Computational Number Theory and Public Key Cryptography (INF 245), fall 2020

Igor Semaev

August 4, 2022

1 Chapter 8. NTRU and lattice based cryptography

1.1 NTRU

In this section, we describe the NTRU public encryption. The system was introduced at the Crypto'96 conference and published in the proceedings of the Third International Symposium "Algorithmic Number Theory" by Hoffstein, Pipher, Silverman in 1998. The security of the NTRU is based on the hardness of computing short vectors in some special lattices. By some of its characteristics, NTRU is superior to RSA. The complexity of calculating a secret key from an public one or the complexity of recovering the plaintext from the cipher-text, seems depend exponentially on the length N of the block of plain-text to encrypt. The complexity of encryption (decryption) is $O(N^2)$ bit operations. In the RSA, the complexity of encryption (decryption) is estimated by $O(N^3)$ bit operations. RSA security is based on the complexity of the factorization problem, which depends subexponentially on the size of the number to factor, which is the same as the size of the encrypted block of the plain-text. NTRU key generation is a very simple operation. In RSA the key generation involves selecting a pair of primes with special properties to make it difficult to factor their product. However, in RSA, the length of the cipher-text is equal to the length of the plain-text. In NTRU, the length of the cipher-text is much longer. For example, it is 4.4 times more for a system whose security is comparable to that of RSA with a 1024-bit modulus.

1.1.1 Notation

Let N and p be primes, where p is odd and commonly $p = 3$, and q be an even natural number. The computation is inside the ring $R = \mathbb{Z}[x]/(x^N - 1)$ of polynomials with integer coefficients modulo the polynomial $x^N - 1$. Every $F \in R$ may be written as a polynomial of degree $< N$ or an integer vector of length N

$$F = \sum_{i=0}^{N-1} a_i x^i, \quad F = (a_0, a_1, \ldots, a_{N-1}).$$

Let $*$ denote the multiplication in R. Then $F * G = H$, where

$$G = \sum_{i=0}^{N-1} b_i x^i, \quad H = \sum_{i=0}^{N-1} c_i x^i$$

and $c_k = \sum_{i+j \equiv k \bmod N} a_i b_j$. We write $F \equiv G \bmod q$ is the coefficients of F and G are congruent modulo q.

Let L_m be a subset of R whose coefficients are in $[-(p-1)/2, \ldots, (p-1)/2]$. The message is represented by a sequence of blocks encoded by the elements $\in L_m$. Before encryption the block m is to be padded by random bits with OAEP for instance.

Let $L(d_1, d_2)$ denote a set of $f \in R$, where f has d_1 coefficients 1, d_2 coefficients -1 and the rest are 0. One chooses three natural numbers d_f, d_g, d and sets

$$L_f = L(d_f, d_f - 1), \quad L_g = L(d_g, d_g), \quad L_\phi = L(d, d).$$

The polynomials $f \in L(d, d)$ are not invertible in R as $f(1) = 0$ and therefore $\gcd(f(x), x^N - 1) \neq 1$. For a proper decryption in NTRU one needs the polynomials $\in L_f$ to be invertible in R with high probability. That is why it is set $L_f = L(d_f, d_f - 1)$.

1.1.2 NTRU Public and Private Keys

The numbers N, p, q are public. NTRU private key is two secret polynomials $f \in L_f$ and $g \in L_g$. The polynomial f is to be invertible in R modulo q and modulo p. Let F_q and F_p denote the inverse of f modulo q and modulo p, that is

$$f_q * f \equiv 1 \bmod q, \quad f_p * f \equiv 1 \bmod p.$$

The inversions are computed with the Extended Euclidean Algorithm applied to $f, X^N - 1$ modulo q and p. That is all the intermediate polynomials arising in the computation are reduced modulo p or q. Alternatively, one applies the Extended Euclidean Algorithm to $f, X^N - 1$ and finds the polynomials u, v with rational coefficients such that $uf + v(x^N - 1) = 1$. Then $fq \equiv u \pmod q$ and $fp \equiv u \pmod p$. NTRU public key is the polynomial

$$h \equiv p \cdot f^{-1} * g \equiv p \cdot f_q * g \bmod q$$

1.1.3 NTRU Encryption and Decryption

In order to encrypt $m \in L_m$(a string of coefficients of the polynomial m with entries in {1,0,-1} for $p = 3$) one randomly chooses a secret polynomial $\phi \in L_\phi$ and computes the polynomial

$$e \equiv \phi * h + m \bmod q.$$

whose string of coefficients with entries in $\{0, 1, \ldots, q-1\}$ is a cipher-text for m. The length of the cipher-text is about $\ln q / \ln p$ times larger than that of the plain-text.

In order to decrypt one computes $a \equiv f * e \bmod q$. The coefficients $a_0, \ldots, a_{N-1}$ of a are residues modulo q. One takes them in the interval $-q/2 < a_i \leq q/2$. Then one computes

$$m \equiv f^{-1} * a \equiv f_p * a \bmod p. \tag{1}$$

1.1.4 Proof of NTRU Decryption

We will define the conditions for (1). Let $F = \sum_{i=0}^{N-1} a_i x^i$. We denote $|F|_\infty = \max_{0 \leq i < N} |a_i|$ and $|F| = \sqrt{\sum_{0 \leq i < N} a_i^2}$ is the Euclidean norm of F. The analysis below is based on the following empirical fact

$$|F * G|_\infty \leq \gamma \cdot |F| \cdot |G| \tag{2}$$

with a high probability for a constant γ. For instance, $\gamma = 0.15$ for $N \leq 1000$. Then

$$a \equiv f * e \equiv f * \phi * h + f * m \equiv (p \cdot \phi) * g + f * m \bmod q$$

as $f * h \equiv g \bmod q$. Then

$$|p \cdot \phi * g + f * m|_\infty \leq p|\phi * g|_\infty + |f * m|_\infty \leq \gamma(p|\phi| \cdot |g| + |f| \cdot |m|).$$

That is because $|\phi * g|_\infty \leq \gamma|\phi| \cdot |g|$ and $|f * m|_\infty \leq \gamma|f| \cdot |m|$ by (2). Since, $|m| \leq \frac{p-1}{2}\sqrt{N}$, we have

$$|(p \cdot \phi) * g + f * m|_\infty \leq \gamma(p\sqrt{4dd_g} + \frac{p-1}{2}\sqrt{2d_f N}). \tag{3}$$

The parameters N, p, q, d_f, d_g, d are chosen such that the right hand side in (3) is $< q/2$, then

$$|(p \cdot \phi) * g + f * m|_\infty < q/2.$$

For instance, for $(N, p, q) = (107, 3, 64)$ one may take $d_f = 15, d_g = 12$ and $d = 5$. With high probability

$$|(p \cdot \phi) * g + f * m|_\infty < 24 < q/2 = 32.$$

Then $a = (p \cdot \phi) * g + f * m$ and the decryption really works.

1.1.5 Example

The example was taken from the Wikipedia page NTRUEncrypt. Let $(N, p, q) = (11, 3, 32)$ and

$$f = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}, \quad g = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}.$$

Then

$$\begin{aligned} f_p &= 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9 \\ f_q &= 5 + 9X + 6X^2 + 16X^3 + 4X^4 + 15X^5 + 16X^6 + 22X^7 + 20X^8 + 18X^9 + 30X^{10} \end{aligned}$$

are inverses of f modulo 3 and 32. The public key $h = pF_q * g \pmod{32}$ is then

$$8 - 7X - 10X^2 - 12X^3 + 12X^4 - 8X^5 + 15X^6 - 13X^7 + 12X^8 - 13X^9 + 16X^{10}.$$

Let $m = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}$ be a plain-text block. One randomly chooses $\phi = -1 + X^2 + X^3 + X^4 - X^5 - X^7$ and computes the cipher-text $e = \phi * h + m \pmod{32}$ as

$$14 + 11X + 26X^2 + 24X^3 + 14X^4 + 16X^5 + 30X^6 + 7X^7 + 25X^8 + 6X^9 + 19X^{10}.$$

To decrypt one computes $a = f * e \pmod{32}$ as

$$3 - 7X - 10X^2 - 11X^3 + 10X^4 + 7X^5 + 6X^6 + 7X^7 + 5X^8 - 3X^9 - 7X^{10}$$

and takes the coefficients in the interval $(-16, 16]$. Then the decryption is

$$f_p * a \pmod{3} = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}.$$

That is the correct plain-text.

1.1.6 NTRU Basic Analysis

One can search over $g_1 \in L_g$, compute f_1 from $f_1 * h \equiv g_1 \pmod{q}$ and check if $f_1 \in L_f$. Remark that since h is not invertible in R, computing f_1 requires some care, see the next Section, where a similar problem is solved. If $g_1 \in L_g, f_1 \in L_f$ and $f_1 * h \equiv g_1 \pmod{q}$ then the pair f_1, g_1 may be used for decrypting the cipher-text as it is shown below.

Moreover, let $|f_1| \leq \sqrt{2d_f}$ and $|g_1| \leq \sqrt{2d_g}$ and $g_1 \equiv f_1 * h \pmod{q}$. Then firstly the pair f_1, g_1 generates the same key h. Secondly, one can recover the plain-text as

$$|(p \cdot \phi) * g_1 + f_1 * m|_\infty < q/2.$$

as for the correct pair f, g. Therefore $a_1 = (p \cdot \phi) * g_1 + f_1 * m$ and $m = f_1^{-1} * a_1 \pmod{p}$.

Let $h = \sum_{i=0}^{N-1} h_i x^i$ and α be a parameter and let L_h be a lattice generated by the rows of the following matrix

$$H = \begin{pmatrix} \alpha & 0 & \dots & 0 & h_0 & h_1 & \dots & h_{N-1} \\ 0 & \alpha & \dots & 0 & h_{N-1} & h_0 & \dots & h_{N-2} \\ \dots & & & & & & & \\ 0 & 0 & \dots & \alpha & h_1 & h_2 & \dots & h_0 \\ 0 & 0 & \dots & 0 & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & q & \dots & 0 \\ \dots & & & & & & & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & q \end{pmatrix}.$$

Since $f * h \equiv g \pmod q$ the lattice contains the vector

$$\tau = (\alpha f, g) = (\alpha f_0, \alpha f_1, \ldots, \alpha f_{N-1}, g_0, g_1, \ldots, g_{N-1}).$$

Really, one has $f * h \equiv g + qr$, where $r = \sum_{i=0}^{N-1} r_i x^i$. Therefore,

$$(f, -r)H = (\alpha f, g).$$

In other words, $(\alpha f, g)$ is a linear combination of the rows of H with integer coefficients $(f, -r)$. Then

$$|\tau|^2 = \alpha^2 |f|^2 + |g|^2.$$

Since $|f| \approx \sqrt{2d_f}$ and $|g| = \sqrt{2d_g}$ one takes $\alpha = |g|/|f| \approx \sqrt{d_g/d_f}$. One may try to find a shortest (relatively short with norm $\leq |\tau|$) non-zero vector $(\alpha f_1, g_1) \in L_h$. The pair f_1, g_1 may likely be then used for decrypting cipher texts. A natural way to find a short vector in the lattice is to apply a lattice reduction algorithm. However modern reduction algorithms as LLL are not able to find very short vectors in general lattices in reasonable time. That is a hard problem.

1.1.7 Multiple encryption of the same message block

Let a message block $m \in L_m$ be encrypted with different randomly chosen $\phi_1, \ldots, \phi_s$ and $e_1, \ldots, e_s$ be relevant cipher-texts. We show how to recover m given the cipher-texts. By the definition

$$e_i = \phi_i * h + m, \quad i = 1, \ldots, s.$$

So for any $i \neq j$ we get

$$e_i - e_j \equiv \phi_i * h - \phi_j * h \equiv (\phi_i - \phi_j) * h \pmod q.$$

We have to recover $\phi_{ij} = \phi_i - \phi_j$ from this equation. However as h is not invertible in R that is not straightforward. Really, $h \equiv p \cdot f_q * g$ and $g \in L(d_g, d_g)$. So $g(1) = h(1) = 0$ and h is not invertible modulo $X^N - 1$, that is in R, as h and $X^N - 1$ have a common factor $X - 1$.

Nevertheless, one may assume $\gcd(h, X^N - 1) = X - 1$ and compute the polynomials u, v such that

$$u \cdot h + v \cdot (X^N - 1) \equiv X - 1 \pmod q. \tag{4}$$

Then

$$c_{ij} = (e_i - e_j) * u \equiv (\phi_i - \phi_j) * h * u \equiv \phi_{ij} * (X - 1) \pmod q.$$

The polynomial c_{ij} is of degree $\leq N - 1$ as this computation is in R. One takes the coefficients of c in $(q/2, q/2]$. As the coefficients of $\phi_{ij} * (X - 1)$ are small,

$$c_{ij} = \phi_{ij} * (X - 1)$$

in R or $c_{ij} = \phi_{ij} \cdot (X-1) \mod X^N - 1$ in $\mathbb{Z}[X]$. Therefore,

$$c_{ij} = \phi_{ij} \cdot (X-1) - t_{ij}(X^N - 1),$$

where $-2 \leq t_{ij} \leq 2$ is the leading coefficient in ϕ_{ij} as the degree of $\phi_{ij} \cdot (X-1)$ is $\leq N$. To conclude,

$$\phi_{ij} = c_{ij}/(X-1) + t_{ij}(X^N - 1)/(X-1) = c_{ij}/(X-1) + t_{ij}(X^{N-1} + X^{N-2} + \ldots + 1).$$

As t_{ij} is small, it is found by trials. We summarise the algorithm.

1. Compute the polynomial $u \in R$ modulo q from (4).
2. For every $i \neq j$, compute the polynomial $c_{ij} \equiv (e_i - e_j) * u \pmod q$ of degree $< N$ and set its coefficients in $(q/2, q/2]$.
3. Find $-2 \leq t_{ij} \leq 2$ such that $c = c_{ij}/(X-1) + t_{ij}(X^{N-1} + X^{N-2} + \ldots + 1)$ has the coefficients ≤ 2 in absolute values (that may not be unique). Set $\phi_i - \phi_j = c$.
4. Find the coefficients of

$$\phi_i = \sum_{k=0}^{N-1} a_k X^k, \quad \phi_j = \sum_{k=0}^{N-1} b_k X^k.$$

   from the coefficients of $\phi_i - \phi_j$ by the following rule. If $a_k - b_k = 2$, then $a_k = 1, b_k = -1$. If $a_k - b_k = -1$, then $a_k = -1, b_k = 0$ or $a_k = 0, b_k = 1$. If $a_k - b_k = 0$, then $a_k = b_k$. If $a_k - b_k = 1$, then $a_k = 1, b_k = 0$ or $a_k = 0, b_k = -1$. If $a_k - b_k = -2$, then $a_k = -1, b_k = 1$.
5. Also use that $\phi_i \in L(d, d)$ to recover ϕ_i. Compute the plain-text $m \in L_m$ from $e_i \equiv \phi_i * h + m \pmod q$.

1.1.8 NTRU Recommended Parameters

Moderate security parameters according to the initial publication are $(N, p, q) = (107, 3, 64)$ and

$$L_f = L(15, 14), \quad L_g = L(12, 12), \quad L_\phi = L(5, 5)$$

Private key size is 340 bits, public key size 624 bits. Best cryptanalysis $2^{26.5}$ operations in 1998, improved later. Those parameters are not recommended now.

High security parameters according to the initial publication are $(N, p, q) = (167, 3, 128)$ and

$$L_f = L(61, 60), \quad L_g = L(20, 20), \quad L_\phi = L(18, 18)$$

Private key size is 530 bits, public key size 1169 bits. Best cryptanalysis $2^{77.5}$ operations in 1998, improved later.

Highest security parameters according to the initial publication are $(N, p, q) = (503, 3, 256)$ and

$$L_f = L(216, 215), \quad L_g = L(72, 72), \quad L_\phi = L(55, 55)$$

Private key size is 1595 bits, public key size 4024 bits. Best cryptanalysis 2^{170} operations in 1998, improved later.

1.2 Lattices

Let $\mathbb{R}^n$ denote a real vector space of dimension n and let $a = (\alpha_1, \ldots, \alpha_n), b = (\beta_1, \ldots, \beta_n)$ be vectors in $\mathbb{R}^n$. One then defines a dot-product $a \cdot b = \sum_{i=1}^n \alpha_i \beta_i$ and the length(norm) of the vector $|a| = \sqrt{a \cdot a}$. Non-zero vectors a, b are called orthogonal if $a \cdot b = 0$. The Cauchy inequality says

$$|a \cdot b| \leq |a| \cdot |b|$$

for non-zero $a, b \in \mathbb{R}^n$. The equality holds if and only if a, b are linearly dependent. Let $b_1, b_2, \ldots, b_k \in \mathbb{R}^n$ be linearly independent. The set of vectors in $\mathbb{R}^n$ generated by $b_1, b_2, \ldots, b_k$ with integer coefficients is a lattice L of rank k and the set of generators is called a basis. Formally,

$$L = \{ \sum_{i=1}^{k} z_i b_i | z_i \in \mathbb{Z} \}.$$

Lemma 1. *For any $r \geq 0$ the inequality $|\sum_{i=1}^k z_i b_i| \leq r$ has a finite number of solutions in integer $z_1, \ldots, z_n$.*

Proof. It is enough to prove the statement for $k = n$. Let $y = \sum_{i=1}^k z_i b_i$ and $|y| \leq r$. Let B be a matrix with rows $b_1, b_2, \ldots, b_n$ and B_i a matrix produced from B by changing the i-th row by y. By Cramer's rule

$$z_i = \frac{\det B_i}{\det B},$$

where by Hadamard inequality,

$$|\det B_i| \leq \frac{|b_1| \cdot |b_2| \cdot \ldots \cdot |b_n|}{|b_i|} |y|.$$

So

$$|z_i| \leq \frac{|b_1| \cdot |b_2| \cdot \ldots \cdot |b_n|}{|b_i| \det B} r.$$

Therefore for a fixed r each integer variable z_i may only have a finite number of values. That implies the statement. □

So any lattice contains a shortest (of the smallest norm) non-zero vector. The determinant (volume) of the lattice is the positive number

$$\Lambda(L) = |\det B \cdot B^T|^{1/2}.$$

The value $\Lambda(L)$ does not depend on the basis as an integer transform matrix from one basis to another basis of L has the determinant ± 1.

1.2.1 Reduced bases of a lattice

A basis $b_1, b_2, \ldots, b_k$ of the lattice $L \subseteq \mathbb{R}^n$ is called Minkowski reduced if b_1 is the shortest non-zero vector in the lattice, b_2 is the shortest non-zero vector in the lattice such that b_1, b_2 may be extended to a basis for L, etc. That is for $b_1, b_2, \ldots, b_{j-1}$ the vector b_j is the shortest such that $b_1, b_2, \ldots, b_{j-1}, b_j$ may be extended to a basis of L. One can prove that any lattice admits a Minkowski reduced basis and the number of such bases is finite. Minkowski reduced basis is optimal (smallest regarding the norms of the basis vectors), and in particular it contains a shortest non-zero vector of L.

Lemma 2. *Let $k = 2$ and b_1, b_2 be linearly independent vectors. Then b_1, b_2 is a Minkowski reduced basis if and only if $|b_1| \leq |b_2|$ and $|b_2 + zb_1| \geq |b_2|$ for any integer z.*

Proof. Obviously, if b_1, b_2 is a Minkowski reduced basis, then $|b_1| \leq |b_2|$ and $|b_2 + zb_1| \geq |b_2|$ for any integer z. Let's prove the reverse statement. It is enough to prove that b_1 is the shortest non-zero vector in the lattice generated by b_1, b_2. Really, let $c = ub_1 + vb_2$ for integer u, v be any non-zero vector in the lattice. We prove that $|c| \geq |b_1|$. If $v = 0$, then this holds. Let $v \neq 0$. We divide u by v with remainder and get $u = qv + r$, where $0 \leq r < |v|$. With triangular inequality($|a + c| \geq |a| - |c|$ for any vectors a, c)

$$|c| = |v(qb_1 + b_2) + rb_1| \geq |v| \cdot |qb_1 + b_2| - r|b_1| \geq (|v| - r)|b_1| \geq |b_1|.$$

□

It is easy to prove that the second condition $|b_2 + zb_1| \geq |b_2|$ for any integer z is equivalent to $2|b_1 \cdot b_2| \leq |b_1|^2$.

Lagrange-Gauss reduction algorithm.

Input. Linearly independent $b_1, b_2 \in \mathbb{R}^n$, where $|b_2| \geq |b_1|$. **Output.** A reduced basis of the lattice generated by b_1, b_2.

1. Compute $r = \lfloor \frac{b_1 \cdot b_2}{|b_1|^2} \rceil$, where $\lfloor \alpha \rceil$ denotes a closest integer to α. Put $a \leftarrow b_2 - rb_1$.
2. If $|a| < |b_1|$, then $b_2 \leftarrow b_1, b_1 \leftarrow a$, go to 1, else $b_2 \leftarrow a$ and terminate.

Example. Let $b_1 = (10, 13, -16, 3)$ and $b_2 = (13, 17, -21, 4)$. One computes

$$r = \lfloor \frac{10 \cdot 13 + 13 \cdot 17 + (-16) \cdot (-21) + 3 \cdot 4}{10^2 + 13^2 + (-16)^2 + 3^2} \rceil = 1$$

and $a = b_2 - b_1 = (3, 4, -5, 1)$. The basis is $b_1 = (3, 4, -5, 1)$ and $b_2 = (10, 13, -16, 3)$. One computes

$$r = \lfloor \frac{3 \cdot 10 + 4 \cdot 13 + (-16) \cdot (-5) + 1 \cdot 3}{3^2 + 4^2 + (-5)^2 + 1^2} \rceil = 3$$

and $a = b_2 - 3b_1 = (1, 1, -1, 0)$. The basis is $b_1 = (1, 1, -1, 0)$ and $b_2 = (3, 4, -5, 1)$. One computes

$$r = \lfloor \frac{1 \cdot 3 + 1 \cdot 4 + (-1) \cdot (-5) + 0 \cdot 1}{1^2 + 1^2 + (-1)^2 + 0^2} \rceil = 4$$

and $a = b_2 - 4b_1 = (-1, 0, -1, 1)$. The basis is $b_1 = (1, 1, -1, 0)$ and $b_2 = (-1, 0, -1, 1)$, terminate.

For lattices of large rank k an efficient algorithm for constructing a Minkowski reduced basis is unknown. Generally, the basis $b_1, \ldots, b_k$ of a lattice L is called reduced if

$$|b_1|\,|b_2| \ldots |b_k| \leq c_k\,\Lambda(L).$$

The definition depends on the parameter $c_k > 0$. A shortest vector b in the reduced basis $b_1, \ldots, b_k$ satisfies $|b| \leq c_k^{1/k} \Lambda(L)^{1/k}$. There exist reduced bases which are not reduced if c_k is diminished. One can prove that a Minkowski reduced basis is reduced according the above definition for some c_k. LLL algorithm efficiently finds a so called LLL-reduced basis $b_1, \ldots, b_k$, where

$$|b_1|\,|b_2| \ldots |b_k| \leq 2^{\frac{k(k-1)}{4}}\,\Lambda(L).$$

Thus the shortest vector b in the basis satisfies

$$|b| \leq 2^{(k-1)/4} \Lambda(L)^{1/k}.$$

However by Hermite's theorem the length(norm) of the shortest nonzero vector in L is bounded by $\sqrt{\gamma_k}\Lambda(L)^{1/k}$, where $\gamma_k \leq \frac{1.7445k}{2\pi e}$ for large k. So the shortest vector in a LLL-reduced basis is generally a very rough approximation to the shortest non-zero vector of the lattice.

1.3 Gram-Schmidt Orthogonalization

Let $b_1, \ldots, b_k$ be linearly independent vectors in $\mathbb{R}^n$. We want to construct vectors $b_1^*, \ldots, b_k^*$ such that they generated the same space as $b_1, \ldots, b_k$ and pairwise orthogonal. We say that two vectors $a, b \in \mathbb{R}^n$ are orthogonal if their dot-product is 0, this fact is denoted $ab = 0$.

We set

$$\begin{aligned}
b_1^* &= b_1, \\
b_2^* &= b_2 - \mu_{21} b_1^*, \quad \mu_{21} = \frac{b_2 b_1^*}{|b_1^*|^2}, \\
b_3^* &= b_3 - \mu_{32} b_2^* - \mu_{31} b_1^*, \quad \mu_{3j} = \frac{b_i b_j^*}{|b_j^*|^2}, \quad 1 \le j \le 2, \\
&\dots \\
b_k^* &= b_k - \mu_{kk-1} b_{k-1}^* - \ldots - \mu_{k1} b_1^*, \quad \mu_{kj} = \frac{b_k b_j^*}{|b_j^*|^2}, \quad 1 \le j \le k-1.
\end{aligned}$$

By induction, it is easy to see that $b_i^* b_j^* = 0, i \neq j$, so the basis is orthogonal. On can write $b_i = b_i^* + \mu_{ii-1} b_{i-1}^* + \ldots + \mu_{i1} b_1^*$ and therefore

$$|b_i|^2 = |b_i^*|^2 + \mu_{ii-1}^2 |b_{i-1}^*|^2 + \ldots \mu_{i1}^2 |b_1^*|^2 \ge |b_i^*|^2.$$

So $|b_i| \ge |b_i^*|$. Let's denote

$$B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{pmatrix}, \quad B^* = \begin{pmatrix} b_1^* \\ b_2^* \\ \dots \\ b_k^* \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ \mu_{21} & 1 & \dots & 0 & 0 \\ \dots & & & & \\ \mu_{k1} & \mu_{k2} & \dots & \mu_{kk-1} & 1 \end{pmatrix}.$$

That is B is a matrix of size $k \times n$ whose rows are $b_1, \ldots, b_k$. The matrix B^* is a matrix of size $k \times n$ whose rows are $b_1^*, \ldots, b_k^*$ and M is a lower-diagonal matrix of size $k \times k$ whose entries under the main diagonal are Gram-Schmidt coefficients $\mu_{ij}, 1 \le j < i \le k$. We have $B = MB^*$.

Let the lattice L be generated by the rows of B and let $\Lambda(L)$ be its volume. The inequality $\Lambda(L) \le |b_1| \dot{|} b_2| \ldots |b_k|$, first proved by Hadamard, follows from

$$\begin{aligned}
\Lambda(L) &= |\det(BB^T)|^{1/2} \\
&= |\det(MB^* B^{*T} M^T)|^{1/2} = |\det(B^* B^{*T})|^{1/2} \\
&= |b_1^*| \dot{|} b_2^*| \ldots |b_k^*| \le |b_1| \dot{|} b_2| \ldots |b_k|
\end{aligned}$$

as $B^* B^{*T}$ is a diagonal matrix with $|b_1^*|^2, \ldots, |b_k^*|^2$ on the main diagonal.

1.4 LLL Reduced Basis

The basis $b_1, \ldots, b_k$ is called LLL-reduced if the following two conditions are fulfilled.

Size reduced. $|\mu_{ij}| \le 1/2$ for $1 \le j < i \le k$

Lovász condition. $|b_i^* + \mu_{ii-1} b_{i-1}^*|^2 \ge \frac{3}{4} |b_{i-1}^*|^2$, for $2 \le i \le k$.

Theorem 1. *Let $b_1, \ldots, b_k$ be an LLL-reduced basis of a lattice L. Then*

1. $|b_j|^2 \leq 2^{i-1}|b_i^*|^2$ *for* $1 \leq j < i \leq k$
2. $\Lambda(L) \leq \prod_{i=1}^{k} |b_i| \leq 2^{\frac{k(k-1)}{4}} \Lambda(L)$
3. *Suppose b is a shortest vector in the basis $b_1, \ldots, b_k$, then* $|b| \leq 2^{\frac{k-1}{4}} \Lambda(L)^{1/4}$.

Proof. Let's prove the first statement. From Lovász condition

$$|b_i^*|^2 \geq (\frac{3}{4} - \mu_{ii-1}^2)\,|b_{i-1}^*|^2 \geq (\frac{3}{4} - \frac{1}{4})\,|b_{i-1}^*|^2 \geq \frac{1}{2}\,|b_{i-1}^*|^2$$

as $|\mu_{ii-1}| \leq 1/2$. So

$$|b_i^*|^2 \geq 2^{-1}\,|b_{i-1}^*|^2 \geq \ldots 2^{j-i}\,|b_j^*|^2$$

for $1 \leq j \leq i$ and

$$\begin{aligned} |b_i|^2 &= |b_i^*|^2 + \sum_{j=1}^{i-1} \mu_{ij}^2\,|b_j^*|^2 \leq |b_i^*|^2 + \sum_{j=1}^{i-1} \frac{1}{4} 2^{i-j}\,|b_i^*|^2 \\ &= |b_i^*|^2(1 + \frac{1}{4}\sum_{j=1}^{i-1} 2^{i-j}) = |b_i^*|^2(1 + \frac{1}{4}\,(2^i - 2)) \leq 2^{i-1}|b_i^*|^2. \end{aligned}$$

Therefore

$$|b_j^*|^2 \leq 2^{j-1}\,|b_j^*|^2 \leq 2^{j-1}2^{i-j}\,|b_i^*|^2 = 2^{i-1}\,|b_i^*|^2.$$

That holds for every $1 \leq j \leq i$. The first statement is thus proved. In order to prove the second statement one uses Hadamard's inequality and the first statement

$$\Lambda(L) \leq \prod_{i=1}^{k} |b_i| \leq \prod_{i=1}^{k} 2^{(i-1)/2}\,|b_i^*| = 2^{\sum_{i=1}^{k}(i-1)/2} \prod_{i=1}^{k} |b_i^*| = 2^{k(k-1)/4}\,\Lambda(L).$$

Thus the second statement follows and it implies the last statement of the theorem. □

1.5 LLL Reduction Algorithm

Input to the algorithm is a basis $b_1, \ldots, b_k$ of a lattice L in $\mathbb{R}^n$. Output is an LLL-reduced basis for L.

1. Gram-Schmidt orthogonalization. Compute the orthogonal basis $b_1^*, \ldots, b_k^*$ and Gram-Schmidt coefficients μ_{ij} for $1 \leq j < i \leq k$ according to Section 1.3. Construct the matrices B^* and M.

2. Size-reduction subroutine.

for $i = 2$ to k do
for $j = i - 1$ down to 1 do $\quad b_i \leftarrow b_i - \lfloor \mu_{ij} \rceil \, b_j$
for $s = 1$ to j do $\quad \mu_{is} \leftarrow \mu_{is} - \lfloor \mu_{ij} \rceil \, \mu_{js}$.

This changes the basis B and the Gram-Schmidt coefficients in M while the orthogonal basis B^* remains. Then go to step 3.

3. Satisfy Lovász' condition. Find $2 \le i \le k$ such that the condition is not satisfied, that is

$$|b_i^*|^2 < (\frac{3}{4} - \mu_{ii-1}^2) \, |b_{i-1}^*|^2.$$

Then swap b_i and b_{i-1}. This may change the basis B, the orthogonal basis B^*, and M. Renew B^*, M by using the Gram-Schmidt orthogonalization (there is a more efficient procedure not shown here, where it is not necessary to apply the Gram-Schmidt orthogonalization again). Go to step 2.

If the condition is satisfied for all $2 \le i \le k$, then terminate.

Remark that in an efficient implementation of the algorithm one has to keep the current basis B, the norms of vectors in its orthogonal basis B^* and the Gram-Schmidt coefficients M.

1.6 LLL Reduction Algorithm. Example

Let $n = 4, k = 3$ and let the rows of

$$B = \begin{pmatrix} 2 & 2 & 3 & 1 \\ 7 & 7 & 10 & 3 \\ 11 & 10 & 14 & 4 \end{pmatrix}$$

form the basis of a lattice L. We want to construct an LLL-reduced basis for L. The orthogonal basis and the Gram-Schmidt coefficients for B are in the matrices below.

$$B^* = \begin{pmatrix} 2 & 2 & 3 & 1 \\ 2/9 & 2/9 & -1/6 & -7/18 \\ 3/5 & -2/5 & -1/5 & 1/5 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 61/18 & 1 & 0 \\ 44/9 & 14/5 & 1 \end{pmatrix}.$$

One applies the size reduction first. Since, $\mu_{21} = 61/18$ and $\lfloor \mu_{21} \rceil = 3$, one transforms $b_2 \leftarrow b_2 - 3\, b_1$. This changes B and M to

$$B = \begin{pmatrix} 2 & 2 & 3 & 1 \\ 1 & 1 & 1 & 0 \\ 11 & 10 & 14 & 4 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 7/18 & 1 & 0 \\ 44/9 & 14/5 & 1 \end{pmatrix}.$$

Since, $\mu_{32} = 14/5$ and $\lfloor \mu_{32} \rceil = 3$, one transforms $b_3 \leftarrow b_3 - 3\,b_2$. This changes B and M to

$$B = \begin{pmatrix} 2 & 2 & 3 & 1 \\ 1 & 1 & 1 & 0 \\ 8 & 7 & 11 & 4 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 7/18 & 1 & 0 \\ 44/9 & -1/5 & 1 \end{pmatrix}.$$

Since, $\mu_{31} = 67/18$ and $\lfloor \mu_{31} \rceil = 4$, one transforms $b_3 \leftarrow b_3 - 4\,b_1$. This changes B and M to

$$B = \begin{pmatrix} 2 & 2 & 3 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 7/18 & 1 & 0 \\ -5/18 & -1/5 & 1 \end{pmatrix}.$$

The last three transforms does not change the orthogonal basis B^*. The Lovász' condition is not satisfied for $i = 2$ as $|b_2^*|^2 = 5/18, |b_1^*|^2 = 18, \mu_{21} = 7/18$ and so $|b_2^*|^2 < (3/4 - \mu_{21}^2)\,|b_1^*|^2$. One swaps b_1 and b_2 and recomputes

$$B = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & 2 & 3 & 1 \\ 0 & -1 & -1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1/3 & -1/3 & 2/3 & 1 \\ 3/5 & -2/5 & -1/5 & 1/5 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 7/3 & 1 & 0 \\ -2/3 & -1/5 & 1 \end{pmatrix}.$$

Since $\mu_{21} = 7/3$ and $\lfloor \mu_{21} \rceil = 2$, one transforms $b_2 \leftarrow b_2 - 2b_1$. This changes B and M to

$$B = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & -1 & -1 & 0 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ -2/3 & -1/5 & 1 \end{pmatrix}.$$

Since $\mu_{31} = -2/3$ and $\lfloor \mu_{31} \rceil = -1$, one transforms $b_3 \leftarrow b_3 + b_1$. This changes B and M to

$$B = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 1/3 & -1/5 & 1 \end{pmatrix}.$$

The Lovász' condition is not satisfied for $i = 2$ as $|b_2^*|^2 = 5/3, |b_1^*|^2 = 3, \mu_{21} = 1/3$ and so $|b_2^*|^2 < (3/4 - \mu_{21}^2)\,|b_1^*|^2$. One swaps b_1 and b_2 and recomputes

$$B = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1/2 & -1/2 \\ 3/5 & -2/5 & -1/5 & 1/5 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & 2/5 & 1 \end{pmatrix}.$$

The basis B is size reduced. The Lovász' condition is not satisfied for $i = 3$ as $|b_3^*|^2 = 3/5, |b_2^*|^2 = 2, \mu_{32} = 2/5$ and so $|b_3^*|^2 < (3/4 - \mu_{32}^2)\,|b_2^*|^2$. One swaps b_2 and b_3 and recomputes

$$B = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1/2 & -1/2 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 1 & 1 \end{pmatrix}.$$

Since $\mu_{32} = 1$, one transforms $b_3 \leftarrow b_3 - b_2$. This changes B and M to

$$B = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix}.$$

The basis B is size reduced. The Lovász' condition is not satisfied for $i = 2$ as $|b_2^*|^2 = 1, |b_1^*|^2 = 2, \mu_{21} = 0$ and so $|b_2^*|^2 < (3/4 - \mu_{21}^2)\,|b_1^*|^2$. One swaps b_1 and b_2 and recomputes

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad B^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1/2 & -1/2 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{pmatrix}.$$

The basis B is size reduced and the Lovász' condition is satisfied for $i = 1, 2$. This basis is LLL-reduced.

1.7 Hard Computational Problems in Lattices

Let $B = b_1, \ldots, b_k$ be the basis of a lattice L and $a \in \mathbb{R}^n$. The following problems are the most important.

1. The shortest vector problem(SVP). Find the shortest non-zero vector in L. Another problem is to find the shortest non-zero vector in L up to a approximation factor. That is find a non-zero $b \in L$ such that $|b| \leq \gamma \cdot |a|$ for any $a \in L$ for a fixed approximation factor $\gamma \geq 1$.

2. The closest vector problem(CVP). Find a vector $b \in L$ closest to a. In other words, $|a - b| \leq |a - c|$ for any $c \in L$. Similar problem, find $b \in L$ closest to a up to a approximation factor. In other words, $|a - b| \leq \gamma \cdot |a - c|$ for any $c \in L$ for a fixed approximation factor $\gamma \geq 1$.

1.8 LWE problem and Regev's cryptosystem

Let q be a prime and $m \geq n$ be natural numbers. Also let A be an integer $m \times n$-matrix of rank n modulo q and $e = (e_1, \ldots, e_m)$ be a column vector with entries generated independently according to the distribution

$$p_0, p_1, \ldots, p_{q-1} \tag{5}$$

on residues modulo q. Also let $x = (x_1, \ldots, x_n)$ be a secret column vector and $Ax - e \equiv z \mod q$, where $z = (z_1, \ldots, z_m)$. The Learning With Errors (LWE) problem is to find x

given A, z, q and the distribution (5). For example, let $m = 7, n = 4$ and $q = 5$. Let's set

$$A = \begin{pmatrix} 3 & 0 & 1 & 3 \\ 0 & 2 & 3 & 0 \\ 1 & 2 & 2 & 3 \\ 4 & 1 & 1 & 2 \\ 2 & 2 & 4 & 1 \\ 2 & 1 & 4 & 0 \\ 1 & 4 & 0 & 2 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \quad x = \begin{pmatrix} 2 \\ 0 \\ 4 \\ 1 \end{pmatrix}, \quad Ax = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 4 \\ 0 \\ 4 \\ 3 \end{pmatrix} \mod 5.$$

Given A, Ax and q it is easy to find x by solving a system of linear equations modulo q. However, one can disturb Ax with a secret error vector, e.g., $e = (0, 1, -1, 0, 1, 0, 1)$, whose entries were taken independently according to a public probability distribution. One gets

$$z = Ax - e = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 4 \\ 4 \\ 4 \\ 2 \end{pmatrix} \mod 5.$$

Now it is difficult to recover x given A, z, q and the probability distribution. The security of Regev's public key cryptosystem is based on the hardness of the LWE problem.

1.9 Regev's cryptosystem

Let $\alpha > 0$ and let $\phi_\alpha(x) = e^{-\frac{\pi x^2}{\alpha^2}}/\alpha$ denote one-dimensional Normal (Gaussian) probability density function with mean 0 and variance $\alpha^2/2\pi$. One may consider a restriction of ϕ to integers $\mathbb{Z}$ which produces a probability distribution on $\mathbb{Z}$ with density function

$$D_\alpha(x) = \phi_\alpha(x) / \sum_{z \in \mathbb{Z}} \phi_\alpha(z). \tag{6}$$

This distribution is called discrete Gaussian distribution. Given an odd prime number q, the distribution may be further restricted to the interval $-\frac{q-1}{2}, \ldots, -1, 0, 1, \ldots, \frac{q-1}{2}$ which represent residues modulo q and that gives a probability distribution (5).

Private Key Private key is a column vector $x = (x_1, x_2, \ldots, x_n)$ whose entries are residues modulo q.

Public Key Public key is (A, z). Here A is a matrix of size $m \times n$ with entries modulo q and $z = (z_1, z_2, \ldots, z_m)$ is a column vector of residues modulo q such that $z \equiv$

$Ax - e \mod q$, where $e = (e_1, \ldots, e_m)$ be a column vector with entries generated independently according to the distribution D_α. Remark, that e is a secret vector. If one knows e, then one may deduce x given the public key (A, z) and vice versa.

Encryption In order to encrypt a bit $y \in \{0, 1\}$ one chooses a random row vector $s = (s_1, s_2, \ldots, s_m)$ of bits. The encryption for y is

$$(a, b) = (sA, sz + y \lfloor q/2 \rfloor) \mod q.$$

So the encryption of one bit y is a vector (a, b) of residues modulo q of length $n + 1$.

Decryption In order to decrypt (a, b) one computes

$$\begin{aligned} t &= b - ax = sz + y \lfloor q/2 \rfloor) - sAx \\ &= sz + y \lfloor q/2 \rfloor) - s(z + e) = y \lfloor q/2 \rfloor - se \mod q. \end{aligned} \tag{7}$$

One chooses the parameters α, m such that se is close to zero with high probability. The decryption of (a, b) is 1 is t is close to $q/2$ and the decryption is 0 if t is close to 0.

In order to find the private key given the public key one has to solve an instance of the LWE problem.

1.10 Goldreich-Goldwasser-Halevi (GGH) signatures

The system was published in 1997. Let $R = (r_{ij})$ be an $n \times n$ integer non-singular matrix with small entries. The columns of R is the basis of a lattice L. The basis R is the system's private key. One also fixes a public threshold $\tau > 0$ which depends on R. For instance, let $\gamma = \max_i \sum_{j=1}^n |r_{ij}|$ be the maximum ℓ_1-norm of the rows in R. Then $\tau = \gamma\sqrt{n}/2$.

Let U be a random $n \times n$ integer unimodular(of determinant ± 1) matrix and $B = RU$. The columns of B is a basis of the same lattice L, because $BU^{-1} = R$, where U^{-1} is an integer matrix. The basis B is the system's public key. Generally, it is difficult to recover R given B.

The message (a hash value of the message) is encoded by an integer vector m of size n. The signature s is computed by the rule

$$s = R\lfloor R^{-1}m \rceil,$$

where $R^{-1}m$ is a vector with rational entries and $\lfloor R^{-1}m \rceil$ is its entry-wise rounding.

To verify the signature m, s one first checks that $s \in L$. That holds if and only if the system $s = Bx$ has an integer solution x as B is a basis for L. One then checks whether $|s - m| \leq \tau$. The signature s is accepted if the both checks are passed.

Let's prove that the verification really works. According to the algorithm $\lfloor R^{-1}m \rceil = R^{-1}m + x$, where $x = (x_1, \ldots, x_n)^T$ and $|x_i| \leq 1/2$. So $s - m = Rx$. Let $r_1, \ldots, r_n$ denote the rows of R. Then

$$s - m = Rx = (r_1 \cdot x, \ldots, r_n \cdot x)^T.$$

So

$$|s - m|^2 = (r_1 \cdot x)^2 + \ldots + (r_n \cdot x)^2 \leq \frac{1}{4} \left((\sum_{j=1}^{n} |r_{1j}|)^2 + \ldots + (\sum_{j=1}^{n} |r_{nj}|)^2 \right) \leq \frac{n\gamma^2}{4} = \tau^2.$$

Therefore, $|s - m| \leq \tau$.

The system was broken by Nguyen and Regev in 2006. Let $\mathcal{P}_{1/2}(R)$ denote the set of column vectors Rx for $x = (x_1, \ldots, x_n)^T$ and $|x_i| \leq 1/2$. The set is called the hidden parallelepiped. By definition, $s = R(R^{-1}m + x) = m + Rx$ and so $s - m \in \mathcal{P}_{1/2}(R)$. Therefore each message/signature gives one point inside this parallelepiped. The vertices of the parallelepiped are $(\pm R_1 \ldots \pm R_n)/2$, where $R_1, \ldots, R_n$ are columns of the matrix R. Collecting a number of such points one reconstructs the private matrix R up to a column permutations and multiplications by ± 1. That is enough to forge signatures.

For instance, 300 uniformly distributed points inside $\mathcal{P}_{1/2}(R)$ in dimension 2 are shown in Fig.1. One sees $(R_1 + R_2)/2 = \begin{pmatrix} 1.5 \\ 0.5 \end{pmatrix}$ and $(R_1 - R_2)/2 = \begin{pmatrix} -0.5 \\ 1.5 \end{pmatrix}$, where R_1, R_2 are the columns of R. So $R = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}$. Attacking GGH in dimension 200 requires about 20000 signatures.

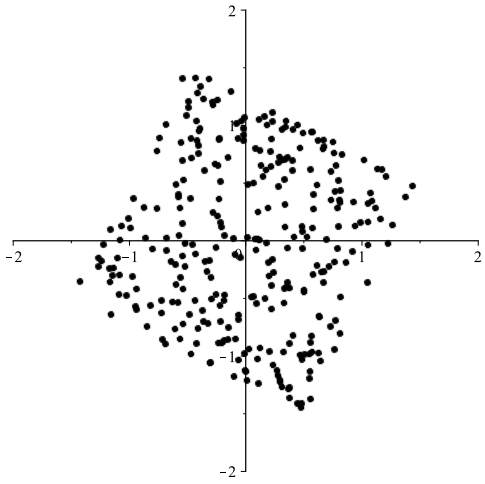

Figure 1: GGH Signature Cryptanalysis