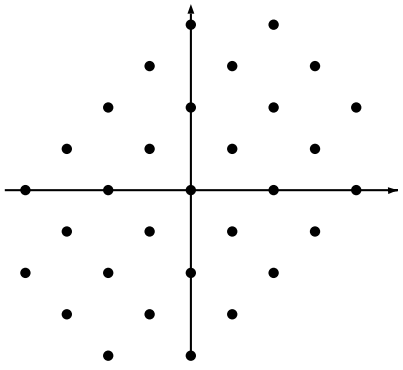
COS 533: Advanced Cryptography — Princeton University
Lecture 15 (November 8, 2017)
Lecturer: Mark Zhandry — Scribe: Kevin Liu

Notes for Lecture 15

1 Lattices

A lattice looks something like the following. You can think it as a regular arragement of points. There are two ways to define lattices precisely. First one is a lattice is a discrete subgroup of $\mathbb{R}^n$ where the group of $\mathbb{R}^n$ is a group with all vectors in $\mathbb{R}^n$ associated with a elementwise addition $+$. A little more useful definition for our purposes is the following: a lattice equals to a set of integer linear combinations of some linearly independent basis $B = \{b_1, b_2, \cdots, b_n\}$. Usually we use the notation $\mathcal{L}(B)$ to denote the lattice that is defined by the basis B, or in other words :



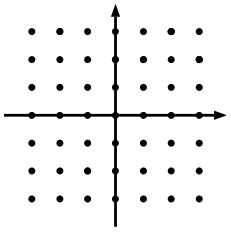
$$\mathcal{L}(B) \;=\; \left\{ \sum_{i=1}^{n} x_i b_i \;\middle|\; x \in \mathbb{Z}^n \right\} = \{ B \cdot x \,|\, B = [b_1, b_2, \cdots, b_n] \text{ and } x \in \mathbb{Z}^n \}$$

So the lattice is spanned by the basis except the coefficients are integers instead of real numbers which span the whole space $\mathbb{R}^n$. Here we denote the basis as a matrix B and any vectors in $\mathcal{L}(B)$ can be written as $B \cdot x$ where x is a integer vector.

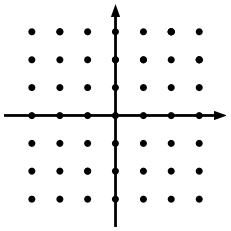
Here we require the number of linearly independent vectors in B equals to the dimension of the space $\mathbb{R}^n$. It can not be more because the vectors should be linearly independent. But you can actually consider the case where the number of vectors in smaller than the dimension. For example, in the above 2D example, we can let $B = (1, 0)^T$ and $\mathcal{L}(B)$ is now all the integer points lie on the line $y = 0$.

Here are some examples:

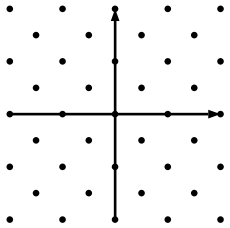
Example 1 The vector $b_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $b_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. It is easy to see that $\mathcal{L}(B) = \mathbb{Z}^2$.



Example 2 Here $B = \begin{pmatrix} 1 & 1 \\ 3 & 2 \end{pmatrix}$. It is easy to verify $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ are both in $\mathcal{L}(B)$. So $\mathcal{L}(B)$ is still $\mathbb{Z}^2$ but it has a different basis. So *the same lattice can be defined by different bases.*



Example 3 Here $B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. $\mathcal{L}(B)$ looks like this and both unit vectors in x and y direction are not in the lattice.



Example 4 Consider $\mathbb{R}^1$ and $b_1 = 1, b_2 = \sqrt{2}$. It is not a lattice because it contradicts the definition where it should be a discrete subgroup of $\mathbb{R}$. Because the sequence $(\sqrt{2}-1)^n$ converges to 0 and $(\sqrt{2}-1)^n$ is a point in the set just by binomial expansion, the set is dense around 0. However, when $B = \{1, 4/3\}$, it is a lattice and the lattice $\mathcal{L}(B) = \{1/3 \cdot x \,|\, x \in \mathbb{Z}\}$.

2 Equivalent Bases of Lattices

The previous example 1 and example 2 highlights the fact that the same lattice can be defined by multiple bases. Now let us try to characterize when two bases define the same lattice. First it will be useful to think about a linear algebra vector space. A vector space $\mathcal{V}(B)$ spanned by basis B is $\mathcal{V}(B) = \{B \cdot x \,|\, x \in \mathbb{R}^n\}$. For vector spaces, $\mathcal{V}(B) = \mathcal{V}(B')$ if and only if there exists an invertible matrix U such that $B = UB'$.

For lattices, there is a similar requirement. But the requirement should be different because there exists an invertible matrix U for matrices in example 1 and example 3 but they do not define the same lattices. So we need some restriction on the invertible matrix.

Definition 1 *We say the matrix $U \in \mathbb{Z}^n$ is unimodular if the determinant of U is either 1 or -1.*

And it turns out that this is the right analog for invertible matrices for lattices. We have the following theorems:

Theorem 2 *Let U be a unimodular matrix. Then U^{-1} is well defined and it is also a unimodular matrix.*

Here the inverse is defined in real vector space but it turns out that U^{-1} also have integer coefficients and determinant ± 1. Let us prove the theorem.

Proof. First, because $UU^{-1} = I$, we have $|UU^{-1}| = |U| \cdot |U^{-1}| = 1$. So given $|U| = \pm 1$, the determinant of U^{-1} is also ± 1.

By the definition of matrix inverse, $U^{-1} = \frac{1}{|U|} \cdot \mathsf{adj}(U)$. Each entry of $\mathsf{adj}(U)$ is a determinant of a submatrix of U so each entry of U^{-1} is still integer.

So U^{-1} is a integer matrix with ± 1 determinant which says it is unimodular. □

Now let us come to another theorem about when two bases define the same lattice.

Theorem 3 *If $B, B' \in \mathbb{R}^{n \times n}$ are full rank (the columns are linearly independent), then $\mathcal{L}(B) = \mathcal{L}(B')$ if and only if there exists a unimodular matrix U such that $B = B'U$.*

Proof. For the $\Rightarrow$ direction, if $\mathcal{L}(B) = \mathcal{L}(B')$, we know that $B = [b_1, b_2, \cdots, b_n]$ and $b_1, \cdots, b_n$ are in the lattice $\mathcal{L}(B')$. It means that there exists an integer matrix $U \in \mathbb{Z}^n$ such that $B = B' \cdot U$. So we only need to show $|U| = 1$. Also we know that there exists an integer matrix U' such that $B' = B \cdot U'$ by the same argument. So then we have $B = B'U = BU'U$, or equivalently, $I = UU'$. Given U', U are integer matrix, their determinants are also integers. The only possibility is their determinants are either 1 or -1.

Then for $\Leftarrow$ direction, suppose $y \in \mathcal{L}(B)$, we know that $y = Bx$ for some integer vector $x \in \mathbb{Z}^n$. Given U is unimodular, we have $y = B'Ux = B'(Ux)$ where Ux is also an integer vector. So $\mathcal{L}(B) \subseteq \mathcal{L}(B')$. By symmetry, $\mathcal{L}(B') \subseteq \mathcal{L}(B)$ which implies $\mathcal{L}(B) = \mathcal{L}(B')$. □

3 Hard Problems on Lattices

We have already defined the concept of lattices. Next, for cryptography, here are some usual hard problems we think about on lattices. For cryptographic usage, we usually assume the basis of a lattice is integral as we do not want to deal with real numbers.

Definition 4 (Shortest Vector Problem (SVP)) *Given a basis $B \in \mathbb{Z}^{n \times n}$, find a vector x such that*

1. *x is a non-trivial vector ($x \neq 0$) and $x \in \mathcal{L}(B)$;*
2. *$|x|_2$ is minimized.*

In other words, given a lattice defined by B, we want to find the shortest vector that is not origin. Let us define $\lambda_1(B) =$ length of the shortest vector in lattice that is not origin, which you will see the meaning of the definition next lecture.

The other computational problem that sometimes we will think about is the closest vector problem which is similar but slight different.

Definition 5 (Closest Vector Problem (CVP)) *Given a basis $B \in \mathbb{Z}^{n \times n}$ and a target vector $t \in \mathbb{Z}^n$, find a vector x such that*

1. *$x \in \mathcal{L}(B)$;*
2. *$|x - t|_2$ is minimized.*

Here given a lattice and a target vector, you want to find the closet vector to t in the lattice. In addition to the problem, we can define relaxations for these problems, the approximate versions. For some ratio $\gamma > 1$, we can define SVP_γ and CVP_γ as follows,

Definition 6 (SVP_γ) *Given basis B, find a non-trivial vector x in the lattice such that $|x|_2 \leq \gamma \cdot \lambda_1(B)$.*

Definition 7 (CVP_γ) *Given basis B and a target t, find a vector x in the lattice such that $|x - t|_2 \leq \gamma \cdot \mathsf{dist}(\mathcal{L}(B), t)$ where the distance is defined as $\min_{y \in \mathcal{L}(B)} |y - t|_2$.*

Moreover, we can even have more relaxed problems — the decisional versions of these problems, GapSVP_γ and GapCVP_γ.

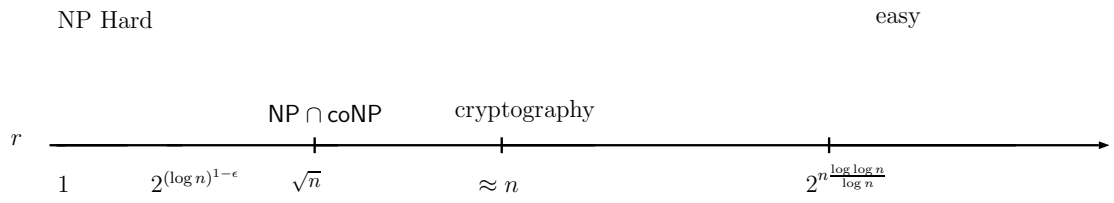
Definition 8 (GapSVP_γ) *Given basis B and some real number $s > 0$, decide either $\lambda_1(B) \leq s$ or $\lambda_1(B) > \gamma \cdot s$.*

The problem is that we should tell whether the shortest vector in the lattice is small or not small at all and we do not care what happens between the two cases. And similarly we have GapCVP_γ,

Definition 9 (GapCVP_γ) *Given basis B, a target t and some real number $s > 0$, decide either $\mathsf{dist}(\mathcal{L}(B), t) \leq s$ or $\mathsf{dist}(\mathcal{L}(B), t) \leq \gamma \cdot s$.*

When the problems is only in two dimensions, they are quite easy. But our lattice may not be two dimensional, let us look at the difficulty of solving these problems in higher dimensions n whch is usually the problem size or the security parameter. In the next picture, you can think of γ is a function of n,

Complexity Landscape for GapSVP_γ



- For $r = \Omega(2^{n \log\log n/\log n})$, the problem is easy;
- For $1 \leq r \leq 2^{(\log n)^{1-\epsilon}}$ and $\epsilon > 0$ (r is at least constant and smaller than any polynomial), the problem is NP Hard where you can reduce Knapsack problem to it;
- For $r = \Theta(\sqrt{n})$, the problem is in $\mathsf{NP} \cap \mathsf{coNP}$. Assuming polynomial hierarchy does not collapse, $\mathsf{NP} \neq \mathsf{coNP}$. Here we do not expect the problem is NP Hard but still we think it is hard;
- When r is about n, here is where cryptography happens.

4 Why Lattices are Useful

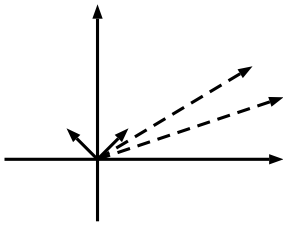
Here is a sketch how you can sign a message using lattices. To motivate this, let us consider the following: we attempt to solve CVP_γ given basis B and a target t, what we can do is to use "lattice rounding" analogous to integer rounding and hope it is close. Here is what we do: compute $B^{-1}t$; If t is in $\mathcal{L}(B)$, then $B^{-1}t$ is actually a integer vector; If t is not in the lattice, we are going to round $B^{-1}t$ to an integer vector, say $[B^{-1}t]$ and $v = B \cdot [B^{-1}t]$ will be a vector in the lattice.

Suppose all entries of B are bounded by δ, then now we can bound the distance:

$$|v - t|_2 = \left|t - B \cdot [B^{-1}t]\right|_2 = \left|BB^{-1}t - B \cdot [B^{-1}t]\right| = \left|B \cdot \left(B^{-1}t - [B^{-1}t]\right)\right|_2$$

And we know each entry of $B^{-1}t - [B^{-1}t]$ is between $-1/2$ and $1/2$, so the norm of $v - t$ is bounded by $\sqrt{n(n\delta/2)^2} = n^{1.5}\delta/2$.

This is our lattice rounding algorithm. The point of this algorithm is that the hardness of CVP_γ depends on basis "quality". In the picture, one pair of vectors are short but the other pair are longer. When applying the rounding algorithm, the short pair will get better results.



This will allow us to do a simple signature scheme. Here is the idea:

- The secret key of the scheme will be some "good" basis B where we just choose some short vectors in the space;
- The public key is a "bad" basis B', or in other words, we choose some large unimodular matrix U and get $B' = B \cdot U$ where the entry of B' is going to be large;
- To sign, we first map the message m into the space to get t (for example, apply a cryptographic hash function to m). The signature is just the CVP solution σ of the point t.
- To verify, we check σ is in the lattice $\mathcal{L}(B')$ and it is close to t which is computed from m.

The idea is that given only the bad basis B', it is hard to find a close vector to t. But it is relatively easy when given the good B. But we need to be careful about how to chose the mapping from the message space to the lattice. For example, we know that given a CVP_γ oracle, we can solve SVP_γ in polynomial time. Suppose the way we map the message is just take the message and interpret it as a sequence of coordinates in the space. Consider the signature experiment, the attacker submits a message m and gets a signature. Basically the attacker gets here is an oracle to CVP_γ as the mapping from messages to points in the lattices is straightforward. And the attacker can solve SVP_γ in polynomial time. By working harder, the attacker can even find a short basis for the lattice which will completely break the scheme. If you use a well chosen cryptographic hash function, then the attacker does not have control of what the target is. Then eventually the scheme will work.

Here are reasons why we use lattices.

1. It is an alternative computational hard problem. Maybe tomorrow it turns out that the elliptic curves are broken. If we have lattices, hopefully we can get things running. And this is extremely important for quantum, because with quantum computers, one can actually break elliptic curves.
2. It is really fast. Verifying the signature in the above scheme is just linear algebra. There is no repeated doubling needed for elliptic curves. But on the flip side, the parameters are large. The public key and private key for the signature scheme is of size quadratic in the security parameter.
3. Also it provides additional functionalities. For example, with lattice-based cryptography, we can do fully homomorphic encryption.