encryption is performed by taking a lattice point corresponding to the plaintext and applying a small random perturbation to obtain a point not in the lattice, but whose closest lattice point is the plaintext. Hence, ciphertexts are instances of the Closest Vector Problem, and the security of the private key depends on the intractability of finding a sufficiently reduced basis for the lattice. The next few sections will describe these steps in more detail.

2.1.1 Generating Keys

At first, it is not clear how to obtain a lattice together with a reduced basis, given that we are assuming that finding a reduced basis is intractable. However, the following observation suggests a method for generating lattice bases together with reduced bases: The volume of the fundamental parallelepiped (i.e. the determinant) of the lattice is unaffected by the choice of basis; therefore, if we chose a collection of vectors that are nearly orthogonal, we obtain a short basis for the lattice that they generate. In particular, instead of generating a lattice and then seeking a reduced basis, we chose a collection of (nearly orthogonal) vectors that will be a reduced basis. One of the methods suggested in [GGH96] is to choose the private basis matrix, R, uniformly at random from $\{-\ell, \ldots, \ell\}^{n\times n}$, where $\ell = 4$ is the recommended parameter. With high probability this will generate a nonsingular matrix, which is important as we will need R^{-1} for decryption. This can be seen by considering R modulo a prime p that divides $2\ell + 1$; since the entries of R are uniformly distributed modulo p, it follows from a basic fact from algebra that R will be non-singular modulo p with high probability and hence non-singular over $\mathbb{Z}$ with high probability. Furthermore, if $\vec{r}_i$ and $\vec{r}_j$ are two distinct columns of R, then $\mathrm{E}\left[\langle \vec{r}_i, \vec{r}_j\rangle\right] = 0$, and

$$\mathrm{Var}\left[\langle \vec{r}_i, \vec{r}_j\rangle\right] = \mathrm{E}\left[\langle \vec{r}_i, \vec{r}_j\rangle^2\right] = \mathrm{E}\left[\left(\sum_{k=1}^{n} (\vec{b}_i)_k (\vec{b}_j)_k\right)^2\right] = \mathrm{E}\left[\sum_{k=1}^{n} (\vec{b}_i)_k^2 (\vec{b}_j)_k^2\right]$$

since $\mathrm{E}\left[(\vec{b}_i)_{k_1}(\vec{b}_j)_{k_1}(\vec{b}_i)_{k_2}(\vec{b}_j)_{k_2}\right] = 0$ for $k_1 \neq k_2$ because the coordinates are chosen independently. Hence $\mathrm{Var}\left[\langle \vec{r}_i, \vec{r}_j\rangle\right] =$

$$\mathrm{E}\left[\sum_{k=1}^{n} (\vec{b}_i)_k^2 (\vec{b}_j)_k^2\right] = n\,\mathrm{E}\left[(\vec{b}_i)_k^2 (\vec{b}_j)_k^2\right] = n \sum_{s=1}^{\ell}\sum_{t=1}^{\ell} s^2 t^2 \cdot \left(\frac{2}{2\ell+1}\right)^2$$

$$= \frac{4n}{(2\ell+1)^2}\left(\frac{\ell(\ell+1)(2\ell+1)}{6}\right)^2 = \frac{n\ell^2(\ell+1)^2}{9}$$

and in particular, $\mathrm{Var}\left[\langle \vec{r}_i, \vec{r}_j\rangle\right] = 400n/9$ when $\ell = 4$. However, this is quite small given that $\mathrm{E}\left[\|\vec{r}_i\|^2\right] = \mathrm{E}\left[\langle \vec{r}_i, \vec{r}_i\rangle\right] = n\sum_{s=1}^{\ell} s^2 \frac{2}{2\ell+1} = \frac{n\ell(\ell+1)}{3}$, and $\mathrm{Var}\left[\|\vec{r}_i\|^2\right] = \mathrm{E}\left[\langle \vec{r}_i, \vec{r}_i\rangle^2\right] - \mathrm{E}\left[\langle \vec{r}_i, \vec{r}_i\rangle\right]^2 =$

$$\mathrm{E}\left[\left(\sum_{k=1}^{n} (\vec{r}_i)_k^2\right)^2\right] - \left(\frac{n\ell(\ell+1)}{3}\right)^2 = \mathrm{E}\left[\sum_{k=1}^{n} (\vec{r}_i)_k^4\right] + \mathrm{E}\left[\sum_{s\neq t}^{n} (\vec{r}_i)_s^2 (\vec{r}_i)_t^2\right] - \frac{n^2\ell^2(\ell+1)^2}{9}$$

$$= n\sum_{a=1}^{n} a^4 \frac{2}{2\ell+1} + n(n-1)\sum_{b=1}^{\ell}\sum_{c=1}^{\ell} b^2c^2\left(\frac{2}{2\ell+1}\right)^2 - \frac{n^2\ell^2(\ell+1)^2}{9}$$

$$= \frac{2n}{2\ell+1}\cdot\frac{\ell(\ell+1)(2\ell+1)(3\ell^2+3\ell-1)}{30} + \frac{4n(n-1)}{(2\ell+1)^2}\cdot\left(\frac{\ell(\ell+1)(2\ell+1)}{6}\right)^2 - \frac{n^2\ell^2(\ell+1)^2}{9} =$$

$$= \frac{n\ell(4\ell^3+8\ell^2+\ell-3)}{45}$$

The fact that $\mathrm{Var}[\langle \vec{r}_i, \vec{r}_j]$ is not very large (relative to the expected lengths of the vectors $\vec{r}_i$) suggests that the $\vec{r}_i$ will be close to orthogonal. However, while it is possible to use bounds from elementary probability, such as Chebyshev's inequality, to choose ℓ and n such that $\frac{|\langle \vec{r}_i, \vec{r}_j \rangle|}{\|\vec{r}_i\| \|\vec{r}_j\|}$ (i.e. the cosine of the angle between $\vec{r}_i$ and $\vec{r}_j$) is smaller than some fixed value with high probability, the parameters suggested by such an analysis are quite pessimistic. Indeed, the approach of [GGH96] is to choose ℓ and n based on numerical experiments. The recommended values are $\ell = 4$ and $n \in \{200, 250, 300, 350, 400\}$.

Once we have such a basis R, we look to apply a "random unimodular transformation" to obtain a non-reduced public basis B. A method for accomplishing this is described in [GGH96], and it consists of applying (i.e. multiplying R on the right by) matrices of the form:

$$V_i = \begin{bmatrix} 1 & & & \star & & & \\ & \ddots & & \vdots & & & \\ & & 1 & \star & & & \\ & & & 1 & & & \\ & & & \star & 1 & & \\ & & & \vdots & & \ddots & \\ & & & \star & & & 1 \end{bmatrix}$$

where the entries of the off-diagonal entries of the i-th column (denoted by $\star$) are chosen randomly and independently from $\{-1, 0, 1\}$ such that $\Pr[\star = 1] = \Pr[\star = -1] = 1/7$. It is not difficult to see that such matrices are always unimodular; however there is a much more subtle question of how many such matrices should be used to ensure that the resulting basis is hard to reduce, while having entries that are not too large. Unfortunately, the intractability of lattice reduction is not sufficiently well understood to make such an analysis feasible. Therefore, the approach of [GGH96] is again to decide the number of matrices V_i to apply based on empirical data. They suggest using $2n$ such matrices and ensuring that each index $i \in \{1, \ldots, n\}$ is used twice.

2.1.2 Encryption/Decryption

We will assume that we have a mapping from messages to n-dimensional integer vectors. Now, given a message $\vec{m} \in \mathbb{Z}^n$, the ciphertext is $\vec{c} = B\vec{m} + \vec{e}$, where $\vec{e}$ is a randomly chosen vector from $\{-\sigma, \sigma\}^n$, for a parameter $\sigma \in \mathbb{Z}$ (in practice $\sigma = 3$). Thus, encryptions are instances of CVP.

To decrypt the ciphertext $\vec{c}$ using the private basis R, we use an approximate closest vector technique due to Babai. Namely, we compute $R^{-1}\vec{c}$, round the entries of the resulting vector to the nearest integer, and then apply R, i.e. $\vec{m}' = R\left\lceil R^{-1}\vec{c}\right\rfloor$. Let U be the unimodular matrix such that $R = BU$. Then

$$R^{-1}\vec{c} = R^{-1}B\vec{m} + R^{-1}\vec{e} = U^{-1}\vec{m} + R^{-1}\vec{e}$$

Therefore, since U^{-1} is an integer matrix, rounding the entries of $R^{-1}\vec{c}$ will give $U^{-1}\vec{m}$ as long as $\|R^{-1}\vec{e}\|_\infty < \frac{1}{2}$. Numerical experiments cited in [GGH96] indicate that this is extremely likely when R is chosen randomly from $\{-\ell, \ldots, \ell\}^{n \times n}$.

2.1.3 Cryptanalysis

The GGH cryptosystem is a very natural application of the closest vector problem as a trapdoor one-way function. However, there are some aspects of the protocol which seem dangerously structured, especially the form of the error vector $\vec{e}$. Indeed, the fact that the error vector is of the form $\vec{e} \in \{-\sigma, +\sigma\}^n$, has proven to be problematic, as we shall see below.

The most significant cryptanalysis of the GGH cryptosystem is given in [Ngu99a]. In this paper Nguyen shows how encryptions reveal information about the plaintexts modulo 2σ, an observation which allows one to simplify the CVP instance presented by a ciphertext and consequently decrypt messages using a basis for $\mathcal{L}$ which need not be as reduced as the private basis R.

Recall that a message $\vec{m} \in \mathbb{Z}^n$ is encrypted as $\vec{c} = B\vec{m} + \vec{e}$, where $\vec{e}$ is a randomly chosen vector from $\{-\sigma, \sigma\}^n$, for a parameter $\sigma \in \mathbb{Z}$. Now let $\vec{s} = (\sigma, \ldots, \sigma)^T \in \mathbb{Z}^n$, and note that

$$\vec{c} + \vec{s} \equiv B\vec{m} \pmod{2\sigma}$$

where the reduction modulo 2σ is componentwise. In particular, since $\vec{c}, \vec{s}$ and B are known we may hope to recover $\vec{m} \pmod{2\sigma}$. Clearly, if B is invertible modulo 2σ, then we can recover $\vec{m}$ $\pmod{2\sigma}$ by computing $B^{-1}(\vec{c} + \vec{s}) \pmod{2\sigma}$. Even if B is not invertible, provided that the kernel of B is not too large, all the possible solutions to this linear system can be found using standard techniques; for example, if we presume that $\sigma = 3$, as is the case in the challenges published on the Internet consisting of one instance of GGH encrypted message, together with a public key, for dimensions 200, 250 300, 350 and 400, then the kernel of B can be computed modulo 2 and modulo 3 using techniques described in [Coh93], allowing all the vectors in the kernel of $B \pmod{6 = 2\sigma}$ to be reconstructed using the Chinese Remainder Theorem. Nguyen cites numerical experiments and several results about the rank of random matrices modulo primes to justify the claim that the kernel of B will consist of relatively few vectors. For instance, none of the challenges has a kernel containing more than 6 vectors. (See [Ngu99a] for the particulars.)

If we have the value of $\vec{m} \pmod{2\sigma}$, call it $\vec{m}_{2\sigma}$, then we know that $\vec{c} - Bm_{2\sigma} = B(\vec{m} - \vec{m}_{2\sigma}) + \vec{e}$, in which case every entry of $\vec{m} - \vec{m}_{2\sigma}$ is divisible by 2σ, i.e. $\vec{m} - \vec{m}_{2\sigma} = 2\sigma\vec{m}'$ for some $m' \in \mathbb{Z}^n$. This allows us to write

$$\frac{\vec{c} - Bm_{2\sigma}}{2\sigma} = B\vec{m}' + \frac{\vec{e}}{2\sigma} \quad \text{or equivalently} \quad \frac{\vec{c} - Bm_{2\sigma}}{\sigma} = 2B\vec{m}' + \frac{\vec{e}}{\sigma}$$

The expression on the left-hand side shows how we may consider the task of finding $\vec{m}'$ as an instance of the closest vector problem, but where the error vector is now $\frac{\vec{e}}{2\sigma}$ instead of $\vec{e}$. In particular, since $\vec{e} \in \{-\sigma, +\sigma\}^n$, we have that $\frac{\vec{e}}{2\sigma} \in \{-\frac{1}{2}, +\frac{1}{2}\}$, i.e. $\|\frac{\vec{e}}{2\sigma}\| = \frac{\sqrt{n}}{2}$ which is significantly smaller than $\sigma\sqrt{n} = \|\vec{e}\|$. The result of this is that this instance of the closest vector problem is significantly easier to solve than version given by the ciphertext. To see this, we note that Babai's CVP approximation (i.e. the decryption method) will yield the correct message as long as the point we are given is within $\frac{1}{2}\,\mathrm{width}(P(\tilde{R}))$ of a lattice point, where $\tilde{R}$ is the basis we are using to decrypt; hence, by producing a CVP instance which is much closer to the lattice, we relax the requirement that $\mathrm{width}(P(\tilde{R}))$ be large, i.e. $\tilde{R}$ need not be as reduced. The right-hand formula shows how we can still write this CVP instance as an integer lattice problem, now with basis $2B$, which is significant because many of the fastest lattice reduction routines are optimized for integer lattices.

Note that we can verify whether the decryption that is obtained, call it $\hat{m}$, is correct by checking if $\vec{c} - B\hat{m} \in \{-\sigma, \sigma\}^n$. This will indicate whether the lattice has been reduced sufficiently, and in the case where there are multiple lattices, i.e. when $B \pmod{2\sigma}$ has a non-trivial kernel, this will allow us to determine which (if any) of the decryptions that is obtained is correct.

Current lattice reduction techniques such as the LLL algorithm and its variants can solve these simplified CVP instances in a reasonable amount of time, and thus in [Ngu99a], Nguyen was able to recover the message for all of the GGH challenges, except for the dimension 400 message, where only $m_{2\sigma}$ was recovered.

2.2 The Ajtai-Dwork Cryptosystem

As we shall see, the Ajtai-Dwork cryptosystem does not give a lattice explicitly in the same way as the GGH scheme. Nonetheless, there are some very natural connections to lattices, including the security proof which shows that decryption is as hard as solving the so-called Unique Shortest Vector Problem which was mentioned at the end of the previous chapter. Before continuing, we note that all computations below are technically with fixed precision real numbers in $2^{-n}\mathbb{Z}$. However, for simplicity of exposition, we will suppress this fact and treat all quantities as infinite precision real numbers.

2.2.1 Generating Keys

The private key will simply be a vector $\vec{u}$ chosen uniformly at random from the n-dimensional ball of radius 1, $\{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$.

Let $m = n^3$ and select m vectors $\vec{v}_1, \ldots, \vec{v}_m$ independently at random according to the distribution $\mathcal{H}_u$, which is defined as follows:

- Let $\mathcal{Q}$ denote the n-dimensional cube $[-n2^n, n2^n]^n$.
- For $i \in \mathbb{Z}$, let H_i denote the $(n-1)$-dimensional hyperplane consisting of the points $\vec{x} \in \mathbb{R}^n$ such that $\langle \vec{x}, \vec{u} \rangle = i$. Thus $H_0 = \vec{u}^{\perp}$ and in general $H_i = \vec{u}^{\perp} + i\frac{\vec{u}}{\|\vec{u}\|}$.
- Choose $\vec{\ell}$ uniformly from $\{\vec{x} \in \mathcal{Q} \mid \langle \vec{x}, \vec{u} \rangle \in \mathbb{Z}\}$. That is, choose a hyperplane H_i with probability proportional to the $(n-1)$-dimensional area of $H_i \cap \mathcal{Q}$, and then chose a point uniformly at random from $H_i \cap \mathcal{Q}$.
- Next, choose a perturbation $\vec{\delta} \stackrel{\text{def}}{=} \sum_{i=1}^{n} \vec{\delta}_i$, where the $\vec{\delta}_i$ are chosen uniformly at random from the n-dimensional sphere of radius n^{-8}.

- Finally, the value of $\mathcal{H}_u$ is given by $\vec{\ell} + \vec{\delta}$.

Hence, the $\vec{v}_i$ are random perturbations (by at most n^{-7}) of points that lie on the $(n-1)$-dimensional hyperplanes that are perpendicular to $\vec{u}$ and whose distance from the origin is an integer multiple of $\|\vec{u}\|$.

The public key will be $(\vec{v}_1, \ldots, \vec{v}_m)$, together with an index $i_0 \in [1, \ldots, n^2]$ such that $\mathrm{width}(P(\vec{v}_{i_0}, \ldots, \vec{v}_{i_0+n-1})) \geq \frac{1}{n}2^n$, where $P(v_{i_0}, \ldots, v_{i_0+n-1})$ denotes the parallelepiped formed by the vectors $v_{i_0}, \ldots, v_{i_0+n-1}$. In [AD96] it is shown that there exists such an $i_0 < n^2$ with high probability.

2.2.2 Encryption/Decryption

Encryption is performed one bit at a time, as follows. The encryption of a '0' is obtained by choosing $r_1, \ldots, r_m$ randomly from $\{0, 1\}$, and then setting $\vec{c} = \sum_{i=1}^{m} r_i \vec{v}_i \pmod{P(v_{i_0}, \ldots, v_{i_0+n-1})}$. Note that a vector $\vec{x}$ can efficiently be reduced modulo a parallelepiped $P(\vec{b}_1, \ldots, \vec{b}_n)$ by computing $\vec{x} - B\left\lfloor B^{-1}\vec{x}\right\rfloor$, where $\lfloor \vec{y} \rfloor$ denotes the vector obtained by rounding all the entries of $\vec{y}$ toward zero. The encryption of a '1' is a point chosen uniformly at random from the parallelepiped $P(v_{i_0}, \ldots, v_{i_0+n-1})$.

Decryption simply involves computing $\langle \vec{c}, \vec{u} \rangle$. If $\langle \vec{c}, \vec{u} \rangle$ is within $\frac{1}{n}$ of an integer, $\vec{c}$ is decrypted as a '0', and otherwise $\vec{c}$ is decrypted as a '1'. Each $\vec{v}_i$ is within n^{-7} of a hyperplane H_i by construction, so $\sum_{i=1}^{m} r_i \vec{v}_i$ is at most a distance $mn^{-7} = n^{-4}$ from a hyperplane H_j, and the vectors of the parallelepiped P are also each within n^{-7} of such a hyperplane. The assumption that the width of P is at least $\frac{1}{n}2^n$, guarantees that even after reducing $\sum_{i=1}^{m} r_i \vec{v}_i$ modulo the parallelepiped $P(v_{i_0}, \ldots, v_{i_0+n-1})$, the resulting ciphertext $\vec{c}$ has the property that $\langle \vec{c}, \vec{u} \rangle$ is within $\frac{1}{n}$ of an integer. On the other hand, if $\vec{c}$ is an encryption of a '1' then we expect the fractional part of $\langle \vec{c}, \vec{u} \rangle$ to be very close to uniformly distributed on $[0, 1)$. Therefore, decryption consists of computing $\langle \vec{c}, \vec{u} \rangle$, and decrypting $\vec{c}$ as a '0' if $\langle \vec{c}, \vec{u} \rangle$ is within $\frac{1}{n}$ of an integer, and as '1' otherwise. Hence, a '0' is always correctly decrypted and a '1' may be incorrectly decrypted as a '0' with probability at most $\frac{2}{n}$.

The remarkable property of this encryption scheme is that it can be shown that the ability to distinguish between encryptions of '0' and encryptions of '1' implies the ability to solve the Unique Shortest Vector Problem for any lattice where the shortest vector is n^8-unique. The proof of this fact is rather involved and quite lengthy, so we cannot give it here. However, we will outline the major steps of the proof in order to give some indication as to how such a reduction is feasible. The details can be found in [AD96].

The first task is to show that the ability to distinguish between encryptions of '0' and '1' (i.e. given t encryptions of $b \in \{0, 1\}$ and t encryptions of $1 - b$, determine with non-negligible probability the value of b) implies the ability to determine whether a collection of encryptions, all of the same bit, represent a '0' or a '1'. This is accomplished by taking the collection of encryptions and partitioning them into several disjoint subsets, and then pretending that these subsets represent valid public keys. If the encryptions all represent '0', then this is very likely to be a valid assumption, since encryptions of '0' are constructed from points that lie close to the hyperplanes H_i, just as the vectors $\vec{v}_i$ of a public key. However, if the encryptions are of '1', then the vectors will be random and therefore they are unlikely to form a valid public key. Bearing this in mind, we generate many encryptions of '0' and of '1' using of each the "public keys" that were constructed from the partitioning of the original encryptions. By assumption, we can distinguish between valid encryptions of '0' and '1' with non-negligible probability, and so if we run our supposed distinguishing algorithm on these encryptions and a non-negligible fraction of the runs of the algorithm correctly distinguish between encryptions of '0' and '1', then we conclude that the "public key" was actually a valid public key, and hence the original encryptions were of '0'. Otherwise, we conclude that the "public key" was not valid, and hence the original encryptions were of '1'.

Now let $\mathcal{L}$ be a lattice which has an n^8-unique shortest vector. A class of random transformations is constructed which, with non-negligible probability, carry the shortest vector of $\mathcal{L}$ to a vector in the unit sphere (just like the private key u in the cryptosystem), while preserving the property that the lattice has an n^8-unique shortest vector. (In fact, these transformations consist of orthogonal transformations together with a scaling factor; hence the n^8-uniqueness is clearly preserved, as orthogonal transformations are length-preserving.) By generating many such transformations and applying them to $\mathcal{L}$, we expect that, for a non-negligible fraction of the resulting lattices, the shortest vector is contained within the unit sphere. If this is the case then the dual lattice of $\mathcal{L}$, $\mathcal{L}^*$, when appropriately transformed (to account for the transformation applied to $\mathcal{L}$) will have the following structure: $\mathcal{L}^*$ contains an $(n-1)$-dimensional sublattice $\mathcal{L}^{*\prime}$ with a basis all of whose vectors are shorter than n^8, and such that the distance between the hyperplane, $H^{*\prime}$, containing $\mathcal{L}^{*\prime}$ and all its cosets (i.e. the translations of the hyperplane $H^{*\prime}$ by the basis vector of $\mathcal{L}^*$ that is not in $\mathcal{L}^{*\prime}$) are all separated by a distance of at least 1 from each other.

Next it is shown that the distribution of $\mathcal{H}_u$ is negligibly far (i.e. indistinguishable) from the distribution ξ induced by choosing a random point on one of the cosets of $H^{*\prime}$ (from within a large cube, as with $\mathcal{H}_u$) and perturbing it in the same manner as in $\mathcal{H}_u$. Hence, the ability to distinguish between $\mathcal{H}_u$ and the uniform distribution implies the ability to distinguish between ξ and the uniform distribution. Assuming this capability, a procedure is constructed that can determine whether a vector in $\mathcal{L}^*$ is contained in $\mathcal{L}^{*\prime}$ or not. By applying this procedure to the differences of many pairs of lattice points, then one can recover a basis for $\mathcal{L}^{*\prime}$, which in turn reveals the shortest vector of $\mathcal{L}$ which lies in the one-dimensional orthogonal complement of $\mathcal{L}^{*\prime}$, by the definition of the dual lattice.

2.2.3 Cryptanalysis

Although the Ajtai-Dwork cryptosystem is very appealing because of its security proof, the fact that each ciphertext only encodes one bit raises some concerns as to its practical applicability. Indeed, the cryptanalysis of Nguyen and Stern, which we shall examine below, only shows that the parameters must be impractically large for the system to be secure in practice. Thus the major achievement of the Ajtai-Dwork cryptosystem, its novel security proof, remains intact. Nonetheless, this attack does highlight the complications that can arise when choosing the parameters for a cryptosystem, even one with a security proof.

In [NS98], Nguyen and Stern present the following attack whose goal is to find the private key $\vec{u}$. We assume that we have a public-key consisting of the vectors $\vec{v}_1, \ldots, \vec{v}_m$, as defined above. Then we construct the $(n+m) \times m$ matrix L_β

$$L_\beta \stackrel{\text{def}}{=} \begin{bmatrix} \beta\vec{v}_1 & \beta\vec{v}_2 & \cdots & \beta\vec{v}_m \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

for a parameter $\beta \in \mathbb{R}$, and let $\mathcal{L}_\beta$ be the lattice generated by the columns of L_β.

By construction, we know that $\langle \vec{v}_i, \vec{u} \rangle$ is within n^{-7} of an integer $V_i \stackrel{\text{def}}{=} \lceil \langle \vec{v}_i, \vec{u} \rangle \rfloor$. Since n^{-7} is quite small, if we can find an integral combination, $\lambda_1\vec{v}_1 + \cdots + \lambda_m\vec{v}_m$, of the $\vec{v}_i$ that is very short, and where the integer coefficients λ_i are not too large, then we are likely to have $\lambda_1 V_1 + \cdots + \lambda_m V_m = 0$. This explains the choice of the lattice $\mathcal{L}_\beta$, since short vectors of $\mathcal{L}_\beta$ correspond exactly to short combinations of the $\vec{v}_i$ where the coefficients are not too large. More precisely, the following theorem is proved in [NS98]:

Theorem 2.1. *Let $\vec{x} = (\beta(\lambda_1\vec{v}_1 + \cdots + \lambda_m\vec{v}_m), \lambda_1, \ldots, \lambda_m)^T$ be a lattice point in $\mathcal{L}_\beta$, i.e. $\lambda_i \in \mathbb{Z}$. If $n^7\|\sum_{i=1}^m \lambda_i\vec{v}_i\| + \sum_{i=1}^m |\lambda_i| < n^7$, then $\sum_{i=1}^m \lambda_i V_i = 0$. In particular, this equality is satisfied if $\beta^2 \geq \frac{1}{2n^7-1}n^{14}$ and $\|\vec{x}\|^2 < \frac{1}{2n^7-1}n^{14}$.*

Thus, every sufficiently short vector in $\mathcal{L}_\beta$ gives $\lambda_1, \ldots, \lambda_m$ satisfying $\lambda_1 V_1 + \cdots + \lambda_m V_m = 0$, and so if we define $\vec{V} = (V_1, \ldots, V_m)^T$, and $\vec{\Lambda} = (\lambda_1, \ldots, \lambda_m)$, we have that $\langle \vec{\Lambda}, \vec{V} \rangle = 0$, i.e. $\vec{\Lambda}$ is orthogonal to $\vec{V}$. Define $\vec{V}^\perp$ to be the collection of all points in $\mathbb{Z}^m$ that are orthogonal to $\vec{V}$. It is not hard to see that $\vec{V}^\perp$ is an $(m-1)$-dimensional sublattice of Z^m and that $(\vec{V}^\perp)^\perp$ is generated by an integer multiple of $\vec{V}$. In particular, if all the greatest common divisor of all the entries of $\vec{V}$ is 1 (which is very likely to be the case) then $(\vec{V}^\perp)^\perp$ is generated by $\vec{V}$. Thus, if we can find $m-1$ linearly independent Λ_j by finding linearly independent short vectors in $\mathcal{L}_\beta$, then we obtain a basis for $\vec{V}^\perp$. This allows us to compute $(\vec{V}^\perp)^\perp = \pm\vec{V}$ using an m-dimensional cross product, i.e. the value of the k-th coordinate of $\pm\vec{V}$ is given by $(-1)^{k-1}$ times the determinant of the k-th $(m-1) \times (m-1)$ minor of the matrix whose rows are the linearly independent vectors Λ_j. Recalling that the entries of $\vec{V}$ are $V_i = \lceil \langle \vec{v}_i, \vec{u} \rangle \rfloor$, we recover a system of approximate linear equations:

$$\begin{bmatrix} \cdots & \vec{v}_1^T & \cdots \\ \cdots & \vec{v}_2^T & \cdots \\ & \vdots & \\ \cdots & \vec{v}_m^T & \cdots \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} \approx \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_m \end{bmatrix}$$

Solving this (over-specified) system using only n values of $\vec{v}_i$ and V_i gives an approximation u' to the private key, u. Let M denote an $n \times n$ matrix whose rows are n distinct $\vec{v}_i^T$. Since the $\vec{v}_i$ are chosen almost uniformly from a large cube, we expect that M^{-1} will have small entries; in particular, we expect that the approximation to $\vec{u}$ will be very good. Moreover, there are many possible choices for M, so we may obtain many approximate solutions and choose the one that performs the best. The empirical evidence from [NS98] suggests that this is not even necessary and that a single choice of M is sufficient. In fact, the empirical results from [NS98] suggest that the Ajtai-Dwork cryptosystem is not secure in practice for $n \leq 32$. Since the public key in this case of $n = 32$ is on the order of 20 megabytes, and ciphertexts encoding a single bit have a length of 6144 bits, they conclude that without significant modification, the Ajtai-Dwork cryptosystem is not practically viable. This being said, the real value in the Ajtai-Dwork cryptosystem was never the promise of practical viability, but rather its novel security guarantee, which is unaffected by this cryptanalysis.

2.3 The NTRU Cryptosystem

The first version of the NTRU cryptosystem was proposed by Hoffstein et al in 1996, and after several refinements was updated in 1998 [HPS98]. This is the version we shall consider below.

The description of the NTRU cryptosystem is given entirely in terms of quotient rings of integer polynomials, however there is a natural connection to lattices and lattice reduction, as we shall soon see.

All computations are performed in the ring $\mathcal{R} = \mathbb{Z}_q[x]/(x^n - 1)$, where $\mathbb{Z}_q$ denotes the integers modulo q.[1] This has the practical advantage that an element $a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$ of $\mathcal{R}$ can be represented as an n-tuple of integers $[a_0, a_1, \ldots, a_{n-1}]$. Using this representation, addition in $\mathcal{R}$ is performed componentwise, and multiplication (which we will denote by $*$) is a circular convolution:

$$a * b = \sum_{k=0}^{n-1} \left(\sum_{\substack{i+j \equiv k \\ \bmod n}} a_i b_j \right) x^k$$

2.3.1 Generating Keys

Let $\mathcal{L}(a, b) \stackrel{\text{def}}{=} \{f \in \mathcal{R} \mid f \text{ has } a \text{ coeffs. equal to 1, } b \text{ coeffs. equal to } -1 \text{ and all other coeffs. equal to } 0\}$

Let d_f be an integer less than $n/2$. Then the private key f is a random element of $\mathcal{L}(d_f, d_f - 1)$. For reasons that will soon be clear, we also require that f be invertible in $\mathcal{R}$, i.e. $f \in \mathcal{R}^\times$, and that f be invertible when considered modulo $p \stackrel{\text{def}}{=} 3$.

Similarly, let d_g be an integer less than $n/2$ and randomly choose $g \stackrel{\text{R}}{\leftarrow} \mathcal{L}(d_g, d_g)$. The public key will be $h \stackrel{\text{def}}{=} f^{-1} * g$. The security of the cryptosystem will rely on the assumption that it is infeasible, given $h = f^{-1} * g$, to find a $f' \in \mathcal{R}^\times$ and $g' \in \mathcal{R}$ satisfying $h = {f'}^{-1} g'$ and possessing small enough coefficients that the decryption algorithm (described below) will still work.

[1] In practice currently, q is either 128 or 256, and n is 251, 347 or 503 (see www.ntru.com).

2.3.2 Encryption/Decryption

To encrypt a message $m \in \{-\frac{p-1}{2}, \ldots, \frac{p-1}{2}\}^n \subseteq \mathcal{R}$, randomly choose $\phi \xleftarrow{\mathrm{R}} \mathcal{L}(d_\phi, d_\phi)$, and compute the ciphertext:

$$c = p \cdot (\phi * h) + m$$

To decrypt the ciphertext, we first compute

$$f * c = p \cdot (\phi * g) + f * m$$

Let us assume that we have chosen the parameters d_f, d_g, d_ϕ and d_m such that, with high probability, the coefficients of $p \cdot (\phi * g) + f * m \pmod{x^n - 1}$ are between $-q/2$ and $q/2$ (before reducing modulo q). In this case, if we "center" $f * c = p \cdot (\phi * g) + f * m \pmod{q}$, by choosing its coefficients between $-q/2$ and $q/2$, and then reduce modulo p we obtain $f * m \pmod{p}$, with only a small probability of error. Recall that f was required to be invertible in $\mathbb{Z}_p[x]/(x^n - 1)$, and call this inverse f_p^{-1}. Finally, if we apply f_p^{-1} and take the result modulo p, we obtain $m \pmod{p}$, and since all coefficients of m are in $\{-\frac{p-1}{2}, \ldots, \frac{p-1}{2}\}$, this allows us to recover m.

In the above decryption procedure, we assumed that d_f, d_g and d_ϕ were such that the coefficients of $p \cdot (\phi * g) + fm \pmod{x^n - 1}$ are between $-q/2$ and $q/2$ with high probability. While it is possible to find appropriate values of d_f, d_g, d_ϕ and d_m using elementary methods from probability theory, the values that are found in this way are quite pessimistic, and so the values given in [HPS98], are based on numerical experiments. For concreteness, we list the three parameter selections from [HPS98]:

n	p	q	d_f	d_g	d_ϕ
107	3	64	15	12	5
167	3	128	61	20	18
503	3	256	216	72	55

More recently, the authors of the NTRU cryptosystem have proposed a new variant where p is actually chosen to be a small polynomial that is relatively prime to $x^n - 1$ (instead of a small integer relatively prime to q). This requires several other modifications to the encryption and decryption procedures, but much of the structure is the same. We refer to the documentation at [HPS] for the details.

2.3.3 Cryptanalysis

As of yet, we have been no mention of lattices in the above description of the NTRU cryptosystem. However, if one considers $p \cdot h$ as a linear map (i.e. an $n \times n$ matrix over $\mathbb{Z}_q$) acting on ϕ considered as an n-dimensional vector in $\{-1, 0, 1\}^n \subseteq \mathbb{Z}_q^n$, then the encryption process can be thought of as perturbing the "lattice point" $(p \cdot h) * \phi$ by m. Thus, given a ciphertext c, the "closest" point to c of the form $(p \cdot h)$, is likely to be at a distance m from the ciphertext. However, the attacks we shall consider are to recover a decryption key f' given the public key h and consider a different, albeit related lattice construction.

Let $H \in \mathbb{Z}_q^{n\times n}$ be the matrix corresponding to the linear map $a \mapsto h * a$ in $\mathcal{R}$, and note that $H = F^{-1}G$ where F and G are the linear maps $a \mapsto f * a$ and $a \mapsto g * a$, respectively. Now consider the $2n$-dimensional Coppersmith-Shamir lattice, $\mathcal{L}^{CS}$, generated by the columns of

$$L^{CS} \stackrel{\text{def}}{=} \begin{bmatrix} I & 0 \\ H & qI \end{bmatrix} = \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ \hline h_0 & h_{n-1} & \cdots & h_1 & q & 0 & \cdots & 0 \\ h_1 & h_0 & \cdots & h_2 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & \cdots & h_0 & 0 & 0 & \cdots & q \end{array}\right)$$

Proposition 2.2. *Let $f \circ g \in \mathbb{Z}_q^{2n}$ denote the concatenation of f and g as vectors in $\mathbb{Z}_q^n$. Then $f \circ g \in L^{CS}\mathbb{Z}^n$.*

Proof. Consider f and g as a vectors in $\{0, \ldots, q-1\}^n$. Similarly, consider the linear map $H : \mathcal{R} \to \mathcal{R}$ defined by $a \mapsto h * a$. H can be thought of as a matrix in $\mathbb{Z}_q^{n\times n}$, or equivalently as a matrix with entries in $\{0, \ldots, q-1\}$. Then $Hf = g + q \cdot \vec{v}$, where $\vec{v} \in \mathbb{Z}^n$. Therefore, we have that $C(f \circ (-\vec{v})) = f \circ g$ is in $\mathcal{L}^{CS} = L^{CS}\mathbb{Z}^n$. □

From the preceding proposition, the general method of attack should be clear. Since f and g have small coefficients by construction, we expect $f \circ g$ to be a short vector in the lattice $\mathcal{L}^{CS}$. Indeed, this attack, introduced by Coppersmith and Shamir, was the first main attack against the earliest version of NTRU (see [DC97]). In light of this attack, the sucurity parameters (n, d_f, d_g, d_ϕ) were adjusted, and in [HPS98] a part of the justification of the security of the NTRU cryptosystem is that the parameters were chosen to make such an attack infeasible using contemporary lattice reduction techniques.

We now turn to a variant of this attack, due to May in [May99], that has forced the creators of the NTRU cryptosystem to increase the recommended parameters yet again.

Recall that the LLL algorithm guarantees that the first vector of the reduced basis is within a factor of $2^{\frac{n-1}{2}}$ of the length of the shortest vector in the lattice. Therefore, if one has a lattice where the second shortest vector is more than $2^{\frac{n-1}{2}}$ times as long as the shortest vector, then the LLL algorithm must return the shortest vector. This case is rather extreme since $2^{\frac{n-1}{2}}$ is very large, even for moderate values of n. However, a similar effect is noticeable for more reasonable lattices. If we denote by λ_1 the length of the shortest non-zero lattice vector, call it $\vec{v}$, and by λ_2 the length of the shortest lattice vector that is linearly independent of $\vec{v}$, then empirically, the quality of the basis returned by lattice reduction algorithms appears to improve as the quantity $\frac{\lambda_2}{\lambda_1}$ gets larger. Therefore, one might try to artificially augment the "gap" between the shortest vector and the second shortest vector in order to obtain shorter vectors via lattice reduction. This is the approach taken by May, which proceeds as follows.

Let $\vec{v} = f' \circ g'$ be a shortest vector in $\mathcal{L}^{CS}$ and let $\sigma \in S_n$ be the cyclic permutation $[a_1, a_2 \ldots, a_n] \mapsto [a_n, a_1, \ldots, a_{n-1}]$. It is not hard to see, from the cyclic structure of L^{CS}, that $\sigma^k(f') \circ \sigma^k(g') \in \mathcal{L}^{CS}$ for all $0 \leq k < n$, and that all these vectors have the same norm. This is the situation we wish to avoid, however, since this means that $\frac{\lambda_2}{\lambda_1} = 1$. Therefore, we consider the lattice generated by the following variant of L^{CS}

$$L^r(\theta) \stackrel{\text{def}}{=} \left(\begin{array}{ccc|ccccccc}
1 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
0 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\hline
\theta \cdot h_0 & \cdots & \theta \cdot h_1 & \theta \cdot q & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\
\theta \cdot h_{r-1} & \cdots & \theta \cdot h_r & 0 & \cdots & \theta \cdot q & 0 & \cdots & 0 \\
h_r & \cdots & h_{r+1} & 0 & \cdots & 0 & q & \cdots & 0 \\
\vdots & & \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\
h_{n-1} & \cdots & h_0 & 0 & \cdots & 0 & 0 & \cdots & q
\end{array}\right)$$

$L^r(\theta)$ is obtained by taking L^{CS} and then multiplying rows $n+1$ though $n+r$ by $\theta \stackrel{\text{def}}{=} q+1$. This has the effect of lengthening all vectors whose $n+1$ through $n+r$ coefficients are not all zero. The hope is that g' will have a unique "run" of r zeros, i.e. that there is exactly one index $i \in [1, 2, \ldots, n]$ such that $g'_i = g'_{i+1} = \cdots = g'_{i+r-1} = 0$. If this is the case, then all of the rotations $\sigma^k(f') \circ \sigma^k(g')$ will be lengthened to have length at least $\sqrt{2d_f - 1 + (q+1)^2 + 2d_g - r}$, except for one of them which will still have length $\sqrt{2d_f - 1 + 2d_g}$.

By using this approach, in conjunction with some other techniques described in [May99], May showed that NTRU encryptions using the small parameter value of $n = 107$ are not secure, and as a result, the NTRU specification now suggests values of $n \in \{251, 347, 503\}$.

In a follow-up paper, [MS01], May and Silverman discuss a natural generalization to this approach where, instead of looking for unique "runs", one chooses a random subset of coordinates with the hope there is a unique cyclic shift of g where all these coordinates are zero. Much of the analysis is similar to the case of the "run" lattice, however this attack has the benefit that it is more difficult to protect against, whereas it is relatively easy to ensure that the longest zero-run in g is not unique.

Finally, we should note the particularly small key-size for the NTRU cryptosystem. In the GGH scheme a complete $n \times n$ matrix (at least $O(n^2)$ bits) consisting of the public basis vectors is given, and in the Ajtai-Dwork cryptosystem, m (n-dimensional) vectors are given representing $O(\log(n2^n) \cdot n \cdot m)$ $O(n^5 \log n)$ bits. However, in the NTRU scheme only $O(n \log q)$ bits are required to obtain (implicitly) a lattice of dimension $2n$. This is undoubtedly one of the reasons the NTRU cryptosystem is still viable today, albeit with larger parameters than the original proposals.

Conclusion

In the preceding chapters we have seen how lattice reduction is on the one hand an intractable problem and on the other hand a very powerful cryptanalytic tool with algorithms that can perform surprisingly well in practice. To a certain extent, this dichotomy of applications illustrates how lattice reduction is not a very well understood problem. For instance, Micciancio's result shows that the shortest vector in a lattice cannot be approximated within a factor of $\sqrt{2}$, whereas the LLL algorithm only guarantees an exponentially large approximation factor. These bounds leave open the question of whether a polynomial approximation to the shortest vector can be achieved in polynomial time, and if so, how small the degree of the approximation factor can be. Of course, the complementary problem is to try to improve the inapproximability results to show that the shortest vector cannot be approximated to within some factor which is an increasing function of the dimension. However, in [GG97], Goldreich and Goldwasser prove that if it can be shown, via a many-one/Karp reduction, that approximating the shortest vector to within a factor of $\sqrt{n}$ is $\mathcal{NP}$-hard, then the polynomial hierarchy collapses. Therefore, it seems unlikely that much stronger inapproximability results about SVP will be found.

We have also seen how cryptographic constructions based on lattice problems such as the GGH and Ajtai-Dwork cryptosystems, require lattices of very high dimension before benefitting from security suggested or implied by the intractability of lattice reduction. This begs the question of whether lower-dimensional lattice problems can be constructed that isolate the very hardest instances of SVP or CVP. An alternative approach suggested by the NTRU cryptosystem. Perhaps it is inevitable that the lattices must have very large dimensions to construct secure encryption schemes but at the same time one may find lattices with a particular structure (such as the Coppersmith-Shamir lattice, $\mathcal{L}^{CS}$) that allow for a compact representation while still retaining the resistance to reduction of the general lattices of the same dimension.

These questions highlight some of the weaknesses of current lattice cryptosystems as well as the gap in knowledge regarding the complexity of lattice problems. Nonetheless, the resilience of the current version of the NTRU cryptosystem and the theoretical significance of results such as the Ajtai-Dwork security proof and Micciancio's inapproximability result offer hope that the intractability of lattice reduction may ultimately provide a viable alternative to the assumptions that integer factorization and the discrete logarithm problem are intractable. On the other hand, all is not lost if advances in lattice reduction techniques improve to the point that no reasonable lattice cryptosystem may be constructed, for as we have seen, lattice reduction is a powerful tool in its own right and is applicable to many problems other than attacks on lattice cryptosystems.

Acknowledgements

First, I should like to thank my advisor Professor Michael Rabin for introducing me to the subject of lattice cryptosystems and for many inspiring conversations as I explored the various facets of this topic. I would also like to thank Professor Noam Elkies for several enlightening discussions about applications of lattice reduction in computational mathematics. Finally, David Molnar pointed me to a few interesting and important results about lattice cryptosystems and the complexity of lattice problems; he also provided some helpful comments on a draft of this thesis.

Bibliography

[AD96] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence, 1996.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems, 1996.

[BBP97] David Bailey, Peter Borwein, and Simon Plouffe. On the rapid computation of various polylogarithmic constants. *Mathematics of Computation*, 66(218):903–913, 1997.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1993.

[CR88] B. Chor and R.L. Rivest. A knapsack-type public-key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, 5(34):901–909, 1988.

[CS93] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York, NY, USA, 2nd edition, 1993.

[DC97] A. Shamir D. Coppersmith. Lattice attacks on ntru. In *Proceedings Eurocrypt'97*, volume LNCS, pages 52–61. Springer-Verlag, 1997.

[Dwo] Cynthia Dwork. Stanford university cs359: Lattices and their applications to cryptography and cryptanalysis (lecture notes: http://theory.stanford.edu/ csilvers/cs359/).

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-30(4):469–472, July 1985.

[Elk00] Noam D. Elkies. Rational points near curves and small nonzero $|x^3 - y^2|$ via lattice reduction. In *Algorithmic Number Theory Symposium (ANTS)*, 2000.

[GG97] Oded Goldreich and Shafi Goldwasser. On the limits of non-approximability of lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(031), 1997.

[GGH96] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(056), 1996.

[HPS] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. http://www.ntru.com.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*, number Theory, pages 267–288, 1998.

[JHS00] J. Pipher J. Hoffstein and J. H. Silverman. Nss: The ntru signature scheme, 2000.

[JS98] Antoine Joux and Jacques Stern. Lattice reduction: A toolbox for the cryptanalyst. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 11(3):161–185, Summer 1998.

[LLL82] Arjen K. Lenstra, H. W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[May99] A. May. Cryptanalysis of ntru, 1999.

[McE78] R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. *JPL DSN Progress Report*, 4244:114–116, 1978.

[MH78] R.C. Merkle and M.E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, IT-24:525–530, 1978.

[Mic] Daniele Micciancio. Mit cs291: Lattices in cryptography and cryptanalysis (lecture notes: http://www.cs.ucsd.edu/users/daniele/cse291fa99.html).

[Mic98] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. In *IEEE Symposium on Foundations of Computer Science*, pages 92–98, 1998.

[MS01] Alexander May and Joseph H. Silverman. Dimension reduction methods for convolution modular lattices. In *CaLC*, pages 110–125, 2001.

[Ngu99a] Phong Nguyen. Cryptanalysis of the goldreich-goldwasser-halevi cryptosystem from crypto '97. In *CRYPTO*, pages 288–304, 1999.

[Ngu99b] Phong Nguyen. La géométrie des nombres en cryptologie, 1999.

[NS98] Phong Nguyen and Jacques Stern. Cryptanalysis of the ajtai-dwork cryptosystem. In *CRYPTO*, pages 223–242, 1998.

[NS00] Phong Nguyen and Jacques Stern. Lattice reduction in cryptology: An update. In *ANTS*, pages 85–112, 2000.

[Rab79] M. O. Rabin. Digital signatures and public key functions as intractable as factorization. *MIT Laboratory for computer science, Technical report MIT/LCS/TR-212*, Jan. 1979.

[SAS93] J. Stern S. Arora, L. Babai and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. pages 724–733, 1993.

[Sho94] P.W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science*, pages 124–134, 1994.