Edoardo Persichetti Lecture 10: Lattices 29/04/2014

*Notes written by Karol Żebrowski*

# Good vs bad lattice basis

Let's imagine that we are given a lattice $L$ and its basis $B = \{\underline{b}_1, ..., \underline{b}_n\} \subset \mathbb{R}^n$, and let's assume that this basis is orthogonal. Then, any vector $\underline{v}$ of the lattice can be written as $\underline{v} = a_1\underline{b}_1 + ... + a_n\underline{b}_n$ and his norm is given by

$$||\underline{v}||^2 = {a_1}^2||\underline{b}_1||^2 + ... + {a_n}^2||\underline{b}_n||^2$$

Hence, the obvious solutions for the Shortest Vector Problem(SVP) are the shortest among the vectors $\{\pm\underline{b}_i\}$.

Similarly, assuming the orthogonality of the basis, the Closest Vector Problem (CVP) becomes trivial. Indeed, let's take any $\underline{w} \in \mathbb{R}^n$. Then, it can be expressed in the form $\underline{w} = t_1\underline{b}_1 + ... + t_n\underline{b}_n$, for some coefficients $t_i \in \mathbb{R}$. Any vector $\underline{v}$ in the lattice $L$ can be expressed as $\underline{v} = a_1\underline{b}_1 + ... + a_n\underline{b}_n$ for some $a_i \in \mathbb{Z}$, and the distance between $\underline{v}$ and $\underline{w}$ is given by

$$||\underline{w} - \underline{v}||^2 = (a_1 - t_1)^2||\underline{b}_1||^2 + ... + (a_n - t_n)^2||\underline{b}_n||^2$$

so the obvious solution for CVP is $\underline{v} = \lfloor t_1 \rceil\underline{b}_1 + ... + \lfloor t_n \rceil\underline{b}_n$, where $\lfloor r \rceil$ is the integer closest to $r$.

We can say that orthogonal bases are "good", because knowing such a basis for some lattice makes SVP and CVP easy. As a measure of the"goodness" of the basis $B = \{\underline{b}_1, ..., \underline{b}_n\}$ for given lattice $L$ we can use the Hadamard ratio

$$H(B) = \left(\frac{\det(L)}{\|\underline{b}_1\| \cdots \|\underline{b}_n\|}\right)^{1/n}$$

The bigger the ratio, the better the basis. Notice that for orthogonal bases this ratio equals 1.

# Babai's Rounding Algorithm

Assume we have some basis $B$ for a lattice $L$. Let $P$ be the parallelepiped spanned by this basis. Then this parallelepiped shifted by all the vectors $\underline{v} \in L$ covers the space $\mathbb{R}^n$. So, for any $\underline{w} \in \mathbb{R}^n$ there exist such parallelepiped $P' \ni \underline{w}$. Babai's idea for solving CVP is to take the closest vertex of that parallelepiped. This is equivalent to rounding the coefficients of $\underline{w}$ to the closest integers, namely if

$$\underline{w} = t_1\underline{b}_1 + ... + t_n\underline{b}_n$$

then we choose the solution $\underline{v}$ defined as

$$\underline{w} = \lfloor t_1 \rceil\underline{b}_1 + ... + \lfloor t_n \rceil\underline{b}_n$$

**Example 1** Let $L$ be a lattice given by basis $\{\underline{b}_1, \underline{b}_2\}$, where $\underline{b}_1 = (1, 0)$ and $\underline{b}_2 = (0, 1)$, and $w = (-0.4, 0.4)$. Note that this basis is clearly orthogonal. Then our solution for CVP is

$$\underline{v} = \lfloor -0.4 \rceil\underline{b}_1 + \lfloor 0.4 \rceil\underline{b}_2 = 0\underline{b}_1 + 0\underline{b}_2 = (0, 0)$$

that is clearly the closest vector. If we choose another, "less" orthogonal basis of $L$, for example $\underline{b}_1 = (3, 2)$ and $\underline{b}_2 = (2, 1)$ our result is

$$\underline{v} = \underline{1.2b}_1 + \underline{-2b}_2 = 1\underline{b}_1 - 2\underline{b}_2 = (-1, 0)$$

which is not as good as the previous one.

# The GGH Cryptosystem

Introduced by Goldreich, Goldwasser and Halevi in 1997.

- **Key Generation** Let $B = \{\underline{b}_1, ..., \underline{b}_n\}$ be a good basis of lattice $L$ and let $U$ be a unimodular matrix such that $B' = UB$ is a bad basis. Then we define
  - pk: $B'$
  - sk: $(B, U)$
- **Encryption** For a message $\underline{m} \in \mathbb{Z}^n$ we take a small random vector $\underline{r} \in \mathbb{Z}^n$ (here small means $|r_i| \leq d$ for some constant $d$). As a ciphertext we output $\underline{c} = \underline{m}B' + \underline{r}$
- **Decryption** For a ciphertext $\underline{c}$ we use Babai's rounding technique and the good basis $B$ to find the closest vector in the lattice, which is exactly $mB'$. Given that we can compute $m = mB'(B')^{-1}$.

**Remark 1** In practice, to produce a random $\underline{r}$ we use some hash function on the message $\underline{m}$.

**Remark 2** In general, GGH has time complexity $O(n^3 \log n)$, but when we use $B' = HNF(B)$ (Hermite Normal Form) it speeds up to $O(n^2 \log n)$.

# The LLL Lattice Basis Reduction Algorithm

Due to Lenstra, Lenstra and Lovasz, it is used to convert a bad basis into a good basis.

- **Basic idea** Suppose we have two vectors $\underline{v}_1$ and $\underline{v}_2$ for which $|\underline{v}_1| < |\underline{v}_2|$. Then, in order to obtain a pair of vectors which span the same subspace, but which are additionally orthogonal we could use Gram-Smidt idea and take $\underline{v}_1$ and instead of $\underline{v}_2$

  $$\underline{v}_2^* = \underline{v}_2 - \frac{\underline{v}_1 \cdot \underline{v}_2}{||\underline{v}_2||}\underline{v}_1$$

  But usually the fraction $\frac{\underline{v}_1 \cdot \underline{v}_2}{||\underline{v}_2||}$ is not an integer, so instead we take

  $$\underline{v}_2^* = \underline{v}_2 - \left\lfloor \frac{\underline{v}_1 \cdot \underline{v}_2}{||\underline{v}_2||} \right\rceil \underline{v}_1$$

- **Main algorithm** For a given lattice basis $B = \{\underline{b}_1, ..., \underline{b}_n\}$ we make use of the above idea as follows:
  - Until there is pair of vectors $|\underline{b}_i| < |\underline{b}_j|$ in $B$, such that $\underline{b}_j^*$ obtained from this procedure is actually shorter than $\underline{b}_j$ i.e. $|\underline{b}_j^*| < |\underline{b}_j|$, exchange $\underline{b}_j$ for $\underline{b}_j^*$.
  - It can be shown that the algorithm will eventually stop giving a good basis $B^* = \{\underline{b}_1^*, ..., \underline{b}_n^*\}$ , which is "close" to orthogonal

# The NTRU Cryptosystem

**Definition 1** $T(d_1, d_2)$ is the set of polynomials, which have $d_1$ coefficients equal to 1, $d_2$ to $-1$ and the rest to 0.

**Definition 2** Let's define the ring

$$R = \frac{\mathbb{Z}[x]}{x^N - 1}$$

and for integers $p$ and $q$ two other rings by taking $R \mod p$ and $R \mod q$

$$R_p = \frac{\mathbb{Z}_p[x]}{x^N - 1}, R_q = \frac{\mathbb{Z}_q[x]}{x^N - 1}$$

**Definition 3** For a given polynomial $p(x)$ and integer $q$ we will say that $p'(x)$ is its Center Lift $p'(x) = CL(p(x)$ if

- $p(x) \equiv p'(x) \mod q$
- for each coefficient $p'_i$ of $p'(x)$ holds $|p'_i| < \frac{q}{2}$

**NTRU cryptosystem:**

- **Parameters** $N$, $p$, $q$, $d$, such that
  - $N$ is prime
  - $gcd(N, q) = gcd(p, q) = 1$
  - $q > (6d + 1)p$
- **Key Generation** We choose random $f(x) \in T(d + 1, d)$ and $g(x) \in T(d, d)$ such that $f(x)$ is invertible in $R_p$ and $R_q$. Next compute

$$f_p(x) = f^{-1}(x) \mod p \in R_p$$
$$f_q(x) = f^{-1}(x) \mod q \in R_q$$

  Output
  - pk: $h(x) = f_q(x)g(x) \mod q \in R_q$
  - sk: $f(x)$
- **Encryption** As a message we take a polynomial $m(x)$ with small coefficients, meaning $|m_i| < \frac{p}{2}$. Choose a random $r(x) \in T(d, d)$ and output the ciphertext polynomial $c(x)$

$$c(x) = ph(x)r(x) + m(x) \mod q$$

- **Decryption** For a given ciphertext $c(x)$ we follow these steps:
  - compute $a(x) \equiv f(x)c(x) \mod q$
  - Center Lift $a'(x) = CL(a(x)) \mod q$
  - compute $b(x) \equiv f_p(x)a'(x) \mod p$
  - output Center Lift $CL(b(x)) \mod p$

- **Correctness proof:** By the definitions we have

$$a'(x) \equiv a(x) \equiv f(x)c(x) \equiv f(x)(ph(x)r(x) + m(x)) \equiv pf(x)f_q(x)g(x)r(x) + f(x)m(x) \equiv$$

$$\equiv pg(x)r(x) + f(x)m(x) \bmod q$$

Now let's observe that because $g(x), r(x) \in T(d, d)$ the norm of the coefficients of $pg(x)r(x)$ is $\leq 2dp$ , and similarly because $f(x) \in T(d+1, d)$ and $m(x)$ is Center Lifted $\bmod p$ the norm of the coefficeints of $f(x)m(x)$ is $\leq (2d+1)\frac{p}{2}$. Summing up, the norm of the coefficients of $pg(x)r(x) + f(x)m(x)$ is $\leq 2dp + (2d+1)\frac{p}{2} = \frac{(6d+1)p}{2} < \frac{q}{2}$. So polynomials $a'(x)$ and $pg(x)r(x) + f(x)m(x)$ are Center Lifted $\bmod q$ and equivalent $\bmod q$, and hence they are equal in $R$. So in particular

$$a'(x) \equiv pg(x)r(x) + f(x)m(x) \equiv f(x)m(x) \bmod p$$

$$b(x) \equiv f_p(x)a'(x) \equiv f_p(x)f(x)m(x) \equiv m(x) \bmod p$$

Hence, by Center Lifting $b(x)$ we get $m(x)$.

- **NTRU facts:**
  - faster than RSA - just $\frac{2}{3}N^2$ additions/subtractions
  - no security reduction
- **Brute force attack** For publicly known $h(x)$ we try to find a polynomial $f(x) \in T(d+1, d)$ such that

$$f(x)h(x) \equiv g(x) \bmod q$$

for some $g(x) \in T(d, d)$. So it takes $\frac{|T(d+1,d)|}{N}$ tries, because if $h(x)$ works, than also its cyclic shifts $x^k f(x)$ work for $k = 1, ..., N$.

# Lattice based attack on NTRU

Given NTRU Public Key $h(x)$ if we want to break the cryptosystem we need to find $f(x) \in T(d+1, d)$, $g(x) \in T(d, d)$ and a polynomial $u(x)$ satisfying

$$f(x)h(x) = g(x) + qu(x)$$

or equivalently

$$g(x) = f(x)h(x) - qu(x)$$

Consider the following matrix $M$ based on NTRU Public Key $h(x)$ with coefficients $h_0, h_1, ..., h_{n-1}$

$$M = \left[\begin{array}{c|cccc} & h_0 & h_1 & ... & h_{n-1} \\ & h_{n-1} & h_0 & ... & h_{n-2} \\ I & ... & ... & ... & ... \\ & h_1 & h_2 & ... & h_0 \\ \hline 0 & \multicolumn{4}{c}{qI} \end{array}\right]$$

where 0, $I$, $qI$ are respectively the zero matrix, the identity matrix and $q$ times the identity matrix (all having dimension $N \times N$). Then we can notice that if we take $f$, $g$ and $u$ as vectors of length $N$, they satisfy

$$(f, g) = (f, -u) \cdot M$$

Now we will consider the rows of $M$ as a basis for full-rank lattice $L_{NTRU}$ of dimension $2N$. We can observe that

1. $\det L_{NTRU} = q^N$
2. $||(f,g)|| \approx \sqrt{4d} \approx \sqrt{4\frac{N}{3}} \approx 1,155\sqrt{N}$
3. following the Gaussian Heuristic the expected length of the shortest vector in $L_{NTRU}$ is

$$\sigma(L_{NTRU}) = \sqrt{\frac{2N}{2\pi e}} \det(L_{NTRU})^{1/2N} = \sqrt{\frac{N}{\pi e}}\sqrt{q} \approx \sqrt{\frac{2}{\pi e}}N \approx 0,484N$$

   (since we assume $q \approx 2N$).

So likely $(f,g)$ is the shortest vector in $L_{NTRU}$ and by solving $\mathcal{O}(N^\epsilon)$-Approximate SVP where $\epsilon < \frac{1}{2}$(or just SVP) we will find $(f,g)$. It is because $\frac{\sigma(L_{NTRU})}{||(f,g)||} \asymp \mathcal{O}(N^{\frac{1}{2}})$.