Lattices
&
Cryptology

February 14, 2011

Contents

0.1 Topological Group 2
0.2 Lattice Reduction 4
0.2.1 Shortest Vector Problem (SVP) 4
0.2.2 Successive Minima 6
0.3 Computational Problems 6
0.3.1 LLL Algorithm 7
0.3.2 Approximate CVP Algorithm 8
0.3.3 Babai's rounding technique 10
0.4 Lattices in Cryptology 10
0.4.1 GGH Public Key Cryptosystem 11
0.4.2 Cryptanalysis of GGH Cryptosystem 12
0.4.3 NTRU Cryptosystem 12

Lattices occurred naturally in connection with the geometric number theory, Lie-algebra and group theory. Recently, they have got a significant role in coding theory and cryptology. In this note we will briefly discuss the basics of lattices and their applications in cryptology.

0.1 Topological Group

A **topological group** is a group (G, o) together with a topology on G such that the group's binary operation o and the group's inverse function are continuous functions with respect to the topology.

Remark. In a topological group one may perform a algebraic operations because of group structure and may talk about continuity because of topological structure.

Definition 0.1.1 (**Discrete Group**)**.** Discrete group is a topological group with a discrete topology, i.e. all of whose points are open.

Example 0.1.1. The integers $\mathbb{Z}$ is a discrete subgroup of $\mathbb{R}$. The Gaussian integers $\mathbb{Z}[i] = \{m + ni : m, n \in \mathbb{Z}\}$ is a discrete subgroup of $\mathbb{C}$. The group of n^{th} roots of unity $\{\exp(\frac{2\pi im}{n}) : m = 0, 1, ..., n-1\}$ is a discrete subgroup of S^1 for each positive integer n.

Remark. Any group G can be made into a discrete group by giving G the discrete topology. Therefore, the topology of a discrete group is not very interesting. What is interesting is the study of discrete subgroups of a continuous group like $\mathbb{R}^m$ or $GL(m, \mathbb{C})$.

Let $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n$ be arbitrary vectors in $\mathbb{R}^m$. Denote by $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n)$ the set of all integral linear combinations of the $\mathbf{b}_i$'s:

$$\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}$$

Note. $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n)$ is a subgroup of $\mathbb{R}^m$, but it is not necessary discrete. e.g. $\mathcal{L}(1, \sqrt{2})$ is not discrete(requires proof).

Definition 0.1.2 (**Lattices**)**.** A lattice in $\mathbb{R}^m$ is a discrete subgroup of $\mathbb{R}^m$.

Theorem 0.1.1. *The subgroup $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n)$ is a lattice in either of the following two cases*

1. $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n \in \mathbb{Q}^m$.
2. $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n \in \mathbb{R}^m$ *are linearly independent.*

Proof. ?. □

Definition 0.1.3. Let $L = \mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n)$,be a lattice,

1. L is said to be spanned by the generators $\mathbf{b}_i$'s. When the $\mathbf{b}_i$'s are further linearly independent, we say that $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n$ is a **basis** of the lattice L.

2. Dimension or rank of a lattice L in $\mathbb{R}^m$ is the dimension d of its linear span denoted by $span(L)$. The lattice is said to be **full rank** when $d = m$.

3. If $\mathbf{b}_i$'s are treated as a column vector of length m and B be a $m \times n$ matrix, with $\mathbf{b}_i$'s as columns. Lattice generated by B is given by

$$\mathcal{L}(B) = \{B\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\}$$

4. Basis of a lattice is not unique.

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } B_2 = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$$

   generate the same lattice $\mathbb{Z}^2$. Such bases are called **equivalent**.

5. Any lattice basis of L must have exactly d (dimension of lattice) elements. There always exist d linearly independent lattice vectors; however, such vectors do not necessarily form a basis, as opposed to the case of vector spaces.

Theorem 0.1.2 (**Existence of Lattice Bases**). *Let L be a d-dimensional lattice of $\mathbb{R}^n$. Let $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_d \in L$ be linearly independent vectors. There exists a lower triangular matrix $[u_{i,j}] \in \mathcal{M}_d(\mathbb{R})$ such that the vectors $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d$ defined as $\mathbf{b}_i = \sum_{j=1}^{i} u_{i,j}\mathbf{c}_j$ are linearly independent and such that $L = \mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d)$.*

Corollary. *Any lattice of $\mathbb{R}^n$ has at least one basis.*

Remark. All lattices can be written as $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d)$ for some linearly independent $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d$. So lattices can also be alternatively defined as a nonempty subset L of $\mathbb{R}^m$ such that there exist linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d$ in $\mathbb{R}^m$ such that $L = \mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d)$.

Theorem 0.1.3. *B_1 and B_2 be two bases of a lattice in the matrix form iff $B_2 = B_1U$ for some unimodular matrix U.*

Note. A matrix U is unimodular iff $det(U) = \pm 1$.

Definition 0.1.4. Let $L = \mathcal{L}(B)$ be a lattice of rank n.

- We define **Fundamental Parallelopiped** of the lattice as

$$\mathcal{P}(B) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : \quad 0 \leq x_i \leq 1 \quad \forall i \right\}$$

- The volume of fundamental parallelopiped $\mathcal{P}(B)$ is denoted by $det(L)$ and is called the **determinant** of the lattice L. i.e.

$$det(L) = \sqrt{det(B^T B)}$$

Note. Determinant of a lattice is well defined i.e. it does not depend of the choice of the basis.

0.2 Lattice Reduction

Fundamental theorem of linear algebra states that any finite dimensional vector space has a basis. We have just seen that every lattice possess a basis. Fundamental theorem of linear algebra also assert that every finite dimensional Euclidean Space has an orthonormal basis. Lattice, on the other hand, may not have an orthogonal basis. Goal of lattice reduction is to get a lattice basis, which is not far from being orthogonal.

Definition 0.2.1 (**Gram-Schmidt Orthogonalization Process**)**.** Suppose we have vectors $B = [\mathbf{b}_1|\mathbf{b}_2|....|\mathbf{b}_n] \in \mathbb{R}^{m \times n}$ generating a vector space $V = span(B)$. These vectors are not necessary orthogonal, but we can always find a orthogonal basis $B^* = [\mathbf{b}_1^*|\mathbf{b}_2^*|....|\mathbf{b}_n^*]$ for V as follows:

$$\begin{aligned}
\mathbf{b}_1^* &= \mathbf{b}_1 \\
\mathbf{b}_2^* &= \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^* && \text{where } \mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\langle \mathbf{b}_1^*, \mathbf{b}_1^* \rangle} \\
\mathbf{b}_i^* &= \mathbf{b}_i - \sum_{j<i} \mu_{i,j}\mathbf{b}_j^* && \text{where } \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}
\end{aligned}$$

It can be easily verified that $\mathbf{b}_i^*$s are mutually orthogonal and form a basis of V.

0.2.1 Shortest Vector Problem (SVP)

Given a lattice, find a non zero vector in the lattice such that the norm (Euclidean) of the vector is minimal.
i.e., to find $\mathbf{v} \in \mathcal{L}(B)/\{\mathbf{0}\} \quad s.t. \quad \|\mathbf{v}\| \leq \|\mathbf{w}\|$ for any $\mathbf{w} \in \mathcal{L}(B)/\{\mathbf{0}\}$

Note. Is the SVP well defined ? I.e., is there always a lattice vector whose norm is minimal ? Yes it is. In fact, if

$$\lambda = \inf\{\|\mathbf{v}\| : \mathbf{v} \in \mathcal{L}(B)/\{\mathbf{0}\}\}$$

we will show shortly that there always exists a vector $\mathbf{u} \in \mathcal{L}(B)/\{\mathbf{0}\}$ such that $\|\mathbf{u}\| = \lambda$.

Proposition 0.2.1. *For every lattice basis B and its corresponding Gram-Schmidt orthogonalization B^*, we have*

$$\lambda = \inf\{\|\mathbf{v}\| : \mathbf{v} \in \mathcal{L}(B)/\{\mathbf{0}\}\} \geq min\|\mathbf{b}_i^*\|$$

Note. Now onward we denote $min\|\mathbf{b}_i^*\|$ as β.

Remark (**SVP is well defined**). By definition of

$$\lambda = \inf\{\|\mathbf{v}\| : \mathbf{v} \in \mathcal{L}(B)/\{\mathbf{0}\}\}$$

there exists a sequence $\{\mathbf{v}_i\}$ of lattice vectors such that

$$\lim_{i\to\infty} \|\mathbf{v}_i\| = \lambda$$

Without loss of generality we can assume that

$$\mathbf{v}_i \in \mathcal{B}(\mathbf{0}, 2\lambda) = \{\mathbf{z} : \|\mathbf{z}\| \leq 2\lambda\}$$

Since $\mathcal{B}(\mathbf{0}, 2\lambda)$ is compact, we can find a subsequence $\{\mathbf{w}_i\}$ of $\{\mathbf{v}_i\}$, such that

$$\lim_{i\to\infty} \|\mathbf{w}_i\| = \|\mathbf{w}\| = \lim_{i\to\infty} \|\mathbf{v}_i\| = \lambda$$

Now we will show that $\mathbf{w}$ is a lattice vector. Since

$$\lim_{i\to\infty} \|\mathbf{w}_i - \mathbf{w}\| = 0$$

So for sufficiently large i

$$\|\mathbf{w}_i - \mathbf{w}\| \leq \frac{\beta}{3}$$

$$\|\mathbf{w}_i - \mathbf{w}_j\| \leq \|\mathbf{w}_i - \mathbf{w}\| + \|\mathbf{w} - \mathbf{w}_j\| \leq \frac{2\beta}{3} < \beta$$

So for sufficiently large i, lattice vectors $\mathbf{w}_i$ are equal. So $\mathbf{w} = \mathbf{w}_i \in \mathcal{L}(B)$.

Note. Shortest vector is not unique as if $\mathbf{w}$ is a shortest vector, so is $-\mathbf{w}$. The number of shortest vectors in a lattice is called the **kissing number** of the lattice.

0.2.2 Successive Minima

We know that if $\mathbf{w}$ is a shortest vector of L, then so is $-\mathbf{w}$. So, one must be careful when defining the second-to-shortest vector of a lattice. To overcome this problem, Minkowski defined the other minima as follows.

Let L be a lattice of rank n. We define the i^{th} successive minimum as

$$\lambda_i(L) = \inf\{r : dim(span(L \cap \bar{\mathbf{B}}(\mathbf{0}, r))) \geq i\}$$

where $\bar{\mathbf{B}}(\mathbf{0}, r) = \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x}\| \leq r\}$ is a closed ball of radius r around $\mathbf{0}$.

I.e. i^{th} minimum $\lambda_i(L)$ of a lattice L is the radius of the smallest sphere containing i linearly independent non-zero lattice points.

Note. First successive minimum of L is same as the length of shortest lattice vector and clearly, $\lambda_1(L) \leq \lambda_2(L) \leq \lambda_3(L) \leq ..... \leq \lambda_d(L)$

Theorem 0.2.2 (**Minkowski's First Theorem**). *For any full-rank lattice L of rank n.*

$$\lambda_1(L) \leq \sqrt{n}(det(L))^{1/n}$$

Theorem 0.2.3 (**Minkowski's Second Theorem**). *For any full-rank lattice L of rank n.*

$$\left(\prod_{i=1}^{n} \lambda_i(L)\right)^{1/n} \leq \sqrt{n}\,(det(L))^{1/n}$$

0.3 Computational Problems

Minkowski's first theorem upper bounds the successive minimum. I.e. any lattice L of rank n contains a non-zero vector, length of which is bounded above by $\sqrt{n}(det(L))^{1/n}$. However, it does not provide any way to get that vector. There is no known efficient algorithm to find such short vectors. Following are the most basic computational problems involving short vectors in lattices.

- **Search SVP** : Given a lattice basis $B \in \mathbb{Z}^{m \times n}$, find $\mathbf{v} \in \mathcal{L}(B)$ such that $\|\mathbf{v}\| = \lambda_1(\mathcal{L}(B))$.
- **Optimization SVP** : Given a lattice basis $B \in \mathbb{Z}^{m \times n}$, find $\lambda_1(\mathcal{L}(B))$.
- **Decisional SVP** : Given a lattice basis $B \in \mathbb{Z}^{m \times n}$ and a rational $r \in \mathbb{Q}$, determine if $\lambda_1(\mathcal{L}(B)) \leq r$ or not .

Note. The basis above is restricted to integers only, this is to make input representable into 4 bytes.

Following are the approximation variants of the above problem. For some approximation factor $\gamma \geq 1$, we define

- **Search SVP$_\gamma$** : Given a lattice basis $B \in \mathbb{Z}^{m \times n}$, find $\mathbf{v} \in \mathcal{L}(B)/(\mathbf{0})$ such that $\|\mathbf{v}\| \leq \gamma \, \lambda_1(\mathcal{L}(B))$.
- **Optimization SVP$_\gamma$** : Given a lattice basis $B \in \mathbb{Z}^{m \times n}$, find d such that $d \leq \lambda_1(\mathcal{L}(B)) \leq \gamma \, d$.
- **Promise SVP$_\gamma$** : An instance of the problem is given by a pair (B, r), where $B \in \mathbb{Z}^{m \times n}$ is a lattice basis and a rational $r \in \mathbb{Q}$. In YES instance $\lambda_1(\mathcal{L}(B)) \leq r$. In NO instance $\lambda_1(\mathcal{L}(B)) > \gamma \, r$.

Another fundamental lattice problem is CVP (**closest vector problem**).As before for $\gamma \geq 1$, we have

- **Search CVP$_\gamma$** : Given a lattice basis $B \in \mathbb{Z}^{m \times n}$ and a vector $\mathbf{t} \in \mathbb{Z}^m$, find $\mathbf{v} \in \mathcal{L}(B)/(\mathbf{0})$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \gamma \, \text{dist}(\mathbf{t}, \mathcal{L}(B))$.
- **Optimization CVP$_\gamma$** : Given a lattice basis $B \in \mathbb{Z}^{m \times n}$ and a vector $\mathbf{t} \in \mathbb{Z}^m$, find d such that $d \leq \text{dist}(\mathbf{t}, \mathcal{L}(B)) \leq \gamma \, d$.
- **Promise CVP$_\gamma$** : An instance of the problem is given by a pair $(B, \mathbf{t}, r)$, where $B \in \mathbb{Z}^{m \times n}$ is a lattice basis, $\mathbf{t} \in \mathbb{Z}^m$ and a rational $r \in \mathbb{Q}$. In YES instance $\text{dist}(\mathbf{t}, \mathcal{L}(B)) \leq r$. In NO instance $\text{dist}(\mathbf{t}, \mathcal{L}(B)) > \gamma \, r$.

0.3.1 LLL Algorithm

In this section, we will address an approximation algorithm to the shortest vector problem(SVP), by A. K. Lenstra, H. W. Lenstra and L. Lovasz. This algorithm takes as input a basis of a lattice and and outputs a lattice basis consisting of smaller vectors called a reduced basis. Let us first define what is meant by reduced basis.

Note. For simplicity we will consider full rank lattices only, in this section.

Definition 0.3.1 (**δ-LLL Reduced Basis**). A basis $B = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\}$ is called a δ-LLL Reduced Basis if the following conditions hold:

1. $|\mu_{i,j}| \leq \frac{1}{2}$for $j < i$ and $i \leq i \leq n$.
2. $\delta \|\mathbf{b}_i^*\|^2 \leq \|\mu_{i+1,i}\mathbf{b}_i^* + \mathbf{b}_{i+1}^*\|^2$ for $i \leq i < n$

where $\mathbf{b}_i^*$'s are Gram Schmidt Orthogonalization of $\mathbf{b}_i$'s and $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}$ and $\frac{1}{4} < \delta < 1$.

Note. The second condition in the above definition 0.3.1 can be simplified to

$$\delta\|\mathbf{b}_i^*\|^2 \leq \mu_{i+1,i}^2 \|\mathbf{b}_i^*\|^2 + \|\mathbf{b}_{i+1}^*\|^2$$

$$\|\mathbf{b}_{i+1}^*\|^2 \geq (\delta - \mu_{i+1,i}^2)\|\mathbf{b}_i^*\|^2 \geq (\delta - \frac{1}{4})\|\mathbf{b}_i^*\|^2$$

Taking $\delta = \frac{3}{4}$, we have

$$\|\mathbf{b}_{i+1}^*\|^2 \geq \frac{1}{2}\|\mathbf{b}_i^*\|^2$$

Proposition 0.3.1. *Let $B = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\}$ be a δ-LLL-reduced basis . Then*

$$\|\mathbf{b}_1\| \leq \left(\frac{2}{\sqrt{4\delta - 1}}\right)^{n-1} \lambda(\mathcal{L}(B))$$

.

Remark. First vector of δ-LLL-reduced basis is relatively short.

The LLL-Algorithm

Input: Lattice basis $B = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\} \in \mathbb{Z}^n$
Output: δ-LLL reduced basis for $\mathcal{L}(B)$
Start: Compute $\mathbf{b}_1^*, \mathbf{b}_2^*, ..., \mathbf{b}_n^*$
Reduction Step:
for $i = 2$ to n **do**
for $j = i - 1$ to 1 **do**
$\mathbf{b}_i \leftarrow \mathbf{b}_i - c_{i,j}\mathbf{b}_j$, where $c_{i,j} = \left\lceil \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle \right\rfloor$

Swap Step:
if $\exists\ i\ s.t.\ \delta\|\mathbf{b}_i^*\|^2 > \|\mu_{i+1,i}\,\mathbf{b}_i^* + \mathbf{b}_{i+1}^*\|^2$ **then**
$\mathbf{b}_i \longleftrightarrow \mathbf{b}_{i+1}$
goto **Start**.

Output: $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n$.

Note. $\lceil . \rfloor$ denotes the nearest integer function.

Note. Let $M = \max\{n, \log(\max_i \|\mathbf{b}_i\|)\}$. It can be shown that the running time of the above algorithm is polynomial in M.

0.3.2 Approximate CVP Algorithm

Closest vector problem states that given a basis $B \in \mathbb{Z}^{m \times n}$ of a lattice and a target vector $\mathbf{w} \in \mathbb{R}^m$, find a point $\mathbf{x} \in \mathcal{L}(B)$ such that $\forall \mathbf{y} \in \mathcal{L}(B)$, $\|\mathbf{x} - \mathbf{w}\| \leq \|\mathbf{y} - \mathbf{w}\|$. In this section, we will discuss an algorithm, known as the **Nearest Plane Algorithm**, due to L. Babai, which will address the search variant of approximate CVP.

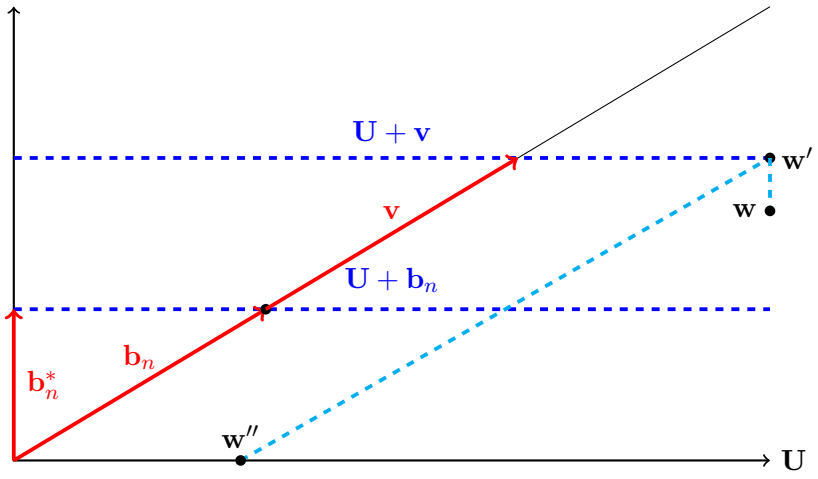

Figure 1: Pictorial representation of Babai nearest plane algorithm

Remark. The distance from a vector $\mathbf{w} \in \mathbb{R}^m$ to a closest lattice vector $\mathbf{v} \in L$ can be quite large compared with the lengths of short vectors in the lattice. E.g. let L be a lattice in $\mathbb{R}^2$ with basis $(1, 0)$ and $(0, 1000)$. Then $\mathbf{w} = (0, 500)$ has distance 500 from the closest lattice point, despite the fact that the first successive minimum is 1.

Babai's Nearest Plane Algorithm

The algorithm uses induction on the dimension n of the lattice. The idea is as follows:
Consider a plane (vector space) generated by $(n-1)$ lattice vectors. Find the translated plane at each lattice point. Choose the one which is nearest to the target vector. Inductively apply the algorithm to the sublattice generated by those $n-1$ vectors and to the new translated target vector.

More precisely, let $U = span\{\mathbf{b}_1, ..., \mathbf{b}_{n-1}\}$ and let $L' = L \cap U$ be the sublattice spanned by $\{\mathbf{b}_1, ..., \mathbf{b}_{n-1}\}$. Now to find a vector $\mathbf{v}$ such that plane $U + \mathbf{v}$ is nearest to the target $\mathbf{w}$. Now take the new target vector $\mathbf{w}'' = \mathbf{w}' - \mathbf{v}$, where $\mathbf{w}'$ is projection of $\mathbf{w}$ to the plane $U + \mathbf{v}$. Inductively work out closest vector $\mathbf{v}'$ to $\mathbf{w}''$ in L' and output $\mathbf{v} + \mathbf{v}'$.

Algorithm

INPUT: Basis $B = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \mathbb{Z}^{m \times n}$ and the target vector $\mathbf{w} \in \mathbb{Z}^m$.
OUTPUT: A vector $\mathbf{x} \in \mathcal{L}(B)$ such that $\|\mathbf{x} - \mathbf{w}\| \leq 2^{\frac{n}{2}} \, dist(\mathbf{w}, \mathcal{L}(B))$.

1. Run δ-LLL on B with $\delta = \frac{3}{4}$
2. $\mathbf{b} \longleftarrow \mathbf{w}$
   for $j = n$ to 1 **do**
      $\mathbf{b} \longleftarrow \mathbf{b} - c_j \mathbf{b}_j$ where $c_j = \lceil \langle \mathbf{b}, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle \rfloor$
   Output $\mathbf{w} - \mathbf{b}$.

Note. Above algorithm runs in polynomial time in the input size.

0.3.3 Babai's rounding technique

This is a computationally simpler alternative method to the nearest plain algorithm. Let $\mathbf{b}_1, \mathbf{b_2}, ..., \mathbf{b}_n$ be a basis for a lattice in $\mathbb{R}^n$ and let $\mathbf{w} \in \mathbb{R}^n$ be a target vector. Then

$$\mathbf{w} = \sum_{i=1}^{n} l_i \mathbf{b}_i$$

with $l_i \in \mathbb{R}$. In rounding technique we simply approximate coefficients to their nearest integers, and output

$$\mathbf{v} = \sum_{i=1}^{n} \lfloor l_i \rceil \mathbf{b}_i$$

Babai has proved that the solution is within an exponential constant of the correct answer if the basis is LLL-reduced.

0.4 Lattices in Cryptology

By now, we have seen some of the properties and the computational problems associated with the lattices. In fact, lattices have been explored since the late 18^{th} century by mathematicians especially by the number theorists. Recently, in 1980's lattices were used in cryptanalysis. However, it was the seminal work, in 1996, of Ajtai that led the lattices to become the cynosure of cryptographers' eyes.

Lattice based cryptosystems became even more important due to vulnerability of factorization based cryptosystem against quantum computers. In this section, we will discuss two public key cryptosystems based on lattice. We will start with GGH cryptosystem, which is most concise and elegant scheme, though it has been subject to cryptanalytic attacks. Then we will discuss NTRU cryptosystem, which is most practical lattice-based cryptosystem.

0.4.1 GGH Public Key Cryptosystem

This cryptosystem was developed by Goldreich, Goldwasser and Halevi in 1997. Security of this cryptosysytem is based on the hardness of closest vector problem in a lattice.

The basic idea is that, given any basis for a lattice it is easy to generate a vector which is close to a lattice point (by taking a lattice point and adding small error to it), however it is hard to get original lattice point given that close vector and an arbitrary lattice basis.

Mathematical description of this cryptosystem is as follows:

Let L a full rank lattice of rank n.

Private key: A "good" lattice basis B. Good in the sense of consisting of short(nearly orthogonal) vectors, that allows to solve certain instances of CVP efficiently.

Public key: A "bad" basis H for the same lattice $\mathcal{L}(H) = \mathcal{L}(B)$. Bad is in sense of being worst possible basis from a cryptanalyst's point of view.

Note. We can take $H = UB$, where U is a unimodular matrix chosen appropriately. Micciancio suggested that $H = \mathrm{HNF}(B)$.

Encryption: Encode a message $\mathbf{m} = [\lambda_1, \lambda_2, ..., \lambda_n] \in \mathbb{Z}^n$ to a lattice point $\mathbf{v} = \mathbf{m}.H$ and add a short noise vector $\mathbf{r}$ i.e. the cipher text is $\mathbf{c} = \mathbf{v} + \mathbf{r}$.
Decryption : For description one computes

$$\mathbf{c}B^{-1} = (\mathbf{m}.H + \mathbf{r})B^{-1} = \mathbf{m}.UBB^{-1} + \mathbf{e}B^{-1} = \mathbf{m}U + \mathbf{e}B^{-1}$$

The term $\mathbf{e}B^{-1}$ being small, can be removed using Babai's rounding technique. Finally compute $\mathbf{m} = \mathbf{m}UU^{-1}$ to get the message.

Example 0.4.1. Let L be a lattice in $\mathbb{R}^2$ with a basis B.

Private key $B = \begin{bmatrix} 7 & 0 \\ 0 & 3 \end{bmatrix}$ and so $B^{-1} = \begin{bmatrix} \frac{1}{7} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$. Let $U = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$ be a unimodular matrix.

Public Key $H = UB = \begin{bmatrix} 14 & 9 \\ 21 & 15 \end{bmatrix}$

Let the message vector $\mathbf{v} = [3, -7]$. and the error vector $\mathbf{e} = [1, -1]$

Encryption: Cipher text $\mathbf{c} = \mathbf{v}B + \mathbf{e} = [104, -79]$

Decryption: $\mathbf{c}B^{-1} = \left(\frac{-104}{7}, \frac{-79}{3}\right)$. Which is rounded to $(-15, -26)$. Now

one can recover message as

$$\mathbf{m} = (-15, -26)U^{-1} = (3, -7)$$

0.4.2 Cryptanalysis of GGH Cryptosystem

Computing a Private Key

Running a lattice basis reduction algorithm on the public basis H may give us a sufficiently good basis to solve required CVP efficiently.

Note. Lattices with sufficiently large dimension will not be vulnerable to such attacks.

Computing the Plain Text

One may try all possible error vectors $\mathbf{e}$'s until $\mathbf{c} - \mathbf{e}$ is lattice vector. Alternatively one can compute $\mathbf{c}(H)^{-1} = \mathbf{m} + \mathbf{e}(H)^{-1}$ and try to work out the possible values $\mathbf{m}$ for some entries of $\mathbf{e}(H)^{-1}$.

Note. For avoiding such attacks one should use appropriate randomized padding schemes for message encoding.

Solving CVP Directly

One can use the Babai nearest plane algorithm to solve CVP with $\mathbf{c}$ as a target vector and with public basis H.

Note. For avoiding such attacks one should use large dimension lattice and error vectors should not taken to be very small.

0.4.3 NTRU Cryptosystem

This cryptosystem is due to Hoffstein, Pipher and Silverman. The original work was framed in term of polynomial rings. However, it can also be described using special types of lattices. We will first describe it in term of polynomial rings.

Let us consider an integer $N \geq 1$ and two moduli p and q, and let

$$R = \frac{\mathbb{Z}[x]}{x^N - 1} \qquad R_p = \frac{\mathbb{Z}_p[x]}{x^N - 1} \text{ and } R_q = \frac{\mathbb{Z}_q[x]}{x^N - 1}$$

be the polynomial rings. Define,

$$\mathcal{T}(d_1, d_2) = \left\{ a(x) \in R : \begin{array}{l} a(x) \text{ has } d_1 \text{ coefficients equal to } 1 \\ a(x) \text{ has } d_2 \text{ coefficients equal to } -1 \\ a(x) \text{ has all others coefficients equal to } 0 \end{array} \right\}.$$

Parameters: Choose (N, p, q, d) with N and p prime, $gcd(p, q) = gcd(N, q) = 1$, and $q > (6d + 1)p$.

Key Creation: Choose $f(x) \in \mathcal{T}(d + 1, d)$ and $g(x) \in \mathcal{T}(d, d)$. Let $F_q(x) =$ inverse of $f(x)$ in R_q , $F_p(x) =$ inverse of $f(x)$ in R_p and $h(x) = F_q(x).g(x)$.

$$\textbf{private key:} \quad f(x) \text{ and } \textbf{public key:} \quad h(x)$$

Encryption: Encode plaintext as $m(x) \in R_p$ and choose a random(ephemeral key) $r(x) \in \mathcal{T}(d, d)$.

$$\textbf{Ciphertext } e(x) \equiv p.r(x) * h(x) + m(x) \,(\, mod\, q)$$

Decryption:

$$a(x) \equiv f(x) * e(x) \,(mod\, q\,) \in R_q$$
$$a(x) = \text{ center-lift } a(x) (mod\, q\,) \in R$$
$$m(x) \equiv F_p(x) * a(x) \,(\, mod\, p\,)$$

Note.

- The condition $q > (6d + 1)p$ ensures that the decrypted message will be same as the plain text message.
- Ephemeral key is generated as a hash of the plaintext.
- NTRU is in fact a lattice-based public key cryptosystem, because underlying the convolution polynomial ring

  $$\frac{\mathbb{Z}[x]}{x^N - 1} \,(\, mod\, q\,)$$

  is a Convolution Modular Lattice.
- The security of NTRU rests on the difficulty of solving CVP in these lattices.

Example 0.4.2 (**Example of NTRU**)**.** Let

$$(N,\, p,\, q,\, d) \,=\, (7, 3, 41, 2)$$

We have

$$41 = q > (6d + 1)p = 39$$

Let

$$f(x) = x^6 - x^4 + x^3 + x^2 - 1 \in \mathcal{T}(3, 2)$$

and

$$g(x) = x^6 + x^4 + -x^2 - x \in \mathcal{T}(2,2)$$

So,

$$F_q(x) = f(x)^{-1}(\,mod\, q) = 8x^6 + 26x^5 + 31x^4 + 21x^3 + 40x^2 + 2x + 37 \in R_q$$

and

$$F_p(x) = f(x)^{-1}(\,mod\, p) = x^6 + 2x^5 + x^3 + x^2 + x + 1 \in R_p$$

Private Key $= (f(x), F_p(x))$
Public Key $= F_q(x) * g(x) = 20x^6 + 40x^5 + 2x^4 + 38x^3 + 8x^2 + 26x + 30 \in R_q$

Let a message (**plain text**) $m(x) = -x^5 + x^3 + x^2 - x + 1$ and the take the ephemeral key $r(x) = x^6 - x^5 + x + 1$.

Encryption

cipher text $e(x) = pr(x)*h(x)+m(x) \equiv 31x^6+19x^5+4x^4+2x^3+40x^2+3x+25$

Decryption

$$f(x) * e(x) \equiv x^6 + 10x^5 + 33x^4 + 40x^3 + 40x^2 + x + 40(\,mod\, q\,)$$

Center lifting above modulo q we get

$$a(x) = x^6 + 10x^5 - 8x^4 - x^3 - x^2 + x - 1 \in R$$

$$F_p(x) * a(x)(mod\, p\,) = 2x^5 + x^3 + x^2 + 2x + 1(mod\, p\,)$$

Center-lifting above modulo p, we get

$$m(x) = -x^5 + x^3 + x^2 - x + 1$$