Sorbonne Université

Campus Pierre et Marie Curie
Université de Bordeaux

Master Mathématiques et Applications

Sur la difficulté de module unique-SVP de rang 2

Présenté par :
Afonso Li

Semestre 2
Année universitaire 2023-2024

Remerciements

Merci surtout à mon encadrante, Alice, qui m'a beaucoup aidé depuis que nous avons pris contact. Merci d'avoir été si généreuse et patiente pour m'expliquer et m'introduire dans ce nouveau domaine de cryptographie. Merci aussi à Guilhem qui m'a beaucoup aidé dans plein de choses. J'espère avoir aussi l'opportunité de continuer à vous voir l'année prochaine.

Résumé

Avec le développement des ordinateurs quantiques, qui sont plus qu'une simple boîte noire, nous disposons d'algorithmes beaucoup plus puissants sur ces machines. On se questionne si la sécurité de nos cryptosystèmes actuels est résistante face à des algorithmes/attaques quantiques, et la réponse est non. En effet, l'algorithme de Shor montre que les cryptosystèmes actuels à clés publiques comme RSA, Diffie-Hellman et Diffie-Hellman à base de courbes elliptiques sont cassables avec des algorithmes quantiques plutôt facilement. D'un autre côté, nous avons les cryptosystèmes à base de réseaux euclidiens. Non seulement ils sont résistants aux attaques classiques, mais ils sont aussi supposés résistants aux algorithmes quantiques, notamment le problème de retrouver un plus court vecteur du réseau est supposé être NP-dur. C'est là que nous introduisons un cryptosystème basé sur les réseaux, appelé le cryptosystème NTRU, introduit entre 1996 et 1998 dans [HPS98].

Depuis sa création, nous essayons de comprendre sa sécurité et ses réductions en d'autres problèmes sur les réseaux, ce qui reste très difficile. Dans ce rapport, nous allons explorer plusieurs versions du problème NTRU et voir l'apparition de certains modules de rang 2 bien particuliers. Ensuite, nous généraliserons la notion de NTRU vers d'autres modules de rang 2 quelconques.

Nous parlons souvent des réductions entre les problèmes, il faut imaginer qu'une réduction est juste une façon de comparer les difficultés des problèmes. Si P se réduit à Q, alors Q est dit au moins aussi dur à résoudre que P.

NTRU est un problème de réseau euclidien basé sur la difficulté de trouver un vecteur court dans un réseau, connu sous le nom de SVP. Nous avons observé de nombreuses attaques et formulations de ce problème. Nous verrons que non seulement NTRU est un problème de réseau sur des modules de rang 2 très particuliers qu'on appelle unique-SVP_2, mais aussi une sorte de représentant de unique-SVP_2. Dans ce rapport, nous introduirons les notions nécessaires sur les réseaux euclidiens non structurés, puis nous présenterons des notions provenant de la théorie des nombres pour construire des réseaux euclidiens dits structurés. Nous verrons quelques attaques sur le problème NTRU, en utilisant par exemple la présence de sous-corps dans un corps de nombres ou en utilisant des algorithmes de réduction de réseau tels que BKZ. Nous verrons comment réduire un unique-SVP_2 vers NTRU. L'objectif de ce stage est d'utiliser cette réduction pour transposer une attaque sur NTRU à une attaque sur les modules de rang 2 en général.

Nous commencerons par faire une introduction sur les objets mathématiques avec lesquels nous travaillons et les problèmes supposés difficiles sur les réseaux euclidiens. Qui dit problèmes dit aussi algorithmes destinés à les résoudre ; nous consacrerons une partie aux algorithmes de réduction de réseaux. Ensuite, nous introduirons une famille de réseaux dits structurés, basés sur des idéaux et des modules sur un anneau d'entiers d'un corps de nombres. En présentant cette nouvelle

famille de réseaux structurés, nous verrons qu'elle présente des avantages et des inconvénients. Enfin, nous introduirons un réseau clé pour nous, le réseau NTRU, construit à partir d'un module libre de rang 2. Nous ne parlerons pas dans ce rapport du fait que le cryptosystème NTRU se distingue par la rapidité des processus de chiffrement et de déchiffrement et par sa résistance à la cryptanalyse quantique. Nous nous concentrerons principalement sur sa sécurité.

Dans la section sur le module NTRU, nous examinerons deux attaques sur différentes versions du problème NTRU, l'une par BKZ et l'autre par les sous-corps. Ensuite, nous nous intéresserons au problème unique-SVP_2. Nous utiliserons les travaux de l'article [FPMS22] pour montrer comment unique-SVP_2 se ramène à un problème NTRU pour lequel nous connaissons quelques attaques.

Pour conclure, nous expliquerons une tentative d'avoir un algorithme en temps polynomial qui résout le problème unique-SVP_2 et les idées que nous avons eues.

Une autre observation que nous souhaitons aborder concerne la complexité asymptotique des deux attaques sur NTRU. Nous nous demandons si elles sont liées en exploitant la connexion entre les sous-blocs de matrice du réseau NTRU et le réseau NTRU vu dans les sous-corps, en utilisant soit la norme relative soit la trace relative.

Pendant le stage, nous avons ressenti que les questions qui nous intéressaient nécessitaient de plus en plus d'outils en théorie des nombres. La voie naturelle est alors de transformer les réseaux euclidiens en réseaux algébriques et de transporter l'arithmétique de $\mathbb{Z}$ à un anneau d'entiers d'un corps de nombres qui est plus compliqué puisque ce n'est plus nécessairement un anneau principal.

Table des matières

ACRONYMES (en anglais)

GSO : Gram-Schmidt Orthogonalization

HNF : Hermite Normal Form

LLL : Lenstra-Lenstra-Lovász Algorithm

BKZ : Block Korkine Zolotareff Algorithm

SVP : Shortest Vector Problem

CVP : Closest Vector Problem

Notation

Les notations log et ln représentent logarithme en bases 2 et e.

Si $b_1, \ldots, b_n \in \mathbb{R}^m$ sont des vecteurs linéairement indépendants, alors nous utilisons la notation $(b_1^*, \ldots, b_n^*)$ pour l'orthogonalisée de Gram-Schmidt.

Nous utilisons $|||\cdot|||$ pour la norme matricielle induite par la norme euclidienne.

Un peu de compléxité. Nous allons utiliser des notations classiques "**grand O**". Soient $f, g : \mathbb{N} \to \mathbb{R}$, on dit que :

- $f(n) = O(g(n))$ si $\exists k > 0$ et $n_0 > 0$ tel que $\forall n \geqslant n_0, |f(n)| \leqslant k|g(n)|$.
- $f(n) = \Theta(g(n))$ si $\exists k_1, k_2 > 0$ et $n_0 > 0$ tel que $\forall n \geqslant n_0, k_1|g(n)| \leqslant |f(n)| \leqslant k_2|g(n)|$.
- $f(n) = \omega(g(n))$ si $\forall k > 0$ et $\exists n_0 > 0$ tel que $\forall n \geqslant n_0, |f(n)| \geqslant k|g(n)|$.
- $f(n) = \Omega(g(n))$ si $\exists k > 0$ et $n_0 > 0$ tel que $\forall n \geqslant n_0, f(n) \geqslant kg(n)$.

De plus, on note $\tilde{O}(f) := O(f) \times (\log(f))^{O(1)}$. Alors $\tilde{O}(d^t) = O(d^t \ \mathsf{poly}(\log d))$. Soit A un ensemble, on note $U(A)$ pour dire qu'on utilise la loi uniforme sur A.

Notions cryptographiques :

Définition 0.0.1. (Adversaire $\mathcal{A}$)
L'adversaire $\mathcal{A}$ est un algorithme qui a pour but de casser un cryptosystème ou une hypothèse

Définition 0.0.2. Une instance d'un problème correspond à l'ensemble des entrées nécessaires pour calculer une solution à ce problème.

Définition 0.0.3. La distance statistique entre deux lois de probabilités discrètes D_1 et D_2 de supports compatibles est défini comme $dist(D_1, D_2) := \frac{1}{2}\sum_x |D_1(x) - D_2(x)|$. On note par $D_1 \approx_\varepsilon D_2$ si $dist(D_1, D_2) \leqslant \varepsilon$ pour un $\varepsilon > 0$.

Si X est un ensemble fini, on note $U(X)$ la loi uniforme sur X.

Définition 0.0.4. (Loi normale discrète) Soit $S \in GL_n(\mathbb{R})$. La fonction densité normale avec paramètre S est défini sur $\mathbb{R}^n$ par

$$\rho_S(x) = \exp(-\pi \|S^{-1}x\|^2).$$

Lorsque S est une matrice diagonale avec une seule valeur $\sigma > 0$, on note $\rho_\sigma = \rho_S$.

1 Prérequis sur les réseaux euclidiens

1.1 Définitions

Définition 1.1.1. Un réseau (euclidien) $\mathcal{L} \subseteq \mathbb{R}^n$ est l'ensemble des combinaisons entières des vecteurs linéaires indépendantes $b_1, \ldots, b_n \in \mathbb{R}^n$. C'est à dire :

$$\mathcal{L}(b_1, \ldots, b_n) = \{z_1 b_1 + z_2 b_2 + \cdots + z_n b_n | z_i \in \mathbb{Z}\}.$$

L'entier n est dit la dimension (ou le rang) du réseau $\mathcal{L}$.

Remarque 1.1.2. Nous appelons l'ensemble $\{b_1, \ldots, b_n\}$ une base du réseau $\mathcal{L}(b_1, \ldots, b_n)$. Nous avons aussi une autre notation peut-être plus pratique pour $\mathcal{L}(b_1, \ldots, b_n)$ qui est $\mathcal{L}(B) := B\mathbb{Z}^n := \{Bz : z \in \mathbb{Z}^n\}$ où $B = (b_1 \mid \cdots \mid b_n) \in \mathbb{R}^{n \times n}$.

Question 1.1.3. Est-ce que la base d'un réseau euclidien $\mathcal{L}(b_1, \ldots, b_n)$ est unique ? La réponse est clairement non !

Exemple 1.1.4. Soit $n = 2$ et (b_1, b_2) une base d'un réseau euclidien $\mathcal{L}$ alors bien sûr que $(b_1, -b_2)$ l'est aussi. On peut voir par exemple que :

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

est aussi une base pour le réseau $\mathbb{Z}^2$.

Comme la base d'un réseau n'est pas unique, nous utiliserons plus souvent la notation $\mathcal{L} \in \mathbb{R}^n$.

Lemme 1.1.5. Soit $\mathcal{L} \subseteq \mathbb{R}^n$, alors $\mathcal{L}$ est un réseau si et seulement si $\mathcal{L}$ est un sous-groupe discret de $\mathbb{R}^n$.

Nous avons vu que un même réseau peut-être engendré par des bases différentes. Nous avons par exemple vu que le réseau $\mathbb{Z}^2$ est aussi engendré par la base

$$B = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

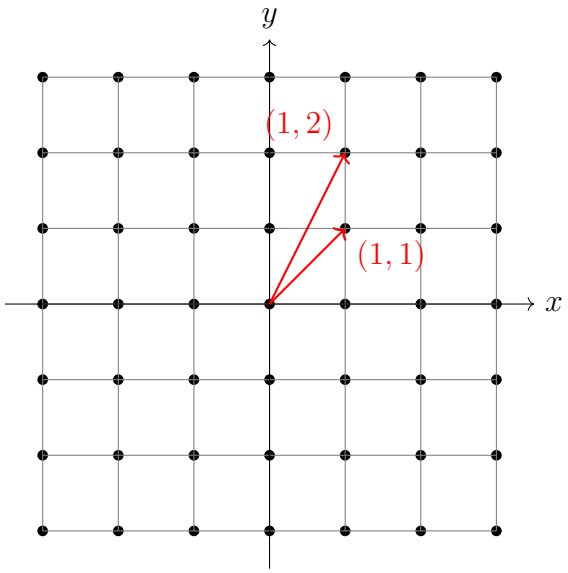

FIGURE 1 – Un réseau un peu basique

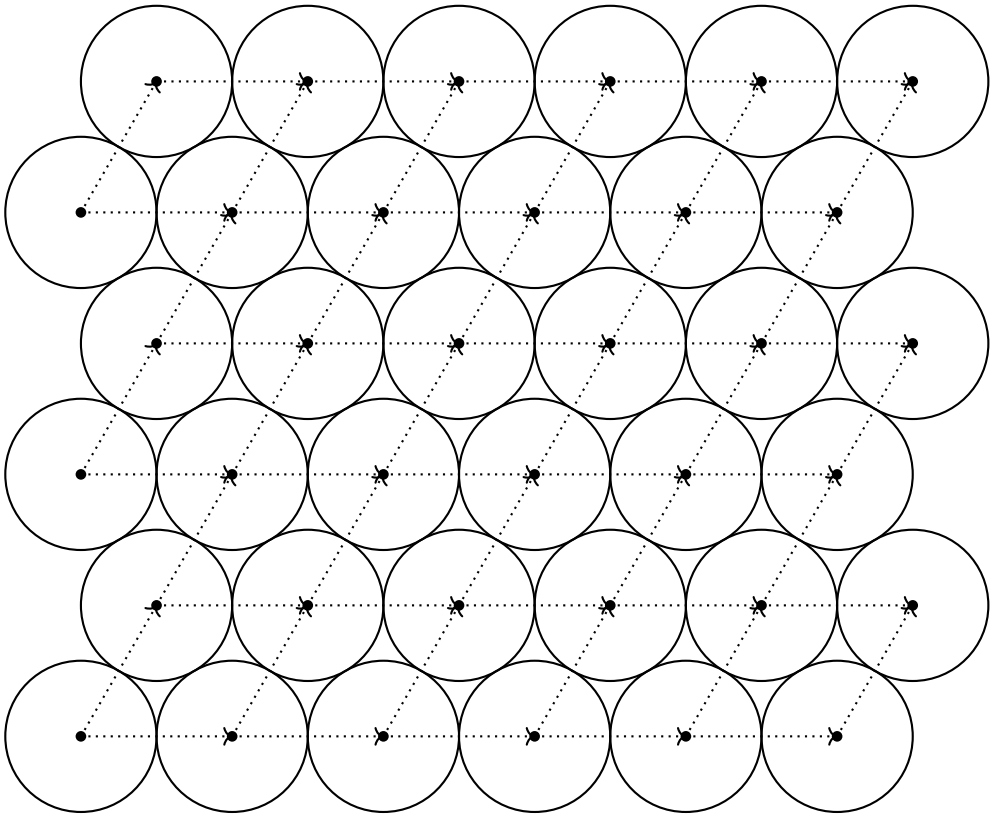

FIGURE 2 – Réseau hexagonal

Théorème 1.1.6. Soient B et C deux bases. Alors $\mathcal{L}(B) = \mathcal{L}(C)$ si et seulement s'il existe une matrice inversible $U \in GL_n(\mathbb{Z})$ telle que $B = CU$.

Démonstration. Soit $U \in GL_n(\mathbb{Z})$ tel que $B = CU$, on écrit $C = BU^{-1}$. Nous avons alors ces inclusions :

$$\mathcal{L}(B) \subseteq \mathcal{L}(CU) \subseteq \mathcal{L}(C) = \mathcal{L}(BU^{-1}) \subseteq \mathcal{L}(B).$$

Donc $\mathcal{L}(B) = \mathcal{L}(C)$. Supposons maintenant que B et C deux matrices de bases qui forment le même réseau $\mathcal{L} = \mathcal{L}(B) = \mathcal{L}(C)$. Il existe des matrices entières U, V tel que $B = CU$ et $C = BV$. Ainsi,

$$B = CU = BVU \iff B(I - VU) = 0$$

Or comme B est une application linéaire injective, alors,

$$I - VU = O \iff VU = I.$$

De façon similaire, $UV = I$. Donc $U \in GL_n(\mathbb{Z})$ et $U^{-1} = V$.

□

Corollaire 1.1.7. Soit $U \in GL_n(\mathbb{R})$, alors U est une base pour $\mathbb{Z}^n$ si et seulement si $U \in M_n(\mathbb{Z})$ et $U^{-1} \in M_n(\mathbb{Z})$.

Le théorème précédent montre que les matrices entières inversibles U peuvent être utilisées pour transformer une base de réseau B en toute autre base pour le même réseau. En pratique, il est souvent plus facile ou plus pratique de transformer B en une base différente en exerçant des opérations.

Définition 1.1.8. Une opération de colonne sur une matrice $B \in \mathbb{R}^{n \times n}$ fait partie de :

- échanger(i, j) : Échanger deux vecteurs i.e. $(b_i, b_j) \leftarrow (b_j, b_i)$ pour $i \neq j$.
- inverser(i) : Changer le signe d'un vecteur i.e. $(b_i) \leftarrow (-b_i)$.
- ajouter(i, c, j) : Ajouter un multiple entier d'un vecteur de la base à un autre vecteur i.e. $(b_i) \leftarrow (b_i + cb_j)$ pour $i \neq j$ et $c \in \mathbb{Z}$.

Définition 1.1.9. Une matrice $U \in \mathbb{Z}^{n \times n}$ est unimodulaire si $|\det(U)| = 1$.

Lemme 1.1.10. Une matrice $U \in GL_n(\mathbb{Z})$ est unimodulaire.

Nous allons montrer qu'une matrice modulaire s'exprime par une séquence d'opérations de colonnes. Nous allons définir la forme normale d'Hermite d'une matrice inversible à coefficients entiers.

Définition 1.1.11. La forme normale d'Hermite (HNF) d'une matrice entière inversible H de taille $n \times n$ a la structure suivante :

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ 0 & h_{22} & h_{23} & \cdots & h_{2n} \\ 0 & 0 & h_{33} & \cdots & h_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & h_{nn} \end{pmatrix}$$

où :

1. h_{ii} (pour $i = 1, 2, \ldots, n$) sont des entiers positifs ou nuls
2. $0 \leq h_{ij} < h_{ii}$ (pour $i < j$)
3. Tous les éléments en dessous de la diagonale sont des zéros.

Théorème 1.1.12. Soit $U \in \mathbb{Z}^{n \times n}$, nous pouvons réaliser une séquence d'opérations sur les colonnes σ tel que $\sigma(U)$ est sous la forme HNF.

Corollaire 1.1.13. Soit U une matrice unimodulaire, il existe une séquence d'opérations de colonnes σ tel que $\sigma(U) = I$.

Corollaire 1.1.14. Soit $U \in \mathbb{Z}^{n \times n}$, alors les propositions suivantes sont équivalentes :

1. $U = \sigma(I)$ pour une certaine séquence d'opérations élémentaires de colonnes σ.
2. $U \in GL_n(\mathbb{Z})$.
3. U est unimodulaire.

1.2 Rappels d'algèbre linéaire

Toute base B peut-être transformée en une base orthogonale pour le même espace vectoriel en utilisant la méthode bien connue d'orthonormalisation de Gram-Schmidt. Supposons que nous ayons des vecteurs $B = (b_1, \ldots, b_n) \in \mathbb{R}^{d \times n}$ générant un espace vectoriel $V = \text{Vect}(B)$. Ces vecteurs ne sont pas nécessairement orthogonaux (ni même linéairement indépendants), mais nous pouvons toujours trouver une base orthogonale $B^* = (b_1^*, \ldots, b_n^*)$ pour V où b_i^* est la composante de b_i orthogonale à l'espace linéaire $\text{Vect}([b_1, \ldots, b_{i-1}])$ des vecteurs précédents. Nous rappelons la définition de la projection orthogonale.

Définition 1.2.1. L'orthogonalisation de Gram-Schimidt d'une base $B = (b_1, \ldots, b_n)$ est la matrice $B^* = (b_1^*, \ldots, b_n^*)$ où les $b_i^* = \pi_i(b_i)$ et π_i est la projection orthogonal à l'espace engendré par les vecteurs $b_1, \ldots, b_{i-1}$.

Lemme 1.2.2. L'orthogonalisation de Gram-Schmidt pour $B = (b_1, \ldots, b_n)$ est donnée par la formule suivante :

$$b_i^* = b_i - \sum_{j<i} \mu_{i,j} b_j^* \quad \text{où} \quad \mu_{i,j} = \frac{\langle b_i,\ b_j^* \rangle}{\langle b_j^*,\ b_j^* \rangle}$$

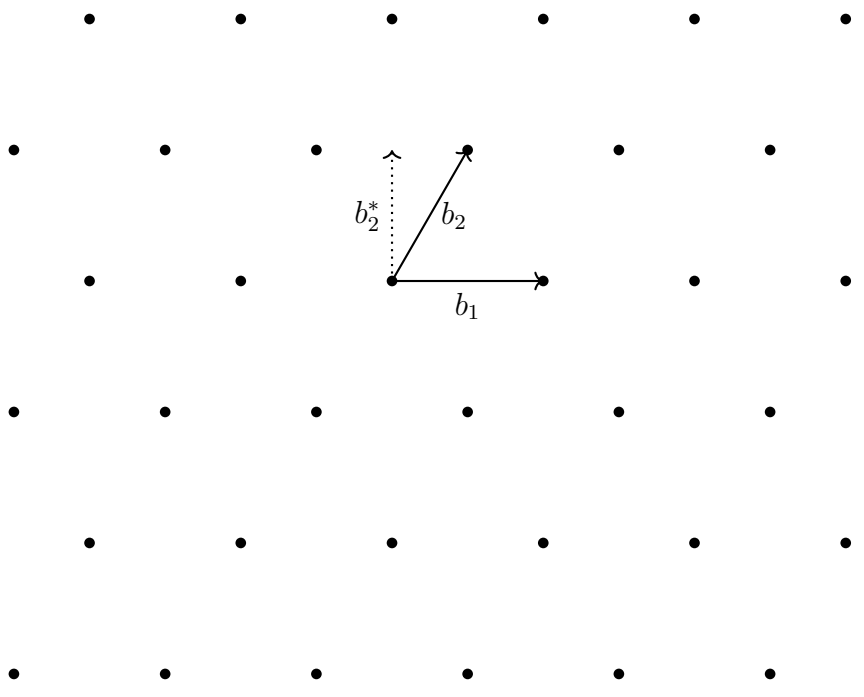

FIGURE 3 – La projection orthogonale

Comme les réseaux $\mathcal{L} \subseteq \mathbb{R}^n$ sont effectivement des objets géométriques, il est alors naturel de s'intéresser à la norme des vecteurs du réseau.

Pour nous, l'une des propriétés les plus importantes des réseaux est la norme du plus court vecteur, définie de la façon suivante.

Définition 1.2.3. Nous notons $\lambda_1(\mathcal{L}) := \min_{x \in \mathcal{L} \neq 0} \|x\|$ la quantité que l'on appelle norme du plus court vecteur du réseau $\mathcal{L}$.

Définition 1.2.4. (Successive Minima)
Soit $\mathcal{L} \subseteq \mathbb{R}^m$ de dimension n. Pour $i \leqslant n$, le i-ème minimum successif est défini comme

$$\lambda_i := \min(\rho > 0 : \dim_{\mathbb{R}}(\mathrm{Vect}_{\mathbb{R}}(\mathcal{L}) \cap B_m(0, \rho)) \geqslant i)$$

i.e., le plus petit rayon de la m-boule contenant i vecteurs linéairement indépendants de $\mathcal{L}$.

Théorème 1.2.5. (Banaszczyk)
Soit $\mathcal{L} \in \mathbb{R}^m$ un réseau de dimension n et $\mathcal{L}^\vee$ son dual, alors

$$1 \leqslant \lambda_1(\mathcal{L}) \lambda_n(\mathcal{L}^\vee) \leqslant n$$

Remarque 1.2.6. Nous pouvons avoir plus d'un plus court vecteur, à juste signe près, on peut prendre le vecteur $-x$. Ainsi, un vecteur court n'est jamais unique.

En cryptographie à base de réseaux, nous disposons de deux grandes familles de réseaux qui vont jouer un rôle indispensable pour énoncer deux grands problèmes : le problème du plus court vecteur dans un réseau (SIS - Short Integer Solutions problem) et le problème d'apprentissage avec erreurs (LWE - Learning With Errors problem). Mais avant cela, introduisons la notion de réseau q-aire.

Définition 1.2.7. Soit $\mathcal{L} \subseteq \mathbb{Z}^n$ un réseau. $\mathcal{L}$ est un réseau q-aire pour un entier $q \geqslant 2$ si $q\mathbb{Z}^n \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$.

Remarque 1.2.8. Les réseaux q-aire sont en correspondance 1-1 avec les codes linéaires de $\mathbb{Z}_q^n$.

Voici les exemples de réseaux q-aire les plus connus.

Définition 1.2.9. Soient $m > n \in \mathbb{N}^*$ et $q \geqslant 2$ des entiers naturels. Soit $A \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$.

1. Le réseau LWE associé à la matrice A est défini comme
   $\Lambda(A) := A \cdot (\mathbb{Z}/q\mathbb{Z})^n + q \cdot \mathbb{Z}^m := \{y \in \mathbb{Z}^m : \exists s \in (\mathbb{Z}/q\mathbb{Z})^n, y = A \cdot s \bmod q\}$
2. Le réseau SIS associé à la matrice A est défini comme
   $\Lambda^{\perp}(A) := \{x \in \mathbb{Z}^m : x^T \cdot A = 0 \bmod q\}$

Introduisons maintenant la notion de réseau dual.

Définition 1.2.10. Soit $\mathcal{L}$ un réseau. Le dual du réseau $\mathcal{L}$ est défini comme

$$\mathcal{L}^{\vee} := \{\hat{b} \in \mathrm{Vect}_{\mathbb{R}}(\mathcal{L}) : \forall b \in \mathcal{L}, \langle b,\ \hat{b} \rangle \in \mathbb{Z}\} \tag{1}$$

Remarque 1.2.11. Il est très facile de vérifier si un vecteur est dans le réseau dual. Si on prend une base $b_1 \dots b_n$ du réseau $\mathcal{L}$, alors un vecteur w est dans $\mathcal{L}^{\vee}$ si et seulement si $\langle w, b_i \rangle \in \mathbb{Z}, \forall i$.

Lemme 1.2.12. Soit $\mathcal{L}$ un réseau. Alors $\mathcal{L}^{\vee}$ est un réseau et $(\mathcal{L}^{\vee})^{\vee} = \mathcal{L}$.

Théorème 1.2.13. Le dual d'un réseau avec une base B est un réseau avec une base $D = B \cdot G^{-1}$ où $G = B^{\top}B$ est la matrice de Gram de B.

Lemme 1.2.14. Prouvez que pour tout $B, D \in \mathbb{R}^{n \times n}$, D est la base duale de B si et seulement si les conditions suivantes sont satisfaites :

— $\mathrm{Vect}(B) = \mathrm{Vect}(D)$, et
— $B^{\top}D = D^{\top}B = I$.

Parlons un peu des projections orthogonaux et le réseau dual.

Soient $B = (b_1, \ldots, b_n)$ une base pour un réseau et $D = (d_1, \ldots, d_n)$ sa base duale pour son réseau dual.

Remarque 1.2.15. En considérant un changement de base arbitraire $B \to BU$ où U est une matrice unimodulaire. Ensuite, le dual de la nouvelle base change comme $D \to DU^{-\top}$, car $(DU^{-\top})^\top(BU) = U^{-1}(D^\top B)U = I$.

Considérons maintenant le processus d'orthogonalisation de Gram-Schmidt. Soit $B = (b_1, \ldots, b_n)$ une base, et $B^* = (b_1^*, \ldots, b_n^*)$ son orthogonalisation de Gram-Schmidt. Soit $\pi_i(x) = x \perp (b_1, \ldots, b_{i-1})$, pour $i = 1, \ldots, n$, les opérations de projection associées à B^*. Considérons le réseau projeté

$$\pi_i(\mathcal{L}(B)) = \mathcal{L}((\pi_i(b_i), \ldots, \pi_i(b_n))).$$

Quel est le dual de $\pi_i(\mathcal{L}(B))$? Nous montrons maintenant que le dual de $\pi_i(\mathcal{L}(B))$ est le sous-réseau de $\mathcal{L}(D)$ généré par $d_i, \ldots, d_n$. (Remarquez qu'aucune projection orthogonale n'est appliquée au réseau dual).

Lemme 1.2.16. Soient $B, D \in \mathbb{R}^{n \times n}$ des bases duales, i.e., les réseaux qu'ils forment sont duaux. Pour tout $i \in [n]$, $(\pi_i(b_i), \ldots, \pi_i(b_n))$ et $(d_i, \ldots, d_n)$ sont des bases duales.

Définissons maintenant l'orthogonalisation de la base duale de la manière habituelle, mais en parcourant les vecteurs de base dans l'ordre inverse de d_n à d_1 :

$$d_i^\dagger = \tau_i(d_i) \quad \text{où} \quad \tau_i(x) = x \perp (d_{i+1}, \ldots, d_n).$$

Il en découle par dualité que pour tout i, le dual de $(b_1, \ldots, b_i)$ est la base projetée $(\tau_i(d_1), \ldots, \tau_i(d_i))$. En général, nous avons ce qui suit.

Théorème 1.2.17. Soit D le dual de B. Alors pour tout $i \leq j$, le dual de $(\pi_i(b_i), \ldots, \pi_i(b_j))$ est $(\tau_j(d_i), \ldots, \tau_j(d_j))$.

Lemme 1.2.18. $\Lambda(A)$ et $\Lambda^\perp(A)$ sont duaux entre eux à scalaire près : $\Lambda(A)^\vee = \frac{1}{q} \cdot \Lambda^\perp(A)$.

Démonstration. Tout d'abord, montrons $\Lambda^\perp(A) \subseteq q \cdot \Lambda(A)^\vee$,
Soit $x \in \Lambda^\perp(A)$, par définition $x^T \cdot A = 0 \bmod q$ i.e. $x^T A = qz$, où $z \in \mathbb{Z}^m$. Soit $y \in \Lambda(A)$, donc $y = As \bmod q$ i.e. $y = As + qz'$.
Alors

$$\begin{aligned}\langle x,\ y\rangle &= \langle x,\ As + qz'\rangle \\ &= \langle x,\ As\rangle + \langle x,\ qz'\rangle \\ &= \langle A^T x,\ s\rangle + \langle x,\ qz'\rangle \\ &= \langle qz,\ s\rangle + \langle x,\ qz'\rangle \\ &= q(\langle z,\ s\rangle + \langle x,\ z'\rangle)\end{aligned}$$

d'où $\frac{1}{q}\langle x,\ y\rangle \in \mathbb{Z}$, ainsi, $\frac{1}{q}x \in \Lambda(A)^\vee$.
Il reste maintenant l'autre implication ; montrons que $q\Lambda(A)^\vee \subseteq \Lambda^\perp(A)$,
Soit $q\hat{y} \in q\Lambda(A)^\vee$, par définition, $\langle \hat{y},\ y\rangle \in \mathbb{Z}$ pour tout $y \in \Lambda(A)$. En particulier, cela tient pour les vecteurs de base de $\Lambda(A)$: $\Lambda(A)$ est généré par les colonnes de A et les vecteurs $qe_1, \ldots, qe_n$, on a $y^T A \in \mathbb{Z}^n$ et $qy \in \mathbb{Z}^m$. Ainsi $(qy^T)A = q(y^T A) = 0 \bmod q$. □

1.3 Déterminant et théorème de Minkowski

Un des invariants de réseau est son déterminant. Définissons cette quantité.

Définition 1.3.1. Soit $\mathcal{L}$ un réseau de base B. Nous définissons le déterminant de $\mathcal{L}$ comme la quantité suivante

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

Lorsque le réseau est de plein rang, la définition devient $det(\mathcal{L}) := |\det(B)|$.

Une définition équivalente sera de définir le déterminant d'un réseau comme le volume du domaine fondamental du réseau

$$\mathcal{P}(B) = \{\sum_{i=1}^{n} x_i b_i | \forall 0 \leqslant x_i < 1\}.$$

Théorème 1.3.2. (Inégalité de Hadamard)
Soit $\mathcal{L}(b_1, \ldots, b_n)$ un réseau, alors $\det(\mathcal{L}) \leqslant \prod_i \|b_i\|$.

Lemme 1.3.3. Soit $\mathcal{L}$ un réseau de base B. Alors $B^\vee := B(B^T \cdot B)^{-1}$ est une base du réseau $\mathcal{L}^\vee$ et $\det(\mathcal{L}^\vee) = \frac{1}{\det(\mathcal{L})}$.

Remarque 1.3.4. Si $\mathcal{L}$ est un réseau de plein rang alors $B^\vee = B^{-T}$.

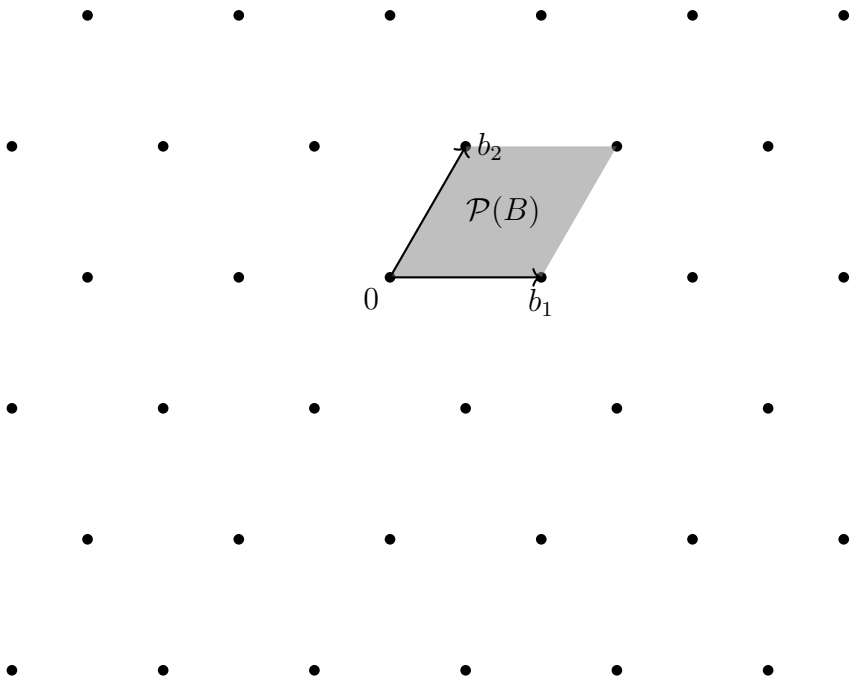

FIGURE 4 – Domaine fondamental du réseau hexagonal

Proposition 1.3.5. Soient q un nombre premier, $m > n > 0$ deux entiers et $A \in (\mathbb{Z}_q)^{m\times n}$. Alors

$$\det(\Lambda(A)) = q^{m-r}$$
$$\det(\Lambda^{\perp}(A)) = q^{r}$$

où r est le rang de la matice A.

Nous allons maintenant parler un peu de la décomposition QR. La décomposition QR est une factorisation d'une matrice rectangulaire A en un produit de deux matrices, Q (orthogonale) et R (triangulaire supérieure). Pour une matrice A de dimensions $m \times n$, la décomposition QR est donnée par :

$$B = QR \tag{2}$$

où

— Q est une matrice orthogonale de dimensions $m \times m$,

— R est une matrice triangulaire supérieure de dimensions $m \times n$ et $r_{ii} \geqslant 0$.

Le processus de décomposition QR peut-être réalisé à l'aide de la méthode de Gram-Schmidt ou d'autres algorithmes plus sophistiqués.

On considère le procédé de Gram-Schmidt appliqué aux colonnes de la matrice $B = (b_1, \cdots, b_n)$. L'algorithme présenté ci-dessous convient à une matrice de rang $n \times n$.

Sachant que

$$b_i^* = b_i - \sum_{j<i} \mu_{i,j} b_j^* \quad \text{où} \quad \mu_{i,j} = \frac{\langle b_i,\ b_j^* \rangle}{\langle b_j^*,\ b_j^* \rangle}$$

En posant $e_i = \frac{b_i^*}{\|b_i^*\|}$. On peut bien sûr écrire

$$\begin{aligned} b_1 &= \langle e_1, b_1 \rangle e_1 \\ b_2 &= \langle e_1, b_2 \rangle e_1 + \langle e_2, b_2 \rangle e_2 \\ &\vdots \\ b_n &= \sum_{j=1}^{n} \langle e_j, b_n \rangle e_j \end{aligned}$$

où $\langle e_i, b_i \rangle = \|b_i^*\|$. Ceci s'écrit matriciellement :

$$B = QR$$

avec

$$Q = (e_1, \cdots, e_n) \quad \text{et} \quad R = \begin{pmatrix} \langle e_1, b_1 \rangle & \langle e_1, b_2 \rangle & \langle e_1, b_3 \rangle & \cdots \\ 0 & \langle e_2, b_2 \rangle & \langle e_2, b_3 \rangle & \cdots \\ 0 & 0 & \langle e_3, b_3 \rangle & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Lemme 1.3.6. Soit $\mathcal{L} \subseteq \mathbb{R}^m$ un réseau et B une base pour $\mathcal{L}$ et soit R la matrice de la décomposition QR. Alors

$$\det(\mathcal{L}) = \prod_{i \leqslant n} r_{ii}$$

où r_{ij} est la coefficient ij de la matrice R.

Ce lemme va nous donner un lien entre plus court vecteur de $\mathcal{L}$ et les coefficients de R.

Théorème 1.3.7. $\min_{i \leqslant n} r_{ii} \leqslant \lambda_1(\mathcal{L}) \leqslant r_{11}$ i.e., Si on avait une base $B = (b_1, \ldots, b_n)$ et $B^* = (b_1^*, \ldots, b_n^*)$ sa matrice formée des vecteurs de Gram-Schmidt, alors $\lambda_1(\mathcal{L}(B)) \geqslant \min_i \|b_i^*\|$.

Pour toute base B, nous avons $\lambda_1(B) \leq \min_i \|b_i\|$, car chaque colonne de B est un vecteur de réseau non-nul. Nous aimerions obtenir une meilleure borne, et plus précisément, une borne qui ne dépend pas du choix de la base. Clairement, les réseau avec une distance minimale arbitrairement grande peuvent être facilement obtenus simplement en multipliant n'importe quel réseau donné par une constante $c > 0$ pour obtenir $\lambda_1(c \cdot \mathcal{L}) = c \cdot \lambda_1(\mathcal{L})$. Que se passe-t-il si nous normalisons le réseau de sorte que

$\det(\mathcal{L}) = 1$? Par définition du déterminant, ce sont des réseaux de densité 1, c'est-à-dire avec environ un point de réseau par unité de volume d'espace. Un tel réseau peut-il encore avoir une distance minimale arbitrairement grande ? De façon équivalente, nous demandons s'il est possible de borner le rapport $\lambda_1(\mathcal{L})/\det(\mathcal{L})^{1/n}$ pour tout réseau de dimension n. (Remarquez que la quantité $\lambda_1(\mathcal{L})/\det(\mathcal{L})^{1/n}$ est invariante sous mise à l'échelle linéaire car $\det(c\cdot\mathcal{L}) = c^n \cdot \det(\mathcal{L})$). Pour des raisons historiques, les mathématiciens ont défini et étudié le carré de cette quantité, qui est appelé la constante d'Hermite d'un réseau.

Lemme 1.3.8. (Blichfeldt)
Soit $\mathcal{L}(B) \subseteq \mathbb{R}^n$ un réseau et $X, Y \subseteq \mathbb{R}^n$ deux parties mesurables. Supposons que $X = \mathcal{P}(B)$ un domaine fondamental de $\mathcal{L}$ et $\forall x, y \in Y$,

$$x - y \in \mathcal{L} \Rightarrow x = y.$$

Alors $\mu(Y) \leqslant \mu(X)$ où μ est la mesure de Lebesgue sur $\mathbb{R}^n$.

Démonstration. Par hypothèse, nous pouvons couvrir $\mathbb{R}^n$ en translatant le domaine fondamental de $\mathcal{L}$ i.e. $\mathbb{R}^n = \cup_{a\in\mathcal{L}}(X + a)$. En intersectant avec Y et par invariance par translation de la mesure de Lebesgue il vient

$$(Y \cap (X + a)) - a = X \cap (Y - a),$$

on a que

$$\mu(Y) = \sum_{a\in\mathcal{L}} \mu(X \cap (Y - a)).$$

En utilisant l'hypothèse sur Y, les $Y - a$ sont des parties disjointes de $\mathbb{R}^n$, donc $\mu(Y) = \sum_{a\in\mathcal{L}} \mu(X \cap (Y - a)) \leqslant \mu(X)$. □

Théorème 1.3.9. (Lemme du corps convexe de Minkowski)
Soit $C \subseteq \mathbb{R}^n$ une partie mesurable symétrique convexe et soit $\mathcal{L} \subseteq \mathbb{R}^n$ un réseau. Si $\det(\mathcal{L}) < \mu(C)/2^n$ ou si C est compact **et** $\det(\mathcal{L}) \leqslant \mu(C)/2^n$ alors il existe un élément non-nul dans $\mathcal{L} \cap C$.

Démonstration. Soit B une base pour $\mathcal{L}$. le sous groupe/réseau $\mathcal{L}' = 2\mathcal{L}$ de $\mathcal{L}$ de base $B' = (2 \cdot b_1, \ldots, 2 \cdot b_n)$ et $\det(\mathcal{L}') = 2^n \det(\mathcal{L})$.

Supposons que $\det(\mathcal{L}') < \mu(C)$. Appliquons le lemme de Blichfeldt à $X = \mathcal{P}(B')$ et $Y = C$ nous dit que il existe $x, y \in C$ distincts tels que $x - y \in \mathcal{L}' = 2\mathcal{L}$. Or C est convexe symétrique donc $\frac{x-y}{2}$ est dans $C \cap \mathcal{L}$, qui traite un cas.

Maintenant, supposons que $\det(\mathcal{L}') \leqslant \mu(C)$ avec C compact. Pour tout $\varepsilon > 0$, l'épaississement

$$C_\varepsilon = \{z \in \mathbb{R}^n \mid \exists x \in C,\ |z - x| < \varepsilon\}.$$

où C est un ensemble borné, symétrique et convexe. C_ε contient C et a une mesure supérieure à $\mu(C)$. En utilisant le cas précédent, on obtient un élément z_ε non nul de $\mathcal{L} \setminus \{0\} \cap C_\varepsilon$. Comme $\mathcal{L}$ est discret, cet ensemble non vide est fini et décroît avec ε. Il est donc constant pour ε assez petit. De plus, comme $C = \bigcap_\varepsilon C_\varepsilon$ est fermé, $\mathcal{L} \setminus \{0\} \cap C_\varepsilon$ est non vide. □

Le théorème de Minkowski donne une borne supérieure à la norme du plus court vecteur d'un réseau.

Théorème 1.3.10. Soit $\mathcal{L} \subseteq \mathbb{R}^n$ un réseau, alors $\lambda_1(\mathcal{L}) \leqslant \sqrt{n} \det(\mathcal{L})^{1/n}$.

Démonstration. Soit $B(0, r)$ une boule centrée en 0 et de rayon r en dimension n. Alors soit $[\frac{-r}{\sqrt{n}}, \frac{r}{\sqrt{n}}]^n \subseteq B(0, r)$ un hypercube. Donc, $vol(B(0, r)) > (\frac{2r}{\sqrt{n}})^n$. Posons $r = \sqrt{n} \cdot \det(\mathcal{L})^{1/n}$, alors $vol(B(0, r)) > 2^n \cdot \det(\mathcal{L})$. Par le théorème précédent, il existe un élément non nul de $\mathcal{L} \cap B(0, r)$. Un tel point est la boule. Ainsi, nous pouvons borner $\lambda_1(\mathcal{L})$:

$$\lambda_1(\mathcal{L}) \leqslant \sqrt{n} \cdot \det(\mathcal{L})^{1/n}.$$

□

1.4 Problèmes de réseaux difficiles

Définition 1.4.1. Soit $\gamma \geqslant 1$. Le problème approximate Shortest Vector Problem (γ-SVP) consiste : étant un réseau $\mathcal{L}$, trouver $s \in \mathcal{L}$ tel que $0 < \|s\| \leqslant \gamma \cdot \lambda_1(\mathcal{L})$.

Définition 1.4.2. Pour $\gamma \geqslant 1$, le problème du vecteur le plus proche approximé par γ (γ-CVP) est le suivant : étant donné une base B d'un réseau $\mathcal{L} = \mathcal{L}(B) \subseteq \mathbb{R}^n$ et un point $\mathbf{t} \in \mathbb{R}^n$, trouver un vecteur $\mathbf{v} \in \mathcal{L}$ tel que $\|\mathbf{t} - \mathbf{v}\| \leq \gamma \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L})$.

Définition 1.4.3. Soit $\gamma \geqslant 1$. Le problème approximate Shortest Independent Vectors problem (γ-SIVP) consiste : étant un réseau $\mathcal{L}$ de dimension n, trouver $s_1, \ldots, s_n \in \mathcal{L}$ linéairement indépendants tel que $\max_i \|s_i\| \leqslant \gamma \cdot \lambda_n(\mathcal{L})$.

Définition 1.4.4. Soit $\gamma \geqslant 1$. Le problème approximate Hermite Shortest Vector Problem (γ-HSVP) consiste : étant un réseau $\mathcal{L}$, trouver $s \in \mathcal{L}$ tel que $0 < \|s\| \leqslant \gamma \cdot \sqrt{n} \cdot \det(\mathcal{L})^{1/n}$

2 Algorithmes de réseaux

Lorsque nous nous intéressons à la cryptanaylyse des cryptosystèmes à base de réseaux, nous aurons besoin de regarder des algorithmes de réductions de réseaux.

2.1 Algorithme LLL

Comme l'algorithme de Lagrange-Gauss, l'algorithme LLL améliorera progressivement notre base B jusqu'à ce qu'elle possède certaines "bonnes" propriétés. Ensuite, il produira la base résultante. Pour l'algorithme LLL, il est pratique de définir d'abord la notion d'une base LLL-réduite avant de présenter l'algorithme qui trouve effectivement des bases LLL-réduites de manière efficace.

Tout d'abord, chaque base LLL-réduite sera réduite en taille au sens où $|\mu_{i,j}| \leq \frac{1}{2}$ pour tous les i, j. Il est facile de convertir n'importe quelle base en une base réduite en taille pour le même réseau en ajoutant ou soustrayant des multiples entiers de $b_1, \ldots, b_{i-1}$ à b_i jusqu'à ce qu'elle soit réduite en taille. (Il est important de le faire dans le bon ordre, en commençant par b_{i-1} et en terminant par b_1). C'est une propriété assez naturelle, et en particulier, elle généralise naturellement la notion de bases réduites en taille en deux dimensions.

Définition 2.1.1. Soit $\delta \in (\frac{1}{4}, 1)$. Soit B la base d'un réseau $\mathcal{L}$ de dimension n et R son R-facteur. La base B est dite δ-LLL-réduite si

1. $\forall i < j, |\mu_{i,j}| \leqslant \frac{1}{2}\mu_{i,i}$
2. $\forall i < n, \delta\mu_{i,i}^2 \leqslant \mu_{i+1,i+1}^2 + \mu_{i,i+1}^2$

La première condition est la réduction de taille de la base, on appelle la deuxième condition, condition de Lovász.

Lemme 2.1.2. Soit $\delta \in (\frac{1}{4}, 1)$ et $\alpha = \frac{1}{\sqrt{\delta - 1/4}}$. Soit $B = (b_1, \ldots, b_n)$ d'un réseau $\mathcal{L}$ de dimension n. Si B est δ-LLL-réduite, alors

$$\|b_1\| \leqslant \alpha^{n-1} \cdot \lambda_1(\mathcal{L})$$
$$\|b_1\| \leqslant \alpha^{\frac{n-1}{2}} \cdot \det(\mathcal{L})^{1/n}.$$

Définition 2.1.3. (Algorithme LLL)
Étant donné une base B de $\mathcal{L}$ et un $\delta \in (\frac{1}{4}, 1)$. L'algorithme LLL consiste en itérant les deux étapes suivantes :

— Réduction de taille : Pour tout j, pour tout i (ordre décroissant), soustraire $\lfloor \frac{\mu_{i,j}}{\mu_{i,i}} \rceil \cdot b_i$ de b_j ;

— Échanger : Pour tout i tel que $\delta \cdot \mu_{i,i}^2 > \mu_{i+1,i+1}^2 + \mu_{i,i+1}^2$, échanger b_i et b_{i+1}.

Algorithm 1 Algorithme LLL

procedure LLL($\{b_1, b_2, \ldots, b_n\}, \delta$)
  $B^* \leftarrow$ GramSchmidt($\{b_1, \ldots, b_n\}$) ▷ Ne pas normaliser
  for $i \leftarrow 1$ à n **do**
    for $j = i - 1$ à 1 **do**
      $\mu_{i,j} \leftarrow \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$ ▷ Utiliser les valeurs les plus récentes de b_i et b_j^*
      if $|\mu_{i,j}| > \frac{1}{2}$ **then**
        $b_i \leftarrow b_i - \lfloor \mu_{i,j} \rceil \cdot b_j$
        Mettre à jour B^* et les $\mu_{i,j}$ concernés si nécessaire
      end if
    end for
    if $\langle b_i^*, b_i^* \rangle > (\delta - \mu_{i+1,i}) \cdot \langle b_{i-1}^*, b_{i-1}^* \rangle$ **then**
      $i \leftarrow i + 1$
    else
      Échanger b_i et b_{i-1}
      Mettre à jour B^* et les $\mu_{i,j}$ concernés si nécessaire
      $i \leftarrow \max(i - 1, 2)$
    end if
  end for
  retourner B ▷ La base réduite LLL de $\{b_1, \ldots, b_n\}$
end procedure

Lemme 2.1.4. Le nombre de boucles dans l'algorithme LLL est borné supérieurement par $n^2 \cdot \max_i \log_{1/\delta} \|b_i\|$.

Définition 2.1.5. Une base $B = (b_1, \ldots, b_n)$ d'un réseau $\mathcal{L}$ est dite SVP-réduite si $\|b_1\| = \lambda_1(\mathcal{L})$.

Définition 2.1.6. Soit $B = (b_1, \ldots, b_n)$ une base pour un réseau $\mathcal{L}$. On appelle B_i un bloc 2×2 de la matrice B qui est

$$B_i := \begin{pmatrix} \|b_i^*\| & \mu_{i,i+1}\|b_i^*\| \\ 0 & \|b_{i+1}^*\| \end{pmatrix}.$$

Remarque 2.1.7. Une base δ-LLL-réduite possède la propriété que tous ses blocs B_i sont SVP-réduits.

Théorème 2.1.8. Si tous les blocs B_i de $B = (b_1, \ldots, b_n)$ d'un réseau $\mathcal{L}$ sont SVP-réduite alors $\|b_1\| \leqslant 2^n \lambda_1(\mathcal{L})$.

Corollaire 2.1.9. Il existe un algorithme en temps polynomial qui résout 2^n-SVP.

2.2 Algorithme de BKZ

L'acronyme BKZ signifie "Blockwise-Korkine-Zolotarev". L'algorithme a été introduit par Schnorr et Euchner en 1994. Cet algorithme se repose sur l'algorithme LLL pour avoir une réduction de réseau meilleur, ce qui en fait un des outils indispensables en cryptographie à base de réseaux.

Voici quelques notations utiles :

Définition 2.2.1. Soit $B = (b_1, \ldots, b_n)$ une base pour un réseau $\mathcal{L}$ et Soient $1 \leqslant i < j \leqslant n$. On note

$$B_{[i,j]} := \begin{bmatrix} \|b_i^*\| & \mu_{i,i+1}\|b_i^*\| & \ldots & \mu_{i,j-1}\|b_i^*\| & \mu_{i,j}\|b_i^*\| \\ 0 & \left\|b_{i+1}^*\right\| & \ldots & \mu_{i+1,j-1}\left\|b_{i+1}^*\right\| & \mu_{i+1,j}\left\|b_{i+1}^*\right\| \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & \left\|b_{j-1}^*\right\| & \mu_{j-1,j}\left\|b_{j-1}^*\right\| \\ 0 & 0 & \ldots & 0 & \left\|b_j^*\right\| \end{bmatrix}$$

On appelle $B_{[i,j]}$ un $k \times k$ bloc de B où $k = j - i + 1$ tel que $B_{[i,j]} \in \mathbb{R}^{k \times k}$. On définit aussi $B_{[i,j]} := B_{[i,n]}$ pour $j > n$. On dit que une base B est BKZ-réduite de taille de bloc k si tous ses $k \times k$ blocs sont SVP-réduits i.e. une base B est BKZ-réduite si les b_i^* ne peuvent pas plus courts en prenant une combinaison linéaire entière de $b_i, \ldots, b_{i+k-1}$.

Comment fonctionne BKZ. Afin de garantir que les $r_{i,i}$ ne diminuent pas trop rapidement, nous allons les réduire localement. Pour ce faire, nous travaillerons sur des blocs de réseau projetés. Plus précisément, pour une base B donnée, nous désignons par $B_{[i,j]}$ la base formée par les vecteurs de base $b_i, \ldots, b_j$ projetés orthogonalement sur les premiers vecteurs de base $1, \ldots, i-1$. C'est-à-dire que $B_{[i,j]}$ est une base pour le réseau donné par le sous-réseau formé par $b_1, b_2, \ldots, b_j$ projeté sur le sous-espace orthogonal des vecteurs $b_1, b_2, \ldots, b_{i-1}$. Notez que b_i^* est le premier vecteur de $B_{[i,j]}$.

Définition 2.2.2. On note par $\mathcal{L}_{[i,j]} := \mathcal{L}(B_{[i,j]})$ qu'on nomme le sous-réseau projeté.

Alternativement, nous pouvons examiner le sous-réseau projectif en utilisant le facteur R, qui est une matrice triangulaire supérieure. La projection d'un vecteur de base orthogonal aux vecteurs de base précédents correspond à supprimer les premières entrées du vecteur. Ainsi, considérer un bloc projeté $B_{[i,j]}$ revient à considérer la sous-matrice carrée de B composée des lignes et colonnes avec des indices entre i et j (inclus).

Pour contrôler les tailles des $r_{i,i}$, nous appelons un oracle SVP (énumération ou crible) sur le bloc $B_{[i,j]}$ de dimension $\beta = j - i + 1$. L'idée est que, une fois que cet oracle trouve le vecteur le plus court dans $B_{[i,j]}$, il l'insère dans la base B à la position i et supprime les dépendances linéaires potentielles en utilisant LLL. L'algorithme suivant formule cette idée.

Algorithm 2 Algorithme de BKZ

```
1: procedure BKZ(B : une base LLL-réduite, β : paramètre de bloc, oracle SVP)
2:     repeat
3:         for i ← 1 to n do
4:             exécuter LLL sur B_[i,min{i+β−1,n}]
5:             v ← le plus court vecteur de B_[i,min{i+β−1,n}]
6:             insérer v dans B
7:         end for
8:     until La base est suffisamment bonne
9: end procedure
```

Heuristique 2.2.3. Soit $\mathcal{L}$ un réseau de dimension n avec un volume $\det(\mathcal{L})$. On attend que la norme du plus court vecteur de $\mathcal{L}$ sous l'heuristique gaussienne soit :

$$gh(\mathcal{L}) := \frac{\det(\mathcal{L})^{1/n}}{vol(B_1)^{1/n}} \cong \sqrt{\frac{n}{2\pi e}} \cdot \det(\mathcal{L})^{1/n}$$

où B_1 est la boule unité en dimension n. On note par $gh(n) = \sqrt{n/(2\pi e)}$ si $\det(\mathcal{L}) = 1$

Remarque 2.2.4. L'algorithme commence par appeler l'oracle SVP sur le premier bloc $B_{[1,\beta]}$. Une fois que ce bloc est réduit par SVP, nous passons au bloc suivant $B_{[2,\beta+1]}$ et appelons l'oracle sur celui-ci. Remarquez que la réduction SVP de $B_{[2,\beta+1]}$, en plus de l'insertion du nouveau vecteur et de l'application de LLL, peut modifier le réseau projeté $B_{[1,\beta]}$ et b_1 peut ne plus être le vecteur le plus court dans le premier bloc, c'est-à-dire qu'il peut potentiellement être réduit davantage. Nous traitons cela plus tard et continuons de cette manière jusqu'à atteindre la fin de la base, c'est-à-dire jusqu'à ce que nous ayons appelé l'oracle sur $B_{[n-\beta+1,n]}$. Une fois que nous avons atteint la fin de la base, c'est-à-dire que nous avons terminé avec $B_{[n-\beta+1,n]}$, nous commençons à réduire la taille de la fenêtre, c'est-à-dire que nous appelons l'oracle sur $B_{[n-\beta,n]}$, $B_{[n-\beta+1,n]}$, etc., et nous nous arrêtons après avoir terminé avec $B_{[n-1,n]}$. Tout ce processus est appelé une tournée BKZ. Une fois que nous avons terminé une tournée, nous revenons en arrière et corrigeons les blocs qui ne sont plus réduits par SVP (la boucle `Repeat-Until` dans l'algorithme 4). Encore une fois, si la deuxième tournée a modifié la base, il n'y a aucune garantie que tous les blocs sont réduits par SVP, et nous passons à la troisième tournée. En principe, ce processus peut-être non borné, mais [HPS11] montre que terminer BKZ après un nombre polynomial de tournées n'impactera pas la qualité de la base de sortie.

3 Réseaux structurés à base d'idéaux et modules

3.1 Rappels de théorie des nombres

3.1.1 Les corps de nombres et leurs plongements

Définition 3.1.1. Un corps de nombres K est un sous-corps de $\mathbb{C}$ qui est de dimension finie comme un $\mathbb{Q}$-espace-vectoriel, i.e., $\dim_{\mathbb{Q}} K < \infty$. Nous allons noter par d la dimension $[K : \mathbb{Q}]$.

Définition 3.1.2. Soit K un corps de nombres. On dit que $\alpha \in K$ est un entier s'il existe $P \in \mathbb{Z}[X]$ unitaire tel que $P(\alpha) = 0$.

Claim 3.1.3. Soit $P \in \mathbb{Z}[X]$ ayant une factorisation $P = QR$ dans $\mathbb{Q}[X]$, alors $Q, R \in \mathbb{Z}[X]$.

Proposition 3.1.4. (Théorème de l'élément primitif)
Tout corps de nombres K est monogène (i.e. engendrée par un seul élément). C'est-à-dire de la forme $K = \mathbb{Q}(\alpha)$ pour $\alpha \in \mathcal{O}_K$.

Lemme 3.1.5. Soit $K = \mathbb{Q}(\alpha)$ un corps de nombres de degré d sur $\mathbb{Q}$. Nous avons un isomorphisme canonique : $K = \mathbb{Q}(\alpha) \simeq \mathbb{Q}[X]/(P_{\alpha,\mathbb{Q}})$.

Définition-Proposition 3.1.6. Soit $\mathcal{O}_K$ l'ensemble des éléments entiers de K. Alors,

1) $\mathcal{O}_K$ est un sous-anneau de K.

2) $\mathrm{Frac}(\mathcal{O}_K) = K$.

3) $\mathcal{O}_K$ est un $\mathbb{Z}$-module de type fini.

Définition 3.1.7. Soit K/k une extension de corps, on note $\Sigma_{K/k}$ l'ensemble de morphismes de K dans $\mathbb{C}$ qui fixe k, i.e. qui sont k-linéaires.

Notation 3.1.8. On note par $\Sigma_{L/K}$ l'ensemble des plongements de L dans $\mathbb{C}$ qui fixe K.

Lemme 3.1.9. Soit $K = \mathbb{Q}(\alpha)$ un corps de nombres de degré d sur $\mathbb{Q}$. Alors nous avons exactement d plongements distincts $\sigma_i = K \to \mathbb{C}$. Plus précisément l'application suivante est une bijection :

$$\begin{aligned} \{\Sigma_{K/\mathbb{Q}}\} &\to \{\text{les } d \text{ racines de } P_{\alpha,\mathbb{Q}} \text{ dans } \mathbb{C}\} \\ \sigma &\to \sigma(\alpha) \end{aligned}$$

Remarque 3.1.10. Si $\sigma : K \to \mathbb{C}$ est un plongement de K alors $\overline{\sigma} : x \mapsto \overline{\sigma(x)}$ provenant de la conjugaison complexe $y \mapsto \overline{y} \in \mathbb{C}$ est aussi un plongement de K.

Définition-Proposition 3.1.11. Si $\overline{\sigma} = \sigma$, on dit que σ est un plongement réel de K. Si $\overline{\sigma} \neq \sigma$ et on dit que $(\sigma, \overline{\sigma})$ est une paire de plongements complexes. La signature (r, s) du corps K est le nombre r de ses plongements réels, et celui s de la moitié du nombre de ses plongements complexes. En utilisant le lemme précédent $[K : \mathbb{Q}] = r + 2s$.

Pour mieux déterminer la norme d'un élément de K. Nous avons donc deux plongements (par coefficients et canonique) définis

Définition 3.1.12. Soit $K/\mathbb{Q}$ un corps de nombres, soit α un élément primitif de K. Il existe exactement d plongements distincts i.e. morphismes de corps de K dans $\mathbb{C}$. On définit l'i-ème plongement $\sigma_i : K \to \mathbb{C}, \alpha \mapsto \alpha_i$. On peut alors classifier les plongements en deux catégories :

1. $\sigma_1, \ldots, \sigma_r$ pour les plongements réels,
2. $\sigma_{r+1}, \ldots, \sigma_{r+2s}$ conjugués deux à deux i.e. $\sigma_{r+i} = \overline{\sigma_{r+s+i}}$ pour $i \in [s]$.

on définit

1. (plongement par coefficients)

$$\begin{aligned} \Sigma : K &\to \mathbb{Q}^d \\ \sum_{i=0}^{d-1} a_i X^i &\mapsto (a_0, \ldots, a_{d-1}) \end{aligned}$$

2. (plongement canonique)

$$\begin{aligned} \tau : K &\to \mathbb{R}^r \times \mathbb{C}^{2s} \\ x &\mapsto (\sigma_1(x), \ldots, \sigma_d(x)) \end{aligned}$$

Définition 3.1.13. Soit $d = r + 2s$ un entier (on verra le lien avec le degré d'un corps de nombres K). Puisqu'on travaille avec les corps de nombres et les réseaux ideaux, on introduit l'espace $H \subseteq \mathbb{R}^r \times \mathbb{C}^{2s} \subseteq \mathbb{C}^d$ défini de la façon suivante

$$H := \{x = (x_1, \ldots, x_{r+2s}) \in \mathbb{R}^r \times \mathbb{C}^{2s} | x_{r+1} = \overline{x_{r+s+1}}, \ldots, x_{r+s} = \overline{x_{r+2s}}\} \subseteq \mathbb{C}^n.$$

Remarque 3.1.14. On voit que l'espace H est inclus dans l'image de K sous le plongement canonique τ. Ce morphisme donne un isomorphisme d'anneaux $K_{\mathbb{R}} := K \otimes_{\mathbb{Q}} \mathbb{R} \cong H$. Il n'est pas si difficile de voir que H muni du produit scalaire venant de $\mathbb{C}^n$ est isomorphe à $\mathbb{R}^n$ comme un espace préhilbertien, cela se voit par la définition suivante.

Le K espace-vectoriel $K_{\mathbb{R}} := K \otimes_{\mathbb{Q}} \mathbb{R}$ est une $\mathbb{R}$-algèbre et tout plongement σ_i définit un morphisme de $\mathbb{R}$-algèbres de $K_{\mathbb{R}}$ dans $\mathbb{C}$ via l'application $\sigma_i(\alpha \otimes r) = r\sigma_i(\alpha)$. On obtient alors un isomorphisme de $\mathbb{R}$-algèbres :

$$\begin{aligned} \tau : K_{\mathbb{R}} &\to H \subseteq \mathbb{R}^r \times \mathbb{C}^{2s} \\ x &\mapsto (\sigma_1(x), \ldots, \sigma_d(x)) \end{aligned}$$

Cetta application τ nous laisse définir une forme bilinéaire symétrique sur $K_{\mathbb{R}}$ qui est définie positive et munit $K_{\mathbb{R}}$ d'une structure hermitienne donnée par :

$$\langle a, b\rangle_\sigma = \sum_i^n \overline{\sigma_i(a)}\sigma_i(b).$$

La norme correspondante s'appelle la norme canonique. Nous pouvons rendre le plongement canonique isométrique, en munissant l'espace $\mathbb{R}^r \times \mathbb{C}^{2s}$ du produit :

$$\langle a, b\rangle_\sigma = \sum_{i=1}^r a_i b_i + 2\sum_{i=r+1}^s \Re(a_i\overline{b_i}).$$

3.1.2 Extension aux espaces vectoriels

Dans ce rapport, on suppose connaître une $\mathbb{Z}$-base $(r_i)_{i\leqslant d}$ de $\mathcal{O}_K$ qui est LLL-réduite. On définit aussi $\delta_K = \max_i \|r_i\|_\infty$, qui peut valoir 1 si nous travaillons avec des corps de nombres cyclotomiques de puissance de 2.

Pour $x \in K_{\mathbb{R}}$, on définit $\overline{x} \in K_{\mathbb{R}}$ l'élément $\overline{x} = \tau^{-1}(\tau(x))$, autrement dit, $\overline{x}$ est l'élément obtenu en prenant la conjugaison complexe de chaque coordonnée du plongement canonique de x. Nous étendons cette notation sur les vecteurs et les matrices sur $K_{\mathbb{R}}$ et notons $\mathbf{x}^\dagger := \overline{\mathbf{x}}^T$ pour tout $\mathbf{x} = (x_1, \ldots, x_n) \in K_{\mathbb{R}}^n$.

Soit n un entier non-nul et soit $K_{\mathbb{R}}^n = \oplus_{i=1}^n K_{\mathbb{R}}$. Ceci est clairement un espace de dimension nd sur $\mathbb{R}$. Chaque facteur de cet espace est un espace euclidien avec la norme canonique, on munit $K_{\mathbb{R}}^n$ d'une structure euclidienne suivante :

$$\langle \mathbf{x}, \mathbf{y}\rangle_{K_{\mathbb{R}}^n} = \mathbf{x}^\dagger \cdot \mathbf{y} = \sum_{i=1}^n \langle x_i, y_i\rangle_\sigma \in \mathbb{R} \text{ et } \|\mathbf{x}\| = \left\|\langle \mathbf{x}, \mathbf{x}\rangle_{K_{\mathbb{R}}^n}\right\|^{1/2}$$

pour tout $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n) \in K_{\mathbb{R}}^n$.

Pour $x = \sum_i x_i r_i \in K_{\mathbb{R}}$, on définit $\lfloor x \rceil = \sum_i \lfloor x_i \rceil r_i$. On utilise la notation $\{x\} = x - \lfloor x \rceil$.

3.1.3 Trace, norme

Définition 3.1.15. La norme algébrique de $x \in K_{\mathbb{R}}$ est définie de la façon suivante :

$$N(x) = \prod_i \sigma_i(x).$$

Définition 3.1.16. Soit L/K une extension de corps de nombres, et $x \in L$. Alors on définit les quantités (trace et norme relatives) de x.

1. $Tr_{L/K}(x) := \sum_{\sigma\in\Sigma_{L/K}} \sigma(x)$
2. $N_{L/K}(x) := \prod_{\sigma\in\Sigma_{L/K}} \sigma(x)$

Proposition 3.1.17. Soit L/K une extension de corps de nombres, alors $N_{L/K}$ satisfait

1. $N_{L/K}(xy) = N_{L/K}(x)N_{L/K}(y)$ pour $x, y \in L$.
2. $N_{L/K}(a) = a^{[L:K]}$ si $a \in K$.
3. Si $L/M/K$ une tour d'extensions de corps, alors $N_{L/K} = N_{M/K} \circ N_{L/M}$.

Proposition 3.1.18. Soit L/K une extension de corps de nombres, alors $Tr_{L/K}$ satisfait

1. $Tr_{L/K}(x + y) = Tr_{L/K}(x) + Tr_{L/K}(y)$ pour $x, y \in L$.
2. $Tr_{L/K}(a) = a[L : K]$ si $a \in K$.
3. Si $L/M/K$ un tour d'extensions de corps, alors $Tr_{L/K} = Tr_{M/K} \circ Tr_{L/M}$.

Théorème 3.1.19. (Dedekind)
Soit K un corps de nombres de degré $d = [K : \mathbb{Q}]$, et un idéal non-nul I de $\mathcal{O}_K$. Alors I admet une $\mathbb{Z}$-base à d-éléments. En particulier, il existe $\omega_1, \ldots, \omega_d$ tels que

$$\mathcal{O}_K = \mathbb{Z}\omega_1 + \cdots + \mathbb{Z}\omega_d$$

Autrement dit, I et $\mathcal{O}_K$ sont isomorphes à $\mathbb{Z}^d$ comme des $\mathbb{Z}$-module, i.e. comme groupes abéliens.

3.1.4 Idéaux

Nous ne travaillons qu'avec des anneaux commutatifs R.

Définition 3.1.20. Soit R un anneau. Un idéal de R est un sous-ensemble $I \subseteq R$ qui est un sous-groupe de $(R, +)$ et tel que $R \cdot I \subseteq I$ i.e. tel que $a' - b' \subseteq I, \forall a', b' \in I$ et $aa' \in I, \forall a \in R, a' \in I$.

Définition 3.1.21. Soit K un corps de nombres de degré d. Un idéal fractionnaire (resp. idéal replet orienté) de K est un sous-ensemble de K de la forme $x \cdot I := \{x \cdot a | a \in I\}$ avec $x \in K^\times$ (resp. $x \in K_{\mathbb{R}}^\times$) et I un idéal de $\mathcal{O}_K$.

Définition 3.1.22. Soient I, J deux idéaux fractionnaires. On définit trois opérations sur I et J

$$\begin{aligned} I + J &:= \{x + y : x \in I, y \in J\} \\ I \cdot J &:= \{\sum_{i=1}^{r} x_i \cdot y_j : r > 0, x_i \in I, y_i \in J\} \\ I^{-1} &:= \{x \in K, xI \subseteq \mathcal{O}_K\} \end{aligned}$$

Proposition 3.1.23. Soient I, J deux idéaux fractionnaires $I + J, I \cdot J$ et I^{-1} sont des idéaux fractionnaires.

Définition-Proposition 3.1.24. Soit $\mathfrak{p} \subseteq \mathcal{O}_K$ alors les propositions suivantes sont équivalentes

1. $\mathcal{O}_K/\mathfrak{p}$ est un anneau intègre.
2. $\forall a, b \in \mathcal{O}_K, ab \in \mathfrak{p} \Rightarrow a \in \mathfrak{p}$ *ou* $b \in \mathfrak{p}$.

On dit qu'un idéal qui vérifie les propriétés $1, 2$ est premier.

Théorème 3.1.25. Tout idéal non-nul $I \subseteq \mathcal{O}_K$ se factorise de manière unique à l'ordre près, sous la forme

$$I = \mathfrak{p}_1^{k_1} \cdots \mathfrak{p}_r^{k_r}$$

en un produit des puissances d'idéaux premiers distincts $\mathfrak{p}_j$ de $\mathcal{O}_K$.

Remarque 3.1.26. Les entiers k_j sont les valuations en $\mathfrak{p}_j$ de l'idéal I.

Lemme 3.1.27. Soient $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ des idéaux fractionnaires d'un corps de nombres K avec $\mathfrak{c}$ non-nul.

1. si $\mathfrak{ac} = \mathfrak{bc}$ alors $\mathfrak{a} = \mathfrak{b}$.
2. $\mathfrak{c}|\mathfrak{b} \iff \mathfrak{b} \subseteq \mathfrak{c}$

Définition 3.1.28. La norme d'un idéal non-nul $I \subseteq \mathcal{O}_K$ est l'indice de I dans $\mathcal{O}_K$:

$$N(I) := |\mathcal{O}_K/I|$$

Cet indice est fini d'après 3.1.19 et aussi du théorème de structure des groupes abéliens de type fini.

Définition 3.1.29. Nous définissons aussi la norme pour un idéal replet orienté de la forme $x \cdot I$. Alors

$$N(x \cdot I) := N(x) \cdot N(I)$$

Proposition 3.1.30. Soient $\mathfrak{p}$ un idéal premier, I, J des idéaux fractionnaires, $\mathfrak{a}$ un idéal, alors

1. Le cas où l'idéal I est principal $I = \alpha\mathcal{O}_K = (\alpha)$, alors $N(I) := |\mathcal{O}_K/I| = |\mathcal{O}_K/\alpha\mathcal{O}_K| = |N_{K/\mathbb{Q}}(\alpha)|$.
2. $N(I \cdot J) = N(I)N(J)$.
3. $N(\mathfrak{p}) = p^k$ pour un $k \geqslant 1$ et p un nombre premier.
4. $N(\mathfrak{a}) \in \mathfrak{a}$.

3.1.5 Modules

Les $\mathbb{Z}$-modules sont très faciles à classifier, tandis que les $\mathcal{O}_K$-modules sont plus compliqués puisqu'il peut arriver que $\mathcal{O}_K$ ne soit pas principal, par exemple avec $K = \mathbb{Q}(\sqrt{-47})$. Pour un $\mathcal{O}_K$-module $M \subseteq K^r$, nous avons une sorte de notion similaire à une base qui est une pseudo-base. Pour nous, tout $\mathcal{O}_K$-module M est de type fini et sans torsion.

Lemme 3.1.31. Soit M un module de rang r inclus dans K^m. Il existe r vecteurs $b_1, \ldots, b_r \in K^m$ qui sont K-linéairement indépendants, et r idéaux fractionnaires $I_1, \ldots, I_r$ tels que

$$M = \sum_{i=1}^{r} I_i \cdot b_i := \left\{ \sum_i x_i b_i \mid x_i \in I_i \text{ pour tout } i \right\}.$$

La liste des paires $(b_i, I_i)_{1 \leq i \leq r} =: (B, \mathbb{I})$ est appelée une pseudo-base du module M.

Lorsque tous les I_i sont $\mathcal{O}_K$ dans la pseudo-base de M, M est dit libre.

3.2 Sur les réseaux idéaux et modules

Définition 3.2.1. Soit $j \in \{1, \ldots, n\}$ et $e_j \in \mathbb{C}^n$ des vecteurs canoniques. On définit $h_j := e_j, \forall j \in \{1, \ldots, r\}$ et pour $r < j \leqslant r+s$, $h_j = \frac{1}{\sqrt{2}}(e_j + e_{j+s})$ et $h_{j+s} = \frac{\sqrt{-1}}{\sqrt{2}}(e_j - e_{j+s})$.

On muni H de la norme l_p induit par $\mathbb{C}^n$. Soit $a_1, \ldots, a_n \in \mathbb{R}, \sum a_i h_i \in H$, alors sa norme p est

$$\left\| \sum_{i=0}^{n} a_i h_i \right\|_p = \left(\sum_{i=1}^{r} |a_i|^p + 2 \sum_{i=r+1}^{r+s} \left(\frac{a_i^2 + a_{i+s}^2}{2}\right)^{\frac{p}{2}} \right)^{\frac{1}{p}}$$

Remarque 3.2.2. Nous notons que, pour tout $p \in [1, \infty]$, cette norme est égale à un facteur près de $\sqrt{2}$ à $(\sum_{i=1}^{n} |a_i|)^{1/p}$, qui est la norme l_p induite sur H à partir de l'isomorphisme avec $\mathbb{R}^n$ décrit ci-dessus ; pour la norme l_2, nous avons en fait une égalité. Cette quasi-équivalence entre H et $\mathbb{R}^n$ nous permettra d'utiliser les définitions et résultats connus sur les réseaux dans notre contexte, la seule petite mise en garde étant le facteur $\sqrt{2}$ lorsqu'on traite les normes l_p pour $p \neq 2$.

Proposition 3.2.3. Soit $K = \mathbb{Q}[X]/(X^d + 1)$ avec d une puissance de 2. Alors pour tout $x \in K$, $\|\tau(x)\| = \sqrt{d}\|\Sigma(x)\|$.

Démonstration. Voir théorème 3.1. de [Cha20]. □

Le fait que l'on travaille avec le plongement canonique n'est pas anodin. En raison du rôle central du plongement canonique dans l'étude des corps de nombres et de nombreuses propriétés géométriques utiles, nous soutenons que le plongement canonique est la notion "correcte" à utiliser dans l'étude des réseaux idéaux.

Lemme 3.2.4. Soit $x \in K$, alors $\|\tau(x)\| \geqslant \sqrt{d}|N(x)|^{1/d}$. En particulier, $\forall x \in \mathcal{O}_K$ non-nul, on a $\|\tau(x)\| \geqslant \sqrt{d}$.

Démonstration. On applique l'inégalité arithmético-géométrique sur le vecteur $(|\sigma_1(x)|^2, \ldots, |\sigma_d(x)|^2)$, qui nous donne :

$$\begin{aligned}(\prod_i |\sigma_i(x)|^2)^{1/d} &= ((\prod_i |\sigma_i(x)|)^2)^{1/d} \\ &= |N(x)|^{2/d} \\ &\leqslant 1/d \cdot \sum_i |\sigma_i(x)|^2 \\ &= 1/d \cdot \|\tau(x)\|^2\end{aligned}$$

en prenant la racine carrée des deux côtés, on obtient l'inégalité qu'on a énoncée. Puisque la norme d'un élément $x \in \mathcal{O}_K$ non-nul est $\geqslant 1$, d'où $\|\tau(x)\| \geqslant \sqrt{d}$. □

Puisque la multiplication des éléments $x, y \in K$ plongés dans H par τ est une multiplication composante par composante (comme τ est un morphisme d'anneaux), alors on a

Lemme 3.2.5. Soient $x, y \in K$, on a

$$\|\tau(x \cdot y)\| \leqslant \|\tau(x)\|_\infty \cdot \|\tau(y)\| \leqslant \|\tau(x)\| \cdot \|\tau(y)\|.$$

Proposition 3.2.6. Soit $M \subseteq K^m$ un module de rang r. Alors les ensembles

$$\Sigma(M) := \{\Sigma(b) | b \in M\} \subseteq \mathbb{Q}^{md} \text{ et } \tau(M) := \{\tau(b) | b \in M\} \subseteq \mathbb{C}^{md}$$

sont des réseaux de rang dr. On les appelle des **réseaux modules**. Si $m = r = 1$ alors M est un idéal fractionnaire. Alors on les appelle des **réseaux idéaux**.

On rappelle qu'un idéal fractionnaire I possède une $\mathbb{Z}$ -base $U = \{u_1, \ldots, u_n\}$, ainsi sous les deux plongements Σ et τ. Pour notre facilité, on identifie un idéal avec son réseau donné par le plongement canonique. On va donc noter $\lambda_1(I)$ au lieu de $\lambda_1(\tau(I))$.
Le discriminant absolu Δ_K d'un corps de nombres K est défini comme le carré du volume de domaine fondamental de $\tau(\mathcal{O}_K)$ i.e. $\Delta_K = \det(\tau(\mathcal{O}_K))^2$. De façon équivalente, $\Delta_K = |\det(Tr(b_i \cdot b_j))|$ où $\{b_1, \ldots, b_n\}$ est $\mathbb{Z}$-base de $\mathcal{O}_K$.

Exemple 3.2.7. Soit $K = \mathbb{Q}[X]/(X^2+1)$, $\mathcal{O}_K = \mathbb{Z}[i]$ de base $\{1, i\}$. Ainsi $\Delta_{K/\mathbb{Q}} = \det(\sigma_i(b_j))^2$ i.e. $\left(\det \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}\right)^2 = 4$.

On rappelle le discriminant d'un polynôme.

Définition 3.2.8. Soit K un corps et $P = a_d X^d + \cdots + a_0 \in K[X]$ de degré d, de racines complexes α_i. Le discriminant de P est

$$\Delta(P) := a_n^{2n-2} \prod_{i<j} (\alpha_i - \alpha_j)^2$$

Exemple 3.2.9. Soit $K = \mathbb{Q}[X]/(X^d + 1)$, $\Delta_K = \Delta(X^d + 1) = d^d$.

Corollaire 3.2.10. Le volume fondamental de n'importe quel réseau idéal $\tau(I)$ est $N(I) \cdot \sqrt{\Delta_K}$.

On peut bien sûr généraliser le corollaire à un module $M \subseteq K^r$ de rang r et de pseudo-base $((b_i, I_i))_i$, on a $\det(\tau(M)) = \Delta_K^{r/2} \cdot |N(det_K(B))| \cdot \prod_i N(I_i)$, où B est la matrice où les colonnes sont b_i.

Remarque 3.2.11. On définit ici aussi une formule analogue pour la norme algébrique d'un module qui n'a pas forcément le même rang que K^m (i.e. les modules inclus dans K^m avec $m > r$).
Soit $M' \in K^m$ de rang $r < m$ d'une pseudo-base $((b_i, I_i))_{i<m}$, alors on définit :

$$N(M') = N(\det{}_{K_{\mathbb{R}}} \sqrt{B^* \cdot B}) \prod_i N(I_i)$$

Les bornes supérieure et inférieure de la distance minimale d'un réseau idéal sont données par le lemme classique suivant. La borne supérieure découle directement du premier théorème de Minkowski, tandis que la borne inférieure provient de l'inégalité entre la moyenne arithmétique et géométrique, ainsi que du fait que $|N(x)| \geqslant N(I), \forall x \in I$ non-nul.

Exemple 3.2.12. Soient $K = \mathbb{Q}[X]/(X^2+1)$, $\mathcal{O}_K = \mathbb{Z}[i]$ et $I = (1+X)$. Le réseau idéal $\tau(I)$ est engendré par $\tau(1+X)$ et $\tau((1+X)\cdot X)$. une base de $\tau(I)$ est $\begin{pmatrix} 1+X & -1-X \\ 1-X & -1-X \end{pmatrix}$ et $\det(I) = 4$. Voir la figure 5

Lemme 3.2.13. Pour tout idéal fractionnaire $I \subseteq K$ un corps de nombres de degré d, on a

$$\sqrt{d} \cdot N(I)^{1/d} \leqslant \lambda_1(I) \leqslant \sqrt{d} \cdot N(I)^{1/d} \cdot \sqrt{\Delta_K^{1/d}}$$
$$\lambda_d(\tau_K(I)) \leqslant \Delta_K^{1/d} \cdot \lambda_1(\tau_K(I))$$

Comme les réseaux idéaux et modules sont en particulier des réseaux, alors les problèmes de réseaux comme SVP ou CVP que nous avons définis avant sont utilisables dans le cadre des réseaux idéaux et modules.

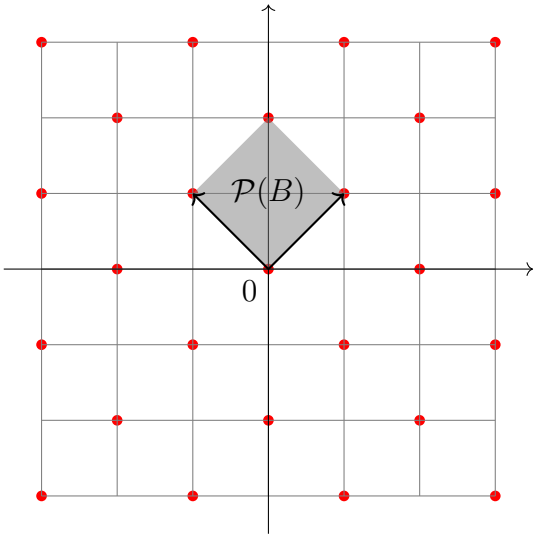

FIGURE 5 – Un réseau idéal dans $\mathbb{Z}[i]$

Définition 3.2.14. Soit K un corps de nombres, $r \geqslant 1$ un entier et $\gamma \geqslant 1$ réel. On définit le problème γ-r-module-SVP le problème de vecteur court avec facteur d'approximation γ restreint aux réseaux de type $\tau(M)$ où $M \subseteq K^r$ est un module de rang r. Si $r = 1$, γ-r-module-SVP devient γ-ideal-SVP.

3.3 Sous-corps et théorie de Galois

La théorie de Galois est un outil très puissant que nous allons utiliser pour attaquer des problèmes sur les réseaux structurés. Faisons alors un petit rappel.

Définition 3.3.1. Soit K un corps de nombres. Une extension de corps de nombres L de K est un corps de nombres L muni d'un morphisme de corps $K \to L$. Dans ce cas, L est un K espace-vectoriel.

Remarque 3.3.2. Si L/K est une extension de corps de nombres et $x \in K$ alors on écrit $\tau_K(x) \in \mathbb{C}^{[K:\mathbb{Q}]}$ pour le plongement canonique de x vu comme un élément de K et $\tau_L(x) \in \mathbb{C}^{[L:\mathbb{Q}]}$ pour le plongement canonique de x vu comme un élément de L. Ces deux plongements sont bien évidemment reliés par le lemme suivant.

Lemme 3.3.3. Soit L/K est une extension de corps de nombres. Tout plongement complexe de K s'étend en un plongement complexe de L et cela couvre tous les plongements complexes de L i.e. l'application $\Sigma_{L/\mathbb{Q}} \to \Sigma_{K/\mathbb{Q}}, \sigma \mapsto \sigma_{|K}$ est surjective, chaque élément de $\Sigma_{K/\mathbb{Q}}$ admet exactement $[L : K]$ antécédents. En particulier, $|\Sigma_{L/K}| = [L : K]$. Comme conséquence, pour tout $x \in K$, on a $\|\tau_L(x)\| = \sqrt{[L : K]} \cdot \|\tau_K(x)\|$.

Lemme 3.3.4. Soit L/K est une extension de corps de nombres. Si I est un idéal fractionnaire de L alors $I \cap K$ est un idéal fractionnaire de K.

Définition 3.3.5. Soit K un corps de nombres. Un automorphisme de K est un morphisme de corps $K \to K$ tel que les $\mathbb{Q}$-points sont fixes. Et on note par $Aut(K)$ l'ensemble des automorphismes de K et on le munit d'une structure de groupe avec $\phi_1 \circ \phi_2 \in Aut(K), \forall \phi_1, \phi_2 \in Aut(K)$.

Définition-Proposition 3.3.6. Soit L/K est une extension de corps de nombres. On a équivalence entre :

1. L est Galoisienne sur K ;
2. $\forall \alpha \in L$, on a $P_\alpha = \prod_{\beta \in Aut(L/K)\cdot\alpha} (X - \beta)$;
3. $|Aut(L/K)| = [L : K]$;
4. $L^{Aut(L/K)} = K$ i.e. les points fixes dans L pour l'action de $Aut(L/K)$ sont exactement les points de K.

Exemple 3.3.7. L'extension n-cyclotomique d'un corps K est le corps de décomposition K_n du polynôme $X^n - 1$ c'est-à-dire l'extension engendrée par les racines n-ièmes de l'unité. Comme on traille sur caractéristique $char(\mathbb{Q}) = 0$ alors $X^n - 1$ est séparable donc l'extension K_n/K est Galoisienne et $Gal(K_n/K) \cong (\mathbb{Z}/n\mathbb{Z})^\times$ d'où $|Gal(K_n/K)| = |(\mathbb{Z}/n\mathbb{Z})^\times| = \varphi(n)$ avec φ l'indicatrice d'Euler.

En applicant un automorphisme de K à un $x \in K$ ne change pas sa norme euclidienne avec le plongement canonique.

Lemme 3.3.8. Soit $x \in K$ et $\varphi \in Aut(K)$, alors $\|\tau_K(x)\| = \|\tau_K(\varphi(x))\|$.

Théorème 3.3.9. (Théorème fondamental de la théorie de Galois)
Soit K un corps de nombres Galoisienne sur $\mathbb{Q}$. Alors il existe une bijection entre les sous groupes de $G = Gal(K/\mathbb{Q})$ et les sous corps de K. Voici la correspondance :

$$\{\text{sous-extensions de } K\} \leftrightarrow \{\text{sous-groupes de } G\}$$
$$\Phi : K' \to Gal(K/K')$$
$$K^{G'} \leftarrow G' : \Psi$$

Ces deux applications sont des bijections réciproques, qui échangent sous-extensions Galoisiennes et sous-groupes distingués.

4 NTRU module. Cryptanaylyse de NTRU

Dans cette partie, nous allons regarder plusieurs attaques, sur les instances particulières de l'idéal-SVP et une autre sur NTRU quand le module q est très large et la norme

de trappe $\frac{\sqrt{q}}{\gamma}$ très petite. Les attaques se servent des structures algébriques des problèmes ou directement des algorithmes de réduction de réseaux.

4.1 NTRU

Soit K un corps de nombres de degré d. Soit $R := \mathcal{O}_K$ l'anneau des entiers de K et on note $R_q := R/qR$ pour q un entier non-nul. En particulier, si $R = \mathbb{Z}[\alpha]$, on a un isomorphisme canonique.

$$R/qR \cong (\mathbb{Z}/q\mathbb{Z})[X]/(P_{\alpha,\mathbb{Q}})$$

Si on a $P(X) = X^d + 1$ alors $R_q = (\mathbb{Z}/q\mathbb{Z})[X]/(X^d + 1)$ où d est une puissance de 2.

Le problème NTRU est une hypothèse computationnelle importante en cryptographie à clé publique. Cependant, du point de vue de la réduction, sa difficulté relative par rapport à d'autres problèmes sur les réseaux euclidiens n'est pas bien comprise. Sa version de décision se réduit au problème de recherche Ring-LWE, mais cela ne fournit qu'une borne supérieure de difficulté. Des réponses ont été apportés dans [PMS21] consistant à fournir des preuves de la difficulté du problème NTRU basées sur des réductions.

Définition 4.1.1. (Instances de NTRU)
Soit $q > 1$ un entier et $\gamma > 1$ réel. Une instance (q, γ)-NTRU est un élément $h \in R$ tel que il existe une paire $(f, g) \in R^2 \setminus (0, 0)$ avec $\|\tau(f)\|, \|\tau(g)\| \leqslant \sqrt{q}/\gamma$ satisfaisant $g \cdot h = f \bmod q$. On appelle la paire $(f, g) \in R^2$ une trappe de h.

Remarque 4.1.2. Il existe en fait deux régimes de NTRU dans la littérature, le premier régime est le cas où f et g sont de la forme $\sqrt{q} \cdot \mathsf{poly}(d)$ et l'autre est le cas où f et g sont de la forme $\sqrt{q}/\mathsf{poly}(d)$. Nous considérons principalement le deuxième cas dans ce rapport de mémoire.

Lemme 4.1.3. Soit q un nombre premier tel que $q \nmid \Delta_K$, alors la proportion d'éléments de R qui sont des instances NTRU est $\leqslant (\frac{2^5}{\gamma^2})^d$.

Définition 4.1.4 ($\mathrm{NTRU}_{\mathrm{vec}}$)**.** Soit $q > 1$ un entier, $\gamma > 1$ un réel, et soit ψ une loi de probabilité sur les instances $\mathbf{NTRU} \subseteq R$. Le problème de (q, γ, ψ)-$\mathrm{NTRU}_{\mathrm{vec}}$ consiste en : étant donné $h \leftarrow \psi$, récupérer $(f, g) \in R^2 \setminus \{(0, 0)\}$ tel que $g \cdot h = f \bmod q$ et $\|\tau(f)\|, \|\tau(g)\| \leq \sqrt{q}/\gamma$.

Lemme 4.1.5. [PMS21, Lemme 3.5] Soit $q > 1$ un entier, $\gamma > \sqrt{2}$ un réel, et h une instance (q, γ)-NTRU. Alors pour toutes trappes $(f, g), (f', g') \in R \setminus \{(0, 0)\}$ avec $\|\tau(f)\|, \|\tau(g)\|, \|\tau(f')\|, \|\tau(g')\| \leq \sqrt{q}/\gamma$ et $g \cdot h = f \mod q$, $g' \cdot h = f' \mod q$, il existe

$x \in K$ tel que $(f, g) = x \cdot (f', g')$. De façon équivalente, il existe un unique $h_K \in K$ tel que pour toute trappe $(f, g) \in R^2 \setminus \{(0, 0)\}$ avec $\|\tau(f)\|, \|\tau(g)\| \leq \sqrt{q}/\gamma$ et $g \cdot h = f \mod q$, nous avons une division dans K qui donne $f/g = h_K$.

Définition 4.1.6 (NTRU_{mod})**.** Soit $q > 1$ un entier, $\gamma > \sqrt{2}$ un réel. Le problème de (q, γ)-NTRU_{mod} consiste en : étant donné h une instance (q, γ)-NTRU, récupérer l'unique élément h_K associé à h défini dans le lemme précédent.

Remarque 4.1.7. $\tau(M_h)$ est un réseau possédant un vecteur exceptionnellement court par 2.2.3, l'existence d'un vecteur court de $\tau(M_h)$ implique l'existence de d vecteurs courts dans $\tau(M_h)$. Par exemple, si $K = \mathbb{Q}(\zeta_n)$ et $\mathcal{O}_K$ son anneau des entiers, alors toute rotation $(\zeta_n^i \cdot f, \zeta_n^i \cdot g)$ possède la même norme euclidienne. Donc si nous avons un vecteur exceptionnellement court dans $\tau(M_h)$, nous avons ce qu'on appelle un sous-réseau dense de dimension d qui correspond à un sous-module de rang 1 de M_h. En récupérant h_K, nous récupérons en fait le sous-module libre de rang 1 de M_h engendré par (f, g). En utilisant un des plongements que nous avions introduits auparavant, nous obtenons un sous-réseau dense de $\tau(M_h)$, par exemple.

Définition 4.1.8. (dec-NTRU)
Soit $q > 1$ un entier et $\gamma > 1$ réel et soit ψ une loi de probabilité sur les instances NTRU $\subseteq R$. Le problème de (q, γ, ψ)-dec-NTRU consiste en : distinguer les échantillons venant de ψ et les échantillons venant de $U(R_q)$.

NTRU en tant que problème de réseau modules. Nous allons maintenant voir que le problème NTRU peut-être considéré comme un cas particulier du problème **module-SVP** dans des modules de rang 2. Ci-dessous, nous définissons une notion de module NTRU. Remarquons que les modules NTRU sont définis pour tout $h \in \mathbb{R}$, même pour ceux qui ne sont pas des instances NTRU.

Définition 4.1.9. (Réseaux modules NTRU)
Soit $q > 1$ un entier et $h \in R$. Le module NTRU M_h associé à h est un R-module libre de rang 2 engendré par les colonnes de la matrice $B_h := \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \in R^{2\times 2}$.

Lemme 4.1.10. (NTRU_{vec} en module-SVP)
Soit $q > 1$ un entier et $\gamma > 1$ réel. Soit $h \in R_q$ une instance (q, γ)-NTRU. Il existe alors une réduction en temps polynomial de $(q, \gamma/2)$-NTRU avec entrée h en module-SVP avec entrée M_h le module NTRU.

Démonstration. Soit $(f, g) \in R^2$ une solution du problème NTRU_{vec} avec entrée h i.e. $g \cdot h = f \mod q$ et $\|f\|, \|g\| \leqslant \sqrt{q}/\gamma$. Le vecteur $v = \begin{pmatrix} g \\ f \end{pmatrix} = (g, f)^T$ appartient à M_h.

C'est très facile à voir puisque, $\begin{pmatrix} g \\ f \end{pmatrix} = g \cdot \begin{pmatrix} 1 \\ h \end{pmatrix} + r \begin{pmatrix} 0 \\ q \end{pmatrix}$ Par la propriété de la norme $\|v\| \leqslant \|g\| + \|f\| \leqslant 2\sqrt{q}/\gamma$ puisque v est non-nul. Ainsi $\lambda_1(M_h) \leqslant 2\sqrt{q}/\gamma$.

Commençons maintenant la réduction. Soit $w := (g', f')$ une solution de module-SVP pour le R-module M_h i.e. w non-nul et $\|w\| \leqslant \lambda_1(M_h)$. Or comme $\lambda_1(M_h) \leqslant 2\sqrt{q}/\gamma$, en particulier, $\|f'\|, \|g'\| \leqslant 2\sqrt{q}/\gamma$ séparémment. Montrons maintenant que (f', g') est une solution pour $(q, \gamma/2)$-NTRU, il reste juste montrer que $g'h = f' \bmod q$. Comme $\begin{pmatrix} g' \\ f' \end{pmatrix} \in M_h$ alors il existe $a_1, a_2 \in R$ tels que $\begin{pmatrix} g' \\ f' \end{pmatrix} = a_1 \cdot \begin{pmatrix} 1 \\ h \end{pmatrix} + a_2 \begin{pmatrix} 0 \\ q \end{pmatrix} = \begin{pmatrix} a_1 \\ a_1 h + a_2 q \end{pmatrix}$. En identifiant $g' = a_1$ et $f' = a_1 h + a_2 q = a_1 h \bmod q$, on finit la preuve. □

4.2 Vecteurs courts dans certains idéaux

Dans cette partie, nous allons voir comment construire un algorithme qui utilise des sous-corps pour retrouver des vecteurs courts dans certains idéaux particuliers.

Définition 4.2.1. Soit I un idéal fractionnaire de K un corps de nombres. On appelle le groupe de décomposition de I le groupe des automorphismes de K qui fixe I. C'est-à-dire :

$$H := \{\varphi \in Aut(K) | \varphi(I) = I\}$$

où $\varphi(I) = \{\varphi(x) | x \in I\}$.

L'intuition derrière l'algorithme que nous allons présenter est la suivante : Soit K un corps de nombres galoisien, $x \in K$, $H < \mathrm{Gal}(K/\mathbb{Q})$. Si pour tout $\varphi \in H$, on a $\varphi(x) = x$, alors $x \in K^H$. De même, pour un idéal I, si $\varphi(I) = I$ pour tout $\varphi \in H$, alors on peut en quelque sorte voir I comme un idéal de K^H. L'avantage de l'algorithme repose sur le fait que K^H est un corps de nombres de dimension inférieure à la dimension de K, ce qui permet d'obtenir des algorithmes de réduction de réseaux plus efficaces. Nous allons donner une version simplifiée de l'algorithme présentée dans [BGPM22], car nous cherchons davantage à comprendre le principe. En conséquence, les vecteurs seront un peu plus longs.

Théorème 4.2.2. Soit K un corps de nombres Galoisien et $I \subseteq K$ un idéal fractionnaire avec H son groupe de décomposition. Soient $\gamma \geqslant 1$ et $d = [K : \mathbb{Q}]$. Il existe un algorithme qui prend en entrée I, fait un appel à un oracle γ-SVP sur un réseau de dimension $d/|H|$ et résout $\gamma' id$-SVP pour I avec $\gamma' = |H| \cdot \Delta_K^{1/d} \cdot \gamma$. L'algorithme tourne en temps polynomial sauf quand on appelle l'oracle SVP.

Voici l'algorithme mentionné dans le théorème 4.2.2. L'idée de l'algorithme est de calculer K^H puis regarder l'idéal $J = I \cap K^H$ qui est un idéal d'un corps de nombres de degré plus petit puis on utilise l'oracle SVP sur J et par définition, un élément court de J est aussi un élément de I. Le point essentiel de la preuve sera de montrer que c'est aussi un élément court de I où on utilisera le fait que I est stabilisé par H.

Algorithm 3 Trouver des vecteurs courts des idéaux

1: Calculer le groupe de décomposition H de I
2: Calculer K^H
3: Calculer $J = I \cap K^H$
4: $v \leftarrow \gamma$-SVP$(\tau_{K^H}(J))$
5: Calculer $s \in J$ tel que $\tau_{K^H}(s) = v$
6: **return** s

Démonstration. Montrons d'abord que l'algorithme 3 satisfait l'énoncé du théorème. Tout d'abord, tous les calculs, à l'exception de l'oracle SVP, s'effectuent en temps polynomial. Expliquons, par exemple, l'algorithme pour obtenir H : nous commençons avec un ensemble vide H, calculons une base de $\tau(I)$, puis, pour tout automorphisme $\tau \in \mathrm{Aut}(K/\mathbb{Q})$, si $\tau(I) = I$, nous mettons à jour H en ajoutant τ à celui-ci.

Observons que $[K_H : \mathbb{Q}] = d/|H|$ alors le réseau $\tau_{K^H}(J)$ est de dimension $d/|H|$ et l'oracle est appelé sur un réseau de dimension appropriée. Par construction $J = I \cap K^H$, alors si $s \in J$ non-nul alors s est un vecteur non-nul de I. Il reste juste alors de trouver une borne supérieure à $\|\tau_K(s)\|$.

Soit $l_1, \ldots, l_d \in I$ $\mathbb{Z}$-linéairement indépendantes et tel que $\|\tau_K(l_i)\| \leqslant \lambda_d(\tau_K(I))$, pour tout i dans $[d]$.

Soit maintenant $i \in [d]$. Par la définition de H, on a que $\forall \varphi \in H, \varphi(I) = I$. I est stable par addition donc

$$Tr_{K/K^H}(l_i) = \sum_{\varphi \in H} \varphi(l_i) \in I$$

et puisque $Tr_{K/K^H}(x) \in K^H, \forall x \in K$. Ainsi,

$$Tr_{K/K^H}(l_i) \in I \cap K^H = J$$

Grâce à cette information, nous pourrons borner supérieurement $\lambda_1(\tau_{K^H}(J))$. Supposons que pour tout $i \in [d]$ nous avons que $Tr_{K/K^H}(l_i) = 0$. Alors par la propriété de trace 3.1.18 nous avons que $Tr_{K/\mathbb{Q}}(l_i) = 0, \forall i$. Mais puisque les $l_1, \ldots, l_d \in I$ sont $\mathbb{Z}$-linéairement indépendantes de K alors ils forment une $\mathbb{Q}$-base de K. Or on sait que l'application trace est K linéaire donc $Tr_{K/\mathbb{Q}}(x) = 0, \forall x \in K$, ce qui est une contradiction puisque

l'application trace est non-nulle. Nous pouvons alors supposer qu'il existe $i \in [d]$ tel que $Tr_{K/K^H}(l_i) \neq 0$ et on note i_0 cette indice. Alors on a

$$\begin{aligned}
\lambda_1(\tau_{K^H}(J)) &\leqslant \left\|\tau_{K^H}(Tr_{K/K^H}(l_{i_0}))\right\| = \sqrt{|H|}^{-1} \cdot \left\|\tau_K(Tr_{K/K^H}(l_{i_0}))\right\| \\
&= \sqrt{|H|}^{-1} \cdot \left\|\tau_K(\sum_{\varphi \in H} \varphi(l_{i_0}))\right\| \\
&\leqslant \sqrt{|H|}^{-1} \cdot \sum_{\varphi \in H} \|\tau_K(\varphi(l_{i_0}))\| \\
&= \sqrt{|H|}^{-1} \cdot |H| \cdot \|\tau_K(l_{i_0})\| \\
&\leqslant \sqrt{|H|} \cdot \lambda_d(\tau_K(I)).
\end{aligned}$$

La première égalité vient du lemme 3.3.3, l'égalité à la quatrième ligne vient du lemme 3.3.8. En se rappelant de $\lambda_d(\tau_K(I)) \leqslant \Delta_K^{1/d} \cdot \lambda_1(\tau_K(I))$ de 3.2.13, cela nous donne finalement

$$\lambda_1(\tau_{K^H}(J)) \leqslant \sqrt{|H|} \cdot \Delta_K^{1/d} \cdot \lambda_1(\tau_K(I)).$$

Cela nous dit que le plus court vecteur de J n'est pas beaucoup plus large que le plus court vecteur de I. Nous allons alors utiliser un oracle SVP sur $\tau_{K^H}(J)$ pour obtenir un $v \in \tau_{K^H}(J)$ tel que $\|v\| \leqslant \gamma\lambda_1(\tau_{K^H}(J))$, puis encore avec le lemme 3.3.3, on a

$$\|\tau_K(s)\| = \sqrt{|H|} \cdot \|\tau_{K^H}(s)\| = \sqrt{|H|} \cdot \|v\| \leqslant |H| \cdot \Delta_K^{1/d} \cdot \gamma \cdot \lambda_1(\tau_K(I)).$$

Nous avons alors trouvé un vecteur court approximé s de I, ce qui termine notre démonstration. □

4.3 Attaque avec les sous-corps sur dec-NTRU lorsque q est grand

Dans la suite, nous allons regarder une attaque qui consiste à transformer une instance-NTRU h d'un sous-corps K en une autre instance-NTRU h qui appartient à un sous-corps $K' \subseteq K$. En se ramenant à une dimension plus petite, on espère que le problème est plus facile à résoudre. L'idée ici est de prendre $h' = N_{K/K'}(h)$. Ainsi, si (f, g) est une trappe de h, alors $(N_{K/K'}(f), N_{K/K'}(g))$ est une trappe pour h'. La norme de f' et g' est contrôlable en fonction de la norme de f et g mais aussi du degré de K' en tant que $\mathbb{Q}$ espace-vectoriel. En choisissant un sous-corps adapté, il est possible de nous assurer que $h' \in K'$ est aussi une instance NTRU et on pourra alors résoudre le problème dans K'.

L'attaque que nous allons montrer ne résout qu'une version décisionnelle du problème de NTRU, c'est-à-dire que distinguer ou pas si l'instance h provient d'une distribution uniforme.

Notre objectif sera alors : étant donné un $h' = N_{K/K'}(h)$, de distinguer si h est une instance NTRU tiré uniformément de R_q. Si nous choisissons bien nos paramètres q et γ, alors h' est une instance NTRU si h l'est aussi avec $\gamma' \leqslant \gamma$. Nous espérons que si $h \leftarrow U(R_q)$ alors $h' \leftarrow U(R_q)$, ou au moins que h' ne devienne pas une instance NTRU si h ne l'était pas, comme nous avons vu que les instances NTRU sont plutôt rares du moment que $\sqrt{q}/\gamma'$ est petit en fonction de $\sqrt{q}$. Cependant, nous ne savons pas comment le prouver, nous allons donc le supposer.

Heuristique 4.3.1. Soit K/K' une extension de corps de nombres avec $[K : \mathbb{Q}] = d$ et $[K' : \mathbb{Q}] = d'$. Soit $q > 1$ un entier et $\gamma' \geqslant 8$ (donc $\sqrt{q}/\gamma' \leqslant \sqrt{q}/8$). On suppose heuristiquement que

$$\mathbb{P}_{h \leftarrow U(\mathcal{O}_K/(q\mathcal{O}_K))}(N_{K/K'}(h) \text{ est une } (q,\gamma')\text{-NTRU instance dans } K') \leqslant 2^{-d'}$$

Par le lemme 4.1.3, on sait que quand q ne divise pas $\Delta_{K'}$ alors la proportion d'instances (q,γ')-NTRU dans K' est inférieur à $(\frac{2^5 \cdot \sqrt{q}/\gamma'^2}{q})^{d'} = (\frac{2^5}{\gamma'^2})^{d'} \leqslant 2^{d'}$ comme $\sqrt{q}/\gamma' \leqslant \sqrt{q}/8$.

Ainsi si $N_{K/K'}(h)$ est tiré uniformément de $\mathcal{O}_{K'}/q\mathcal{O}_{K'}$ alors notre heuristique est probablement vraie. Notre supposition sera alors de supposer que $N_{K/K'}(h)$ se comporte comme un élément distribué uniformément dans $\mathcal{O}_{K'}/q\mathcal{O}_{K'}$.

Principe de l'algorithme : Il transforme une instance NTRU en une nouvelle instance NTRU dans un sous-corps de dimension plus petite et résout le problème dec-NTRU dans le sous corps.

Théorème 4.3.2. Soit $K = \mathbb{Q}[X]/(X^d + 1)$ avec d une puissance de 2. Soient $q > 1$ un entier et $\gamma > 16 \cdot q^{1/4}$ un réel. Soit ψ une loi sur les instances (q,γ)-NTRU. Soit $m := 2^{\lceil C \rceil}$ où $C = \log_2\left(\frac{d\left(\frac{1}{2}\log(q) - \log(\gamma)\right)}{\frac{1}{4}\log(q) - 4}\right)$. Sous l'heuristique 4.3.1, l'algorithme 4 résout (q,γ,ψ)-dec-NTRU avec une probabilité $\geq 1 - 2^{-m}$. Comme on peut le voir, l'algorithme 4 fait un seul appel à l'oracle γ-SVP dans un réseau de dimension $2m$.

Remarque 4.3.3. L'algorithme 4 s'exécute en temps polynomial sauf quand on appelle l'oracle γ-SVP.

Algorithm 4 Résoudre dec-NTRU dans un corps cyclotomique puissance de 2

procedure RES($\gamma \in 16 \cdot q^{1/4}$: gap , q : module, h : une instance (q,γ)-NTRU dans $K = \mathbb{Q}[X]/(X^d+1)$)
  Soit m la plus petite puissance de 2 tel que $\geqslant \frac{d\left(\frac{1}{2}\log(q)-\log(\gamma)\right)}{\frac{1}{4}\log(q)-\log(16)}$
  Soit $K' = \mathbb{Q}[X]/(X^m+1)$ ▷ Un sous-corps de K
  Calculer $h' = N_{K/K'}(h) \in K'$
  Calculer $M_{h'} \in \mathcal{O}_{K'}^{2\times 2}$ le NTRU-module associé à h'
  Obtenir $s \leftarrow \eta$-SVP($\tau_{K'}(M_{h'})$) avec $\eta = q^{1/4}$
  if $\|s\| \leqslant \sqrt{q}/8$ **then**
    return "h est une instance NTRU"
  else
    return "Instance aléatoire"
  end if
end procedure

Exemple 4.3.4.
1. Nous remarquons si γ est fixe et $q = 2^{\sqrt{d}}$. Alors m est environ $\sqrt{d}$ et $\eta = 2^{\sqrt{d}/4}$ est 2^m à constante près. En se rappelant de l'algorithme LLL qui résout SVP si le facteur d'approximation est environ 2^m. Dans ce cas, l'algorithme que nous présentons est en temps polynomial.
2. Si par contre $q = \sqrt{d}$ et γ constante, alors $m \approx d$ et $\eta = \mathsf{poly}(d)$. Alors on a besoin de résoudre $\eta-$SVP en dimension d avec un facteur d'approximation $\mathsf{poly}(d)$. Nous ne connaissons que des algorithmes en temps exponentiel en d pour résoudre ces instances SVP. Alors, il est préférable d'appeler directement un algorithme de réduction de réseau sur $\tau(M_h)$.

Démonstration. Notre but est de distinguer si un élément h est une instance NTRU ou tiré de façon uniforme de R_q.

Soit $q > 1$ un entier et $\gamma > 16 \cdot q^{1/4}$. Nous pouvoir voir que $m \leqslant d$ et c'est une puissance de 2 par définition. Alors K possède bien un sous-corps cyclotomique de degré m sur $\mathbb{Q}$.

Soit h une (q,γ)-NTRU instance et (f,g) une trappe pour h, i.e., $g \cdot h = f \mod q$ et $\|\tau_K(f)\|, \|\tau_K(g)\| \leqslant \sqrt{q}/\gamma$. Posons $H := Gal(K/K')$ et nous savons que $|H| = [K : K'] = d/m$. Prenons un automorphisme $\varphi \in H$. Alors $\varphi(g) \cdot \varphi(h) = \varphi(f) \mod q$, qui n'est pas trivial parce qu'il faut voir que $\varphi(q) = q$. C'est le cas, parce que φ fixe K' et donc $\mathbb{Q}$ aussi. On a que $g \cdot h = f + q \cdot r$, donc $\varphi(g) \cdot \varphi(h) = \varphi(f) + \varphi(q) \cdot \varphi(r) = \varphi(f) + q \cdot \varphi(r) = \varphi(f) \mod q$. En utilisant la norme relative ou trace relative, on peut descendre au sous-corps

et obtenir ainsi l'équation suivante :

$$N_{K/K'}(g) \cdot N_{K/K'}(h) = N_{K/K'}(f) \mod q.$$

Posons $g' = N_{K/K'}(g), f' = N_{K/K'}(f)$ et $h' = N_{K/K'}(h)$. Sachant qu'en descendant au sous-corps, la norme des éléments augment, nous avons $\|\tau_{K'}(g')\| \leqslant \|\tau_K(g')\| \leqslant \|\tau_K(g)\|^{|H|} \leqslant (\sqrt{q}/\gamma)^{|H|}$. De même pour $\|\tau_{K'}(f')\|$. Posons $\gamma' = \sqrt{q} \cdot \frac{\gamma^{|H|}}{\sqrt{q}^{|H|}}$. Alors h' est une instance (q, γ')-NTRU dans K'. Reste à vérifier que pourquoi (f', g') n'est pas $(0, 0)$. C'est tout simplement parce-que définition de la norme relative. Si $f' = N_{K/K'}(f) = 0$ alors cela veut dire qu'il existe un automorphisme dans H qui envoie f sur 0, dans ce cas $f = 0$ qui est une contradiction.

Sachant que $m \geqslant \frac{d\left(\frac{1}{2}\log(q) - \log(\gamma)\right)}{\frac{1}{4}\log(q) - \log(16)}$, nous avons

$$\sqrt{q}/\gamma' = (\sqrt{q}/\gamma)^{d/m} \leqslant (\sqrt{q}/\gamma)^A = \sqrt{q}^{1/4}/16$$

où $A = \frac{\left(\frac{1}{2}\log(q) - \log(\gamma)\right)}{\frac{1}{4}\log(q) - \log(16)}$.

Le $\mathcal{O}_{K'}$-module de rang 2 $M_{h'}$ contient un vecteur court non nul (g', f'), ainsi $\lambda_1(M_{h'}) \leqslant 2 \cdot \sqrt{q}/\gamma' \leqslant q^{1/4}/8$. Posons $\eta = q^{1/4}$, soit v une solution de η-SVP dans $\tau_{K'}(M_{h'})$. Nous avons alors

$$\|v\| \leqslant \eta \cdot \lambda_1(M_{h'}) \leqslant \sqrt{q}/8.$$

Donc lorsque notre h de départ est une instance NTRU, nous obtenons à la fin de l'algorithme un vecteur v tel que $\|v\| \leqslant \sqrt{q}/8$ et par 4.3.1, nous obtenons un affichage "h est une instance NTRU". □

Dans la suite, nous utiliserons des résultats de l'article [DvW21] pour voir comment l'algorithme BKZ résout le problème NTRU_{mod}. Mais il faut faire très attention parce que nous ne sommes plus dans le même régime qu'avant. Dans cette sous-section, la trappe (f, g) est de la forme $\sqrt{q} \cdot \text{poly}(d)$ alors qu'avant nous étions dans le régime où (f, g) est de la forme $\sqrt{q}/\text{poly}(d)$.

4.4 Comment l'algorithme de BKZ récupère le sous module dense du module NTRU dans les paramètres de Overstreched ?

Remarque 4.4.1. Le but de cette section est de présenter des résultats montrant qu'on peut attaquer le problème de NTRU en utilisant des algorithmes de réduction. L'idée est d'utiliser une estimation de la forme de notre réseau ainsi qu'un lemme important qui généralise le lemme 4.4.5. Lorsqu'on appelle un algorithme de réduction de réseau (BKZ ici), cela va modifier ce qu'on appelle le profil du réseau, jusqu'à contredire le lemme généralisé de Pataki et Tural.

On sait que BKZ avec une taille de bloc β assez grand trouve le sous-module dense. Par l'article [DvW21] de Léo Ducas et Wessel van Woerden nous laisse comprendre que l'algorithme BKZ pour une taille de bloc plus peteit ce qu'on pensait au début, récupère le sous-module dense du module NTRU M_h ou de façon équivalente de récupère le sous réseau dense $\mathcal{L}^{GF}$ qui est le sous réseau envoyé par le plongement canonique appliqué au sous-module $(g, f)^T\mathcal{O}_K$ où (g, f) est la clé secrète telle que $g \cdot h = f \bmod q$.

On avait déja dit que pour un réseau $\mathcal{L} = \mathcal{L}(B)$ avec B une base, il existait un invariant qui est le volume du réseau $\det(\mathcal{L}) = \det(B) = \prod_{i=1}^{n} \|b_i^*\|$ avec $b_1^*, \ldots, b_n^*$ les vecteurs de Gram-Schmidt de la matrice B. Donc si on diminue la norme du premier vecteur $b_1 = b_1^*$, la taille des autres vecteurs de Gram-Schmidt vont augmenter. On appelle ces longueurs $(\|b_i^*\|)_{i=1,\ldots,n}$ le profile de la base B qui engendre le réseau $\mathcal{L}$. On dit qu'une base a un bon profile si le profile ne décroît pas trop vite.

On rappelle que une base $B = (b_1|\ldots|b_n)$ est β-BKZ réduite si

$$\|b_\kappa^*\| = \lambda_1(\mathcal{L}_{[\kappa:\min(\kappa+\beta,n)]}),\ \textit{pour tout}\ \kappa \in [n].$$

On reprend la même notation que dans [DvW21] pour la récupération du sous-réseau dense du réseau NTRU.

Définition 4.4.2 (Événement DSD)**.** Pour une exécution de BKZ sur un réseau NTRU $\mathcal{L}$ avec un sous-réseau dense $\mathcal{L}^{GF}$, nous définissons **Découverte du sous-réseau dense** (DSD) : La première fois qu'un vecteur dense du réseau $\mathbf{v} \in \mathcal{L}^{GF}$ est inséré.

Si on applique l'heuristique gaussienne 2.2.3 au réseau $\mathcal{L}_{[\kappa:\min(\kappa+\beta,n)]}$ où $b_\kappa = \lambda_1(\mathcal{L}_{[\kappa:\min(\kappa+\beta,n)]})$ en chaque position κ. Alors 2.2.3 associe chaque valeur $\|b_1^*\|, \ldots, \|b_n^*\|$ entre elles. Pour un $\beta \ll n$, on a $\frac{\|b_\kappa^*\|}{b_{\kappa+1}^*} \cong C_\beta$, où C_β est une constante qui dépend de β. Qu'est ce que cela nous dit ? Le profile forme en fait heuristiquement une série géométrique.

Heuristique 4.4.3. (Hypothèse de la série géométrique (HSG))
Soit B une base β-BKZ réduite, alors son profile satisfait

$$\ln \|b_i^*\| = \frac{n-2i}{2} \cdot \ln C_\beta + \frac{\ln \det(B)}{n}$$

où $C_\beta = gh(\beta)^{2/(\beta-1)}$.

Rappelons qu'un réseau $\mathcal{L}$ est un réseau q-aire pour un entier $q \geqslant 2$ si $q\mathbb{Z}^n \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$. Notre réseau $\Sigma(M_h)$ est clairement un réseau q-aire. Plus précisément c'est une matrice par blocs. Par exemple : soit $K = \mathbb{Q}(\zeta_8) \cong \mathbb{Q}[X]/(X^4+1)$, le réseau $\Sigma(M_h)$ avec $h \in \mathcal{O}_K$ est de dimension 8 et $h = -5\zeta_8^3 - 4\zeta_8^2 + 5\zeta_8 + 3$.

```
sage: K.<zeta> = CyclotomicField(8)
sage: OK = K.ring_of_integers()
sage: h = -5*zeta^3 - 4*zeta^2 + 5*zeta + 3
sage: H = h.matrix(); q = 11
sage: block_matrix([[q, H],[0, 1]])
[11  0  0  0  3  5 -4 -5]
[ 0 11  0  0  5  3  5 -4]
[ 0  0 11  0  4  5  3  5]
[ 0  0  0 11 -5  4  5  3]
[ 0  0  0  0  1  0  0  0]
[ 0  0  0  0  0  1  0  0]
[ 0  0  0  0  0  0  1  0]
[ 0  0  0  0  0  0  0  1]
```

Listing 1 – Exécution SageMath qui donne une base pour réseau NTRU

Avant de faire les réductions de réseaux sur $\tau(M_h)$, étudions le profile de notre exemple :

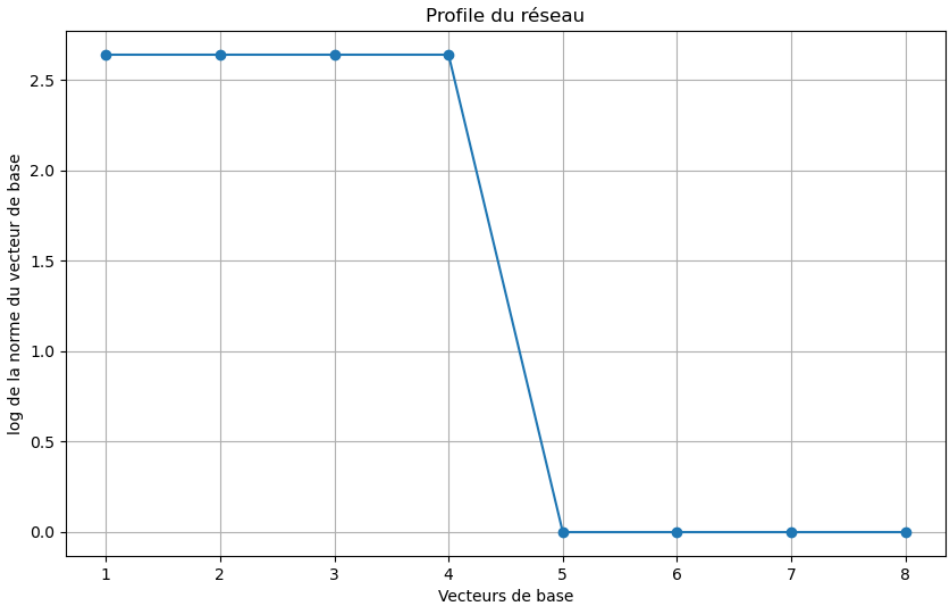

FIGURE 6 – Profile projeté de $\Sigma(M_h)$ avec $h = -5\zeta_8^3 - 4\zeta_8^2 + 5\zeta_8 + 3$ et $K = \mathbb{Q}(\zeta_8)$ et $q = 11$

.

Après la réduction LLL sur un réseau q-aire, le profil doit être plat au début et à la fin, et avoir une partie inclinée au milieu, nous appelons cela une forme en Z.

Heuristique 4.4.4. (Hypothèse de la série géométrique en forme de Z(HSGZ))
Soit B une base HNF pour un réseau q-aire $\mathcal{L}$ de dimension $2d$ avec d q-vecteurs. Après

une réduction β-BKZ, le profile de $\mathcal{L}$ sera :

$$\|b_i^*\| := \begin{cases} q & \text{si } i \leqslant d-m+1, \\ \sqrt{q} \cdot C_\beta^{\frac{2d-1-2i}{2}} & \text{si } d-m+1 < i < d+m, \\ 1 & \text{si } i \geqslant d+m, \end{cases}$$

où $C_\beta = gh(\beta)^{2/(\beta-1)}$ et $m = \frac{1}{2} + \frac{\ln q}{\ln C_\beta}$.

Lemme 4.4.5. (Pataki et Tural)
Soit $\mathcal{L}$ un réseau de dimension n, et $b_1, \ldots, b_n$ une base quelconque de $\mathcal{L}$. Soit $k \leq n$ un entier positif. Alors, pour tout sous-réseau $\mathcal{L}' \subseteq \mathcal{L}$ de dimension k , on a :

$$\det(\mathcal{L}') \geqslant \min_J \prod_{j \in J} \|b_j^*\|,$$

où J parcourt tous les sous-ensembles de $\{1, \ldots, n\}$ de taille k.

Proposition 4.4.6. (DSD – Estimation de Kirchner-Fouque)
Soit $\mathcal{L}^{H,q}$ un réseau NTRU de dimension $2d$, avec un sous-réseau dense $\mathcal{L}^{GF} \subseteq \mathcal{L}^{H,q}$. Sous l'hypothèse de la série géométrique en forme de Z, BKZ-β déclenche l'événement DSD si

$$\det(\mathcal{L}^{GF}) < q^{\frac{m}{2}-\frac{1}{2}} \cdot \alpha_\beta^{-\frac{1}{2}(m-1)^2},$$

où $\alpha_\beta = \mathrm{gh}(\beta)^{2/(\beta-1)}$, et $m = \frac{1}{2} + \frac{\ln(q)}{2\ln(\alpha_\beta)}$.

Rappelons la définition de sous-réseau projeté 2.2.2. Nous notons le sous-réseau intersecté par $\mathcal{L}^{GF}_{\cap[1:r]} := \mathcal{L}^{H,q}_{[1:r]} \cap \mathcal{L}^{GF}$.

Lemme 4.4.7. (Généralisation de Pataki et Tural)
Soit $\mathcal{L}$ un réseau de dimension n avec une base $b_1, \ldots, b_n$, et considérons le sous-réseau $\mathcal{L}_{[1:s]}$. Pour tout sous-réseau de dimension d $\mathcal{L}' \subseteq \mathcal{L}$, nous avons

$$\det(\mathcal{L}_{[1:s]} \cap \mathcal{L}') \leqslant \det(\mathcal{L}') \cdot \left(\min_J \prod_{j \in J} \|b_j^*\| \right)^{-1},$$

où $k := \dim(\mathcal{L}_{[1:s]} \cap \mathcal{L}')$ et J parcourt les sous-ensembles de taille $(d-k)$ de $\{s, \ldots, n\}$.

Corollaire 4.4.8. Soit $\mathcal{L}^{H,q}$ un réseau NTRU avec un sous-réseau dense $\mathcal{L}^{GF}$ de dimension n, si $\dim(\mathcal{L}^{GF}_{\cap[1:d+k]}) = k$ pour un certain $k \geq 0$, alors

$$\det(\mathcal{L}^{GF}_{\cap[1:d+k]}) \leqslant \det(\mathcal{L}^{GF}) \cdot \left(\prod_{j=d+k}^{n} \|b_j^*\| \right)^{-1}.$$

Proposition 4.4.9. [DvW21, Claim 3.5]
L'algorithme BKZ avec la taille de bloc $\beta = Bd$ appliqué à un réseau NTRU $\tau(M_h)$ avec paramètres $q = \Theta(d^Q)$, $\|(f, g)\| = O(d^S)$ retrouve le sous-réseau dense de $\tau(M_h)$ si

$$B = \frac{8S}{Q^2 + 1} + o(1).$$

Dans ce cas, l'algorithme BKZ retrouve le sous-réseau dense de $\tau(M_h)$ avec une complexité $2^{O(\beta)}$.

5 La réduction de module unique SVP vers NTRU

Dans cette section, nous allons voir une réduction du problème de module Unique-SVP pour les $\mathcal{O}_K$-module de rang 2 vers NTRU. Définissons les différentes variantes de ce problème.

À part les minimas successives $\lambda_i(\tau(M))$, nous serons également intéressés par le module de norme minimale $\lambda_1^N(M)$.

Définition 5.0.1. (Module le plus dense)
$\lambda_1^N(M) = \inf\{N(N)|N \subseteq M$ sous-module de rang 1$\}$. Un sous-module de rang 1 de M est dit le plus dense s'il atteint $\lambda_1^N(M)$.

Définition 5.0.2. (Instance γ-uSVP)
Soit $\gamma > 0$. Une instance γ-uSVP consiste en une pseudo-base $(B, \mathbb{I}) = ((b_i, I_i))_i$ d'un $\mathcal{O}_K$-module $M \subseteq K^2$ de rang 2 tel que M contient un vecteur s avec $\|s\| \leqslant \frac{N(M)^{1/(2d)}}{\gamma}$. Où $N(M) = N(det_K(B)) \cdot \prod_i N(I_i)$.

Lemme 5.0.3. [FPMS22, Lemme 2.1] Il existe un algorithme en temps polynomial qui prend en entrée une base $B = (b_1, \ldots, b_n)$ d'un réseau $\mathcal{L}$ de dimension n, un paramètre $s \geqslant \sqrt{n} \cdot \max_i \|b_i\|$ et un centre $c \in \mathrm{Vect}_{\mathbb{R}}(\mathcal{L})$ et qui produit un échantillon à partir d'une distribution $\hat{D}_{B,s,c}$ telle que

— $dist(D_{L,s,c}, \hat{D}_{B,s,c}) \leq 2^{-\Omega(n)}$;
— pour tout $\mathbf{v} \leftarrow \hat{D}_{B,s,c}$, il est vrai que $\|\mathbf{v} - c\| \leqslant \sqrt{n} \cdot s$.

Lemme 5.0.4. [FPMS22, Lemme 2.6] Pour tout module M, il existe un sous-module N de M de rang tel que $N(N) = \lambda_1^N(M)$.

Dans ce mémoire, on s'intéresse aux modules de rang 2 qui contiennent un sous-module dense de rang 1 i.e. des modules $M \subseteq K^2$ avec $\lambda_1^N(M) \ll \sqrt{N(M)}$. Nous définissons le gap de la façon suivante

$$\gamma(M) = \left(\frac{N(M)^{1/2}}{\lambda_1^N(M)}\right)^{1/d}.$$

Le lemme suivant montre que si le gap est assez grand, alors le sous-module dense de rang 1 est unique.

Lemme 5.0.5. [FPMS22, Lemme 2.8] Soit M un module de rang 2 avec gap $\gamma(M) > 0$ et N le sous-module dense de M de rang 1. Soit N' un autre sous-module de M de rang 1 tel que $N(N') < \gamma(M)^d \cdot \sqrt{N(M)}$ alors $N' \subseteq N$. En particulier, si $\gamma(M) > 1$ alors N est unique et $\forall b \in M$ avec $\|b\| < \gamma(M) \cdot N(M)^{1/(2d)}$, $b \in N$.

Remarque 5.0.6. Tout module M associé à une instance γ-uSVP contient un sous-module $s\mathcal{O}_K$ de rang 1 avec $N(s\mathcal{O}_K) \leqslant (\frac{N(M)^{1/(2d)}}{\gamma})^d = \frac{\sqrt{N(M)}}{\gamma^d}$.

Définition 5.0.7. ((D,γ)-uSVP$_{\text{mod}}$ et γ-wc-uSVPmod)
Soit $\gamma > 0$ et D une distribution sur les instances γ-uSVP. Le problème (D,γ) unique SVP pour les modules de rang 2 demande, étant donné un γ-uSVP module M tiré dans D, de récupérer le sous-module de rang 1 le plus dense, $N \subseteq M$. La variante pire cas (γ-wc-uSVP$_{\text{mod}}$) demande de résoudre le problème pour n'importe quelle instance γ-uSVP $(B,\mathbb{I})$.

Les problèmes NTRU sont juste des cas spéciaux des variantes uSVP.

Définition 5.0.8. (instance NTRU)
Soit $q \geqslant 2$ entier et $\gamma > 0$ réel. Une instance (γ,q)-NTRU est une instance γ-uSVP où la pseudo-base est de la forme $((b_1,\mathcal{O}_K),(b_2,\mathcal{O}_K))$ avec $b_1 = (1,h)^T$ avec $h \in \mathcal{O}_K$ et $b_2 = (0,q)^T$.

On va utiliser des algorithmes présentés dans l'article [FPMS22] comme boîte noire. Nous allons utiliser par exemple l'algorithme uSVP-to-NTRU pour transformer une instance γ_{uSVP}-uSVP en une instance (γ_{NTRU},q)-NTRU.

La réduction qui nous intéresse est énoncée sous la forme suivante.

Théorème 5.0.9. [FPMS22, Lemme 4.1] Soit $K = \mathbb{Q}[X]/(X^d+1)$ un corps cyclotomique avec d une puissance de 2. Soit $\gamma^+ > 1$. Il existe $q_0 = \mathsf{poly}(d,\gamma^+) \in \mathbb{R}_{\geqslant 0}$ et un algorithme uSVP-to-NTRU tel que pour tout $q \geqslant q_0$, $\gamma_{\text{NTRU}} > 1$, $\gamma_{\text{HSVP}} \geqslant \sqrt{d}\Delta_K^{1/2d}$. Soit $\gamma_{\text{uSVP}} = \gamma_{\text{NTRU}} \cdot \sqrt{\gamma_{\text{HSVP}}} \cdot 16\sqrt{2} \cdot d^{3/2}$. Nous avons une réduction de γ-wc-uSVP$_{\text{mod}}$ avec gap $\leqslant \gamma^+$ au (γ_{NTRU},q)-wc-NTRU$_{\text{mod}}$. Cette réduction est polynomiale en taille en bits de l'entrée, $\exp(\frac{d\log(d)}{\log(2q/q_0)})$, si nous avons un oracle qui résout γ_{HSVP}-id-HSVP.

Démonstration. Nous allons utiliser de nombreux lemmes pour prouver le théorème. L'idée sera d'énoncer ces lemmes et d'expliquer leur rôle dans la preuve du théorème. □

Remarque 5.0.10. Toute la section est prise de l'article [FPMS22].

5.1 Arrondir un module

Nous allons décrire l'algorithme `DualRound` qui approche un module de rang k contenu dans $K_{\mathbb{R}}^k$ en un module contenu dans $\mathcal{O}_K^k$ (avec une forme similaire). Nous faisons cela en échantillonnant des vecteurs presque orthogonaux dans le réseau dual.

`DualRound` est paramétré par un paramètre d'écart type $\zeta > 0$, une taille de bloc BKZ $\beta \in \{2,\ldots,kd\}$ et une borne d'erreur $\varepsilon > 0$. Il commence par calculer une $\mathbb{Z}$-base courte de $M^\vee$, en utilisant une variante prouvable de l'algorithme BKZ, par exemple : Slide reduction. Cela offre différents compromis entre temps d'exécution et qualité. Il utilise ensuite l'échantillonneur Gaussien discret avec des paramètres centraux orthogonaux t_i.

Algorithm 5 Algorithme $\texttt{DualRound}_{\varsigma,\beta,\varepsilon}$

Entrée(s): Une pseudo-base $(B, \mathbb{I})$ d'un module de rang $k, M \subset K_{\mathbb{R}}^k$.

1: Calculer une base C_0 de $M^\vee$;
2: Exécuter BKZ avec une taille de bloc β dessus pour obtenir une nouvelle base $\mathbb{Z}$ de $C^\vee$ de $M^\vee$;
3: Poser $R = \varepsilon^{-1}\sqrt{kds}$;
4: **for** $i \in \{1, \ldots, k\}$ **do**
5: Poser $t_i = R \cdot e_i$, où e_i est le i-ème vecteur de la base canonique de $K_{\mathbb{R}}^k$;
6: Pour $i \in \{1, \ldots, k\}$, échantillonner $y_i \leftarrow \hat{D}_{C^\vee,\varsigma,t_i}$;
7: **end for**
8: **return** $\mathbf{Y} = (y_1 \mid \ldots \mid y_k)^\dagger$.

Lemme 5.1.1. [FPMS22, Lemme 3.5] Soient $(B, \mathbb{I})$ une pseudo-base d'un module de rang-k $M \subset K_{\mathbb{R}}^k$. Soit $\beta \in \{2, \cdots, kd\}$, $\varepsilon > 0$, et ς tel que $\varsigma \geq (kd)^{kd/\beta+3/2} \cdot \lambda_{kd}(M^\vee)$. L'algorithme `DualRound` s'exécute en temps polynomial en 2^β, $\log(\varsigma/\varepsilon)$ et la taille de son entrée. De plus, étant donné $(B, \mathbb{I})$, $\texttt{DualRound}_{\varsigma,\beta,\varepsilon}$ produit une matrice $\mathbf{Y} \in M_k(K_{\mathbb{R}})$ telle que

— $(\mathbf{Y}B) \cdot \mathbb{I}$ est contenu dans $\mathcal{O}_K^k$;
— $\mathbf{Y} = R \cdot I_k + \mathbf{E}$ pour $R = \varepsilon^{-1}\sqrt{kd} > 0$ et $\|e_{ij}\| \leq \varepsilon R$ pour tout $i, j \in [k]$.

De plus, si $(B', \mathbb{I}')$ est une autre pseudo-base de M et si $\mathbf{Y}'$ est le résultat de `DualRound` donné cette pseudo-base en entrée, alors

$$dist(\mathbf{Y}, \mathbf{Y}') \leq 2^{-\Omega(kd)}.$$

Remarquons que le lemme n'assure pas nécessairement que la matrice $\mathbf{Y}$ soit inversible, donc le module $(\mathbf{Y}B) \cdot \mathbb{I}$ pourrait ne pas être de rang k. Cependant, en choisissant ε suffisamment petit et en utilisant la deuxième condition sur $\mathbf{Y}$, on peut s'assurer que $\mathbf{Y}$ est en effet inversible. C'est l'objectif du lemme suivant.

Lemme 5.1.2. [FPMS22, Lemme 3.6] Soit $\mathbf{Y} \in K_{\mathbb{R}}^{k \times k}$ telle que $\mathbf{Y} = R \cdot I_k + \mathbf{E}$ pour un certain $R > 0$ et $\|e_{ij}\| \leq \varepsilon \cdot R$ pour tout $i, j \in [k]$. Supposons que $\varepsilon \leq 1/(2k)$. Alors $\mathbf{Y}$ est inversible et nous avons $\mathbf{Y}^{-1} = R^{-1} \cdot I_k + \mathbf{E}'$, avec $\|e'_{ij}\| \leq (k+1) \cdot \varepsilon \cdot R^{-1}$ pour tout $i, j \in [k]$. De plus, il est vérifié que $\det(\mathbf{Y}) \in [(1 + (k+1)(k+2)\varepsilon)^{-d/2}, (1 + 3\varepsilon)^{d/2}] \cdot R^{kd}$.

5.2 Pré-conditionnement d'une instance uSVP

Dans cette section, nous utilisons l'algorithme `DualRound` pour pré-traiter le module d'entrée et contrôler son volume. Afin que la forme normale d'Hermite de notre module intégral ressemble à une instance NTRU, nous modifions légèrement la géométrie de notre

module d'entrée pour qu'il ait ce que nous appelons la propriété de coprimalité. Ainsi, nous décrivons un algorithme, appelé `PreCond`, qui combine tout cela et transforme toute instance uSVP en une nouvelle instance uSVP avec à peu près la même géométrie et avec toutes les propriétés requises que nous voulons.

Définition 5.2.1. (Propriété de coprimalité)
Soit $M \subseteq \mathcal{O}_K^2$ un module de rang 2. M est dit d'avoir la propriété de coprimalité si

$$\{x \in \mathcal{O}_K \mid \exists y \in \mathcal{O}_K, (x,y)^T \in M\} = \mathcal{O}_K$$

Exemple 5.2.2. Un module NTRU possède bien la propriété de coprimalité.

Le lemme suivant nous dit que une telle propriété n'est pas très dure à obtenir.

Lemme 5.2.3. [FPMS22, Lemme 4.3] Soit $(B, \mathbb{I})$ une pseudo-base d'un module de rang-2 $M \subset K^2$ avec un gap $\gamma(M) \geq 1$. Il existe un certain $V_0 > 0$ avec $V_0^{1/(2d)} = \text{poly}(\Delta_K^{1/d}, d, \delta_K, \gamma(M))$ et un algorithme `PreCond` tel que les conditions suivantes soient remplies. Soit $\beta \in \{2, \cdots, 2d\}$ et $V > 0$ tel que $V^{1/(2d)} \geq (2d)^{2d/\beta} \cdot V_0^{1/(2d)}$. Alors, en entrée $(B, \mathbb{I})$, V et β, l'algorithme `PreCond` produit une matrice $\mathbf{Y} \in \mathrm{GL}_2(K)$ telle que

- si $(B, \mathbb{I})$ est une instance γ_{uSVP}-uSVP, alors $(\mathbf{Y}B, \mathbb{I})$ est une instance γ'_{uSVP}-uSVP pour $\gamma'_{\text{uSVP}} = \gamma_{\text{uSVP}}/(2\sqrt{2})$;
- le module de rang-2 $M' := (\mathbf{Y}B) \cdot \mathbb{I}$ est contenu dans $\mathcal{O}_K^2$;
- $N(M') \in [1/2^d, 2^d] \cdot V$;
- M' a la propriété de coprimalité ;
- $\mathbf{Y} = R \cdot I_2 + \mathbf{E}$ pour un certain $R = V^{1/(2d)} \cdot N(M)^{-1/(2d)} > 0$ et $\|e_{ij}\| \leq R/5$ pour tout $1 \leq i, j \leq 2$.

Alors, l'algorithme `PreCond` s'exécute en temps espéré polynomial en la taille de l'entrée, en 2^β.

5.3 Transformer une instance uSVP en une instance NTRU

La deuxième partie de la réduction consiste en trouver un module libre de rang 2 qui contient notre instance uSVP et le transforme en un instance NTRU. Afin d'arriver à ce but, nous utiliserons l'algorithme `BalanceIdeal` qui prend en entrée un idéal fractionnaire I et utilise un oracle γ_{HSVP}-*id*-HSVP pour retourner un élément x tel que $I \subseteq (x)$.

Lemme 5.3.1. [FPMS22, Lemme 4.4] Il existe un algorithme `BalanceIdeal` qui prend en entrée un idéal fractionnaire $I \subset K$ et un paramètre $\gamma_{\text{HSVP}} \geq \sqrt{d} \cdot \Delta_K^{1/(2d)}$, et qui produit un élément $x \in K$ tel que $I \subseteq (x)$ et $|\sigma_i(x)| \in [1 - 1/d, 1 + 1/d] \cdot \sigma^{-1}$ pour tout $i \leq d$, où $\sigma = \gamma_{\text{HSVP}} \cdot d^2 \cdot \delta_K \cdot \mathcal{N}(I)^{-1/d}$.

De plus, en donnant accès à un oracle γ_{HSVP}-id-HSVP, l'algorithme s'exécute en temps polynomial et fait un appel à l'oracle γ_{HSVP}-id-HSVP.

L'algorithme `BalanceIdeal` est le suivant.

Algorithm 6 Algorithme BalanceIdeal

Entrée : Une $\mathbb{Z}$-base d'un idéal fractionnaire $I \subset K$ et un paramètre $\gamma_{\text{HSVP}} \geq 1$

Sortie : Un élément $x \in K$

▷ Utilisation d'un oracle γ_{HSVP}-id-HSVP pour obtenir des vecteurs linéairement indépendants courts de I^{-1}

Appeler un solveur γ_{HSVP}-id-HSVP sur I^{-1} pour obtenir $y \in I^{-1}$

Poser $B = (yr_1, \cdots, yr_d)$ (c'est une $\mathbb{Z}$-base de (y))

▷ Utilisation des vecteurs courts pour trouver un élément équilibré dans I^{-1} en résolvant le CVP

Poser $\sigma = \gamma_{\text{HSVP}} \cdot d^2 \cdot \delta_K \cdot \mathcal{N}(I)^{-1/d}$ et $t = (\sigma, \cdots, \sigma)$

Écrire $t = \sum_i t_i \cdot yr_i$, avec $t_i \in \mathbb{R}$

Définir $s = \sum_i \lfloor t_i \rceil \cdot yr_i$

Retourner $x = s^{-1}$

On va maintenant décrire un algorithme qui transforme une instance uSVP en une instance NTRU.

Algorithm 7 Algorithme Conditioned-to-NTRU

Entrée : Une pseudo-base $(B_1 \cdot \mathbb{I})$ d'un module de rang-2 dans $\mathcal{O}_K^2$ et des paramètres q et γ_{HSVP}

Sortie : Une base B_4 d'un module libre de rang-2 et des informations auxiliaires aux

Calculer la pseudo-base HNF $B_2 \cdot \mathbb{J}$ du module de rang-2 engendré par $B_1 \cdot \mathbb{I}$ ▷ $B_2 = \begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}$

Échantillonner $b \leftarrow \texttt{BalanceIdeal}(J_2, \gamma_{\text{HSVP}})$

Calculer $h = \lfloor a \cdot q / b \rceil$

Retourner $B_4 = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix}$ et $aux = (a, b, J_1, J_2)$

Lemme 5.3.2. [FPMS22, Lemme 4.5] Soit $\gamma_{\text{HSVP}} \geq \sqrt{d}\Delta_K^{1/(2d)}$, $q \in \mathbb{Z}_{>0}$ et $(B, \mathbb{I})$ une pseudo-base d'un module de rang-2 $M \subseteq \mathcal{O}_K^2$. Supposons que nous avons accès à un oracle γ_{HSVP}-id-HSVP. Avec en entrée γ_{HSVP}, q et $(B, \mathbb{I})$, l'algorithme Conditioned-to-NTRU s'exécute en temps polynomial en la taille de l'entrée et fait un appel à l'oracle γ_{HSVP}-id-HSVP.

Lemme 5.3.3. [FPMS22, Lemme 4.6] Soit $\gamma_{\text{HSVP}} \geq \sqrt{d} \cdot \Delta_K^{1/(2d)}$, $\gamma_{\text{NTRU}} > 1$ et $q \in \mathbb{Z}_{>0}$ des paramètres. Définissons

$$V = \gamma_{\text{HSVP}}^d \cdot q^d \cdot d^d$$

et

$$\gamma_{\text{uSVP}} = \gamma_{\text{NTRU}} \cdot \sqrt{\gamma_{\text{HSVP}}} \cdot 8 \cdot d^{3/2} \cdot \delta_K.$$

Soit $(B, \mathbb{I})$ une instance γ_{uSVP}-uSVP dans $\mathcal{O}_K^2$, avec la propriété de coprimalité et avec une norme dans $[1/2^{2d} \cdot V, 2^{2d} \cdot V]$. Alors, avec en entrée $(B, \mathbb{I}), \gamma_{\text{HSVP}}, q$, l'algorithme Conditioned-to-NTRU produit (B_4, aux) tel que B_4 est une instance $(\gamma_{\text{NTRU}}, q)$-NTRU.

Les informations auxiliaires produites par l'algorithme Conditioned-to-NTRU seront utilisées pour relever le sous-module dense de l'instance NTRU vers l'instance uSVP.

Voici un schéma qui illustre l'idée de l'algorithme `Conditioned-to-NTRU`.

Module	Pseudo-base	Vecteur court
M_1	$\left[\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}\right]$	$\mathbf{s}_1 = \begin{pmatrix} u \\ v \end{pmatrix}$
$M_2 = M_1$	$\left[\begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}\right]$ *Étape 1* *HNF*	$\mathbf{s}_2 = \mathbf{s}_1$
$M_3 \supseteq M_2$	$\left[\begin{pmatrix} 1 & 0 \\ a & b \end{pmatrix}\right]$ *Étape 2* *Principalisation*	$\mathbf{s}_3 = \mathbf{s}_2$
M_4	$\left[\begin{pmatrix} 1 & 0 \\ \lfloor a \cdot q/b \rceil & q \end{pmatrix}\right]$ *Étape 3* *Distorsion et arrondi*	$\mathbf{s}_4 = \begin{pmatrix} u \\ v \cdot q/b - u \cdot \{a \cdot q/b\} \end{pmatrix}$

FIGURE 7 – Aperçu de l'algorithme Conditioned-to-NTRU.

5.4 Relever les sous modules denses

Pour finir la réduction, il reste à relever le sous module dense de l'instance NTRU de l'algorithme `Conditioned-to-NTRU` au sous module dense de l'instance uSVP.

Lemme 5.4.1. [FPMS22, Lemme 4.7] Il existe un algorithme `LiftMod` tel que les conditions suivantes soient remplies. Soient $q, \gamma_{\mathrm{HSVP}}$ et $(B, \mathbb{I})$ comme dans le Lemme 5.3.3. Soit M_1 le module de rang-2 engendré par $(B, \mathbb{I})$, $[C, (a, b, J_1, J_2)] \leftarrow$ `Conditioned-to-NTRU`$((B, \mathbb{I}), q, \gamma_{\mathrm{HSVP}})$ et soit M_4 le module libre de rang-2 engendré par C.

Soient (v, J) une pseudo-base du sous-module de rang-1 le plus dense de M_4. Alors, en entrée $a, b, (C, \mathcal{O}_K^2)$ et (v, J), l'algorithme `LiftMod` produit $w \in K$ tel que $\mathrm{Vect}_K(w) \cap M_1$ est le sous-module de rang-1 le plus dense de M_1.

De plus, l'algorithme `LiftMod` s'exécute en temps polynomial.

5.5 Combiner pour obtenir la réduction

Démonstration. [FPMS22, Démonstration du théorème D.1]
Démontrons 5.0.9.
Soit $V_0 = \mathrm{poly}(\Delta_K^{1/d}, d, \delta_K, \gamma^+)$ comme dans le Lemme 5.2.3 (défini en utilisant γ^+ au lieu de $\gamma(M)$). Définissons

$$q_0 = \frac{V_0^{1/d} \cdot 4d}{\gamma_{\mathrm{HSVP}}}.$$

On peut vérifier que q_0 est $\mathsf{poly}\left(\Delta_K^{1/d}, d, \delta_K, \gamma^+\right)$ comme dans l'énoncé. Nous prouvons que le théorème est valide pour ce choix de q_0.

Algorithme uSVP-to-NTRU. En ayant comme entrées $(B, \mathbb{I})$, q et γ_{HSVP}, uSVP-to-NTRU, on fixe $V = \gamma_{\mathrm{HSVP}}^d \cdot q^d \cdot d^d$ et $\beta = \left\lceil \frac{2d \log(2d)}{\log\left(\sqrt{q/q_0}\right) + \log(2d)} \right\rceil$. Il exécute ensuite l'algorithme `PreCond` en entrée $(B, \mathbb{I})$, V et β, pour obtenir une matrice $\mathbf{Y} \in \mathrm{GL}_2(K)$.

D'après la définition de q_0, V et β, on peut vérifier que $V^{1/(2d)} \geq (2d)^{2d/\beta} \cdot V_0^{1/(2d)}$. De plus, nous avons $\gamma(M) \leq \gamma^+$ par hypothèse, donc nous pouvons appliquer le Lemme 5.2.3. Cela implique en particulier que l'appel à l'algorithme `PreCond` s'exécute en temps polynomial en la taille de l'entrée, en $2^\beta = 2^{O(d \log(d)/\log(2q/q_0))}$ et en $\zeta_K(2)$.

L'algorithme uSVP-to-NTRU exécute ensuite l'algorithme `Conditioned-to-NTRU` en entrée $(\mathbf{Y}B, \mathbb{I})$, q et γ_{HSVP}. Il obtient une base B' d'un module libre M' et des informations auxiliaires aux'. L'algorithme uSVP-to-NTRU produit enfin (B', O_K^2) et $aux = (aux', \mathbf{Y}, \gamma_{\mathrm{HSVP}}, B')$.

Nous savons d'après le Lemme 5.3.2 que l'appel à `Conditioned-to-NTRU` peut-être effectué en temps polynomial, avec un appel à l'oracle γ_{HSVP}-id-HSVP. Cela conclut la preuve sur le temps d'exécution de l'algorithme uSVP-to-NTRU.

Supposons maintenant que $(B, \mathbb{I})$ était une instance γ_{uSVP}-uSVP, pour γ_{uSVP} comme dans le Lemme 5.3.3. Nous savons d'après le Lemme 5.2.3 que $(\mathbf{Y}B, \mathbb{I})$ est une instance $\gamma_{\mathrm{uSVP}}/(2\sqrt{2})$-uSVP.

De plus, toujours d'après le Lemme 5.2.3, nous savons que le module engendré par $(YB, \mathbb{I})$ est un module de rang-2 dans $\mathcal{O}_K^2$, avec la propriété de coprimalité et tel que $N(M') \in [1/2^{2d}, 2^{2d}] \cdot V$. Nous pouvons donc appliquer le Lemme 5.3.3 et conclure que (B', O_K^2) est une instance γ_{NTRU} comme souhaité (remarquons que V et $\gamma_{\text{uSVP}}/(2\sqrt{2})$ ont la forme souhaitée pour appliquer le Lemme 5.3.3). Cela prouve le premier point de notre théorème.

Algorithme LiftMod'. Appelons $\tilde{M}$ le module intermédiaire $(\mathbf{Y} \cdot B) \cdot I$ calculé par l'algorithme uSVP-to-NTRU.

En entrée une pseudo-base (v', J') d'un sous-module de rang-1 le plus dense de M' et $aux = (aux', \mathbf{Y}, \gamma_{\text{HSVP}}, B')$, l'algorithme `LiftMod'` exécute `LiftMod`$(aux', B', (v', J'))$ et obtient un vecteur w. Il calcule ensuite J tel que $\text{Vect}(w) \cap \tilde{M} = w \cdot J$, et produit la pseudo-base (v, J).

D'après le Lemme 5.4.1, nous savons que l'algorithme `LiftMod'` s'exécute en temps polynomial. De plus, puisque (v', J') était un sous-module le plus dense de M', nous savons que $w \cdot J$ est un sous-module le plus dense du module $\tilde{M}$. Rappelons que nous avons prouvé que $\tilde{M}$ est une instance $\gamma_{\text{uSVP}}/(2\sqrt{2})$-uSVP, donc nous avons $N(w \cdot J)^{1/d} = \lambda_1(\tilde{M}) \leq 2\sqrt{2}/\gamma_{\text{uSVP}} \cdot N(\tilde{M})^{1/(2d)}$. De la forme spéciale de Y, on peut prouver que $N(\mathbf{Y}^{-1} \cdot w) \leq 4^d \cdot R^{-d} \cdot N(M)^{-1/(2d)}$. Nous obtenons donc

$$N(v \cdot J)^{1/d} \leq 4 \cdot \frac{2\sqrt{2}}{R \cdot \gamma_{\text{uSVP}}} \cdot N(\tilde{M})^{1/(2d)} \leq \frac{16}{\gamma_{\text{uSVP}}} \cdot N(M)^{1/(2d)},$$

où nous avons utilisé la définition de R et le fait que $N(\tilde{M}) \leq 2^d \cdot V$ par le Lemme 5.2.3. Puisque $\gamma_{\text{uSVP}} > 16$, nous concluons que $v \cdot J$ est un sous-module de rang-1 de M avec $N(v \cdot J) < N(M)^{1/2}$ et donc d'après le Lemme 5.0.5, nous concluons que $v \cdot J$ est alors le sous-module le plus dense de M. □

6 Attaque sur le problème de Module Unique-SVP

Nous avons jusqu'à présent étudié différentes attaques sur différents variantes de NTRU, principalement deux attaques : une par les sous-corps et l'autre directement par BKZ. Dans la partie précédente, nous avons introduit les modules avec un gap de rang 2 puis nous avons énoncé une réduction permettant de trouver un sous-module dense d'un module de rang 2 quelconque vers trouver un sous-module dense d'un module NTRU. Notre idée pendant le stage est de regarder si l'une des attaques sur NTRU était applicable pour retrouver un sous-module dense d'un module de rang 2 quelconque. Si c'était le cas, alors nous obtiendrions une connaissance sur la difficulté des problèmes sur les modules de rang 2. L'idée est alors d'utiliser la réduction de la partie précédente pour obtenir une attaque.

Rappel de l'énoncé de la réduction qu'on va utiliser.

Théorème 6.0.1. Soit $K = \mathbb{Q}[X]/(X^d+1)$ un corps cyclotomique avec d une puissance de 2. Soit $\gamma^+ > 1$. Il existe $q_0 = \mathsf{poly}(d, \gamma^+) \in \mathbb{R}_{\geqslant 0}$ et un algorithme uSVP-to-NTRU tel que pour tout $q \geqslant q_0$, $\gamma_{\mathrm{NTRU}} > 1$, $\gamma_{\mathrm{HSVP}} \geqslant \sqrt{d}\Delta_K^{1/2d}$. Soit $\gamma_{\mathrm{uSVP}} = \gamma_{\mathrm{NTRU}} \cdot \sqrt{\gamma_{\mathrm{HSVP}}} \cdot 16\sqrt{2} \cdot d^{3/2}$. Nous avons une réduction de γ-wc-uSVPmod avec gap $\leqslant \gamma^+$ au $(\gamma_{\mathrm{NTRU}}, q)$-$wc$-$NTRU_{\mathrm{mod}}$. Cette réduction est polynomiale en taille en bits de l'entrée, $\exp(\frac{d\log(d)}{\log(2q/q_0)})$, et $\zeta_K(2)$, si nous avons un oracle qui résout γ_{HSVP}-id-HSVP.

Nous avons déjà une réduction des problèmes de module unique SVP au NTRU. Plus précisément, sous de bonnes hypothèses de 5.0.9, si nous parvenons à retrouver le sous-module dense d'un module NTRU, alors nous pouvons aussi récupérer le sous-module dense de n'importe quel module de rang 2.

Nous allons de nouveau utiliser le théorème 5.0.9 pour obtenir une attaque sur le problème de Module Unique-SVP.

En effet, dans [CDW21], Cramer, Ducas et Wesolowski ont donné un algorithme quantique pour $\gamma_{\mathrm{HSVP}} = 2^{\widetilde{O}(\sqrt{d})}$-id-HSVP en temps polynomial pour les corps cyclotomiques.

Lemme 6.0.2. [DvW21, Claim 3.5]
L'algorithme BKZ avec la taille de bloc $\beta = Bd$ appliqué à un réseau NTRU $\tau(M_h)$ avec paramètres $q = \Theta(d^Q)$, $\|(f,g)\| = O(d^S)$ retrouve le sous-réseau dense de $\tau(M_h)$ si

$$B = \frac{8S}{Q^2+1} + o(1).$$

Dans ce cas, l'algorithme BKZ retrouve le sous-réseau dense de $\tau(M_h)$ avec une complexité $2^{O(\beta)}$.

Corollaire 6.0.3. Soit $q \geqslant 2$ et $\gamma > 1$. Soit h une instance NTRU, c'est-à-dire qu'il existe $(f,g) \in \mathcal{O}_K^2 \setminus \{0\}$ avec $g \cdot h \equiv f \mod q$ et $\|(f,g)\| \leqslant \sqrt{q}/\gamma$. Sous l'hypothèse que

le réseau NTRU $\tau(M_h)$ possède un profil HSGZ, il existe un algorithme, plus précisément BKZ, avec une taille de bloc $\beta = O\left(\frac{4\cdot(\log q - 2\log\gamma)\cdot d\log d}{\log^2 q + \log^2 d}\right)$, qui résout $\mathrm{NTRU}_{\mathrm{mod}}$.

Démonstration. Il s'agit de faire coïncider nos instances NTRU avec celles de [DvW21]. Soit $q = \Theta(d^Q)$ et $\|(f,g)\| = O(d^S)$.

$$\begin{aligned}\|(f,g)\| = O(d^S) = \frac{\sqrt{q}}{\gamma} &\iff S\cdot O(\log(d)) = \log\left(\frac{\sqrt{q}}{\gamma}\right).\\ &\iff S\cdot O(\log(d)) = \frac{\log(q)}{2} - \log(\gamma).\\ &\iff S = O\left(\frac{\log(q) - 2\cdot\log(\gamma)}{2\cdot\log(d)}\right).\end{aligned}$$

et

$$q = \Theta(d^Q) \iff \Theta\left(\frac{\log(q)}{\log(d)}\right) = Q.$$

Finalement,

$$\begin{aligned}O(\beta) = O(B\cdot d) &= O(\left(\frac{8S}{Q^2+1} + o(1)\right)\cdot d)\\ &= O\left(\frac{8\cdot(\log q - 2\log\gamma)}{2\cdot\log d}\cdot\frac{d}{\left(\frac{\log q}{\log d}\right)^2 + 1}\right)\\ &= O\left(\frac{4\cdot(\log q - 2\log\gamma)\cdot d\log d}{\log^2 q + \log^2 d}\right)\end{aligned}$$

□

Lemme 6.0.4. [CDW21, Théorème 5.1]
Il existe un algorithme qui résout γ_{HSVP}-id-HSVP pour tout $\gamma_{\mathrm{HSVP}} \geqslant \gamma_0 = 2^{O(\sqrt{d})}$.

Nous arrivons au résultat principal de ce stage, qui est correct sous certaines hypothèses que les deux régimes de NTRU coïncident. On suppose que le petit o de l'expression 4.4.9 est égale à 0.

Idée 6.0.5. Soit K un corps cyclotomique de la forme $K = \mathbb{Q}(\zeta_{2^k})$ de degré d. Soient $\gamma_{\mathrm{NTRU}} > 1$ et $\gamma_{\mathrm{HSVP}} \geqslant 2^{O(\sqrt{d})}$. Soient $\gamma_{\mathrm{uSVP}} = \gamma_{\mathrm{NTRU}}\cdot\sqrt{\gamma_{\mathrm{HSVP}}}\cdot 16\sqrt{2}\cdot d^{3/2}$ et $\gamma^+ > \gamma_{\mathrm{uSVP}}$. Alors pour tout $q \geqslant q_0 = \mathsf{poly}(d,\gamma^+)$ Il existe un algorithme qui résout γ_{uSVP}-wc-uSVP*mod* pour des instances uSVP de gap $\in [\gamma_{\mathrm{uSVP}}, \gamma^+]$. De plus, l'algorithme tourne en temps polynomial.

Démonstration. Soit $\gamma_{\mathrm{NTRU}} > 1$ et $\gamma_{\mathrm{HSVP}} \geqslant 2^{O(\sqrt{d})}$. Alors

$$\begin{aligned}\gamma_{\mathrm{uSVP}} &= \gamma_{\mathrm{NTRU}} \cdot \sqrt{\gamma_{\mathrm{HSVP}}} \cdot 16\sqrt{2} \cdot d^{3/2} \\ &= \gamma_{\mathrm{NTRU}} \cdot 2^{O(d/4)} \cdot 2^4 \cdot 2^{1/2} \cdot 2^{\frac{\log(d)\cdot 3}{2}} \\ &= 2^{O(\frac{d+18+6\cdot\log d}{4})}\end{aligned}$$

Comme $\gamma^+ > \gamma_{\mathrm{uSVP}}$, alors si on prend un $q \geqslant \mathsf{poly}(d, \gamma^+)$, on se retrouve dans le régime de Overstreched NTRU, c'est-à-dire lorsque q est assez grand pour avoir une attaque en temps polynomial sur le problème $\mathrm{NTRU}_{\mathrm{mod}}$. Nous avons par le lemme 6.0.4 un oracle pour le problème de γ_{HSVP}-id-HSVP. Soit M une instance uSVP qui est un module $M \subseteq K^2$ de rang 2 et de gap $\in [\gamma_{\mathrm{uSVP}}, \gamma^+]$. Ainsi, par le théorème 5.0.9, nous avons une réduction en temps $\mathsf{poly}\Big(d, \exp(\frac{d\log(d)}{\log(2q/q_0)})\Big)$ qui transforme en une $(\gamma_{\mathrm{NTRU}}, q)$-NTRU instance. Appliquons alors le corollaire 6.0.3 qui résout $(\gamma_{\mathrm{NTRU}}, q)$-wc-$\mathrm{NTRU}_{\mathrm{mod}}$ en nous renvoyant un sous-réseau dense du réseau NTRU, ce corollaire nous donne un algorithme en temps $\exp(O(\frac{4\cdot(\log q - 2\log\gamma_{\mathrm{NTRU}})\cdot d\log d}{\log^2 q + \log^2 d}))$. Toujours par le théorème 5.0.9, nous arrivons à relever ce sous-réseau dense à un sous-réseau dense de M. Ainsi, la complexité de notre algorithme sera

$$\exp\left(O(\frac{4\cdot(\log q - 2\log\gamma_{\mathrm{NTRU}})\cdot d\log d}{\log^2 q + \log^2 d})\right) + \mathsf{poly}\left(\text{taille en bits de l'entrée}, \exp(\frac{d\log(d)}{\log(2q/q_0)})\right).$$

Il faut alors l'exprimer en fonction de γ_{uSVP}. Nous avons que $\gamma_{\mathrm{NTRU}} = \frac{\gamma_{\mathrm{uSVP}}}{\sqrt{\gamma_{\mathrm{HSVP}}}\cdot 16\sqrt{2}\cdot d^{3/2}}$.

Expression originale :

$$\frac{4\cdot(\log q - 2\log\gamma_{\mathrm{NTRU}})\cdot d\log d}{\log^2 q + \log^2 d}$$

Substitution de γ_{NTRU} avec l'expression fournie :

$$\frac{4\cdot(\log q - 2\log\left(\frac{\gamma_{\mathrm{uSVP}}}{\sqrt{\gamma_{\mathrm{HSVP}}}\cdot 16\sqrt{2}\cdot d^{3/2}}\right))\cdot d\log d}{\log^2 q + \log^2 d}$$

Expansion et simplification des termes logarithmiques :

$$\begin{aligned}&= \frac{4\cdot(\log q - 2\log\gamma_{\mathrm{uSVP}} + 2\log(\sqrt{\gamma_{\mathrm{HSVP}}}\cdot 16\sqrt{2}\cdot d^{3/2}))\cdot d\log d}{\log^2 q + \log^2 d} \\ &= \frac{4\cdot(\log q - 2\log\gamma_{\mathrm{uSVP}} + 2\left(\frac{1}{2}\log\gamma_{\mathrm{HSVP}} + \log 16 + \log\sqrt{2} + \frac{3}{2}\log d\right))\cdot d\log d}{\log^2 q + \log^2 d} \\ &= \frac{4\cdot(\log q - 2\log\gamma_{\mathrm{uSVP}} + \log\gamma_{\mathrm{HSVP}} + 9 + \frac{3}{2}\log d)\cdot d\log d}{\log^2 q + \log^2 d}\end{aligned}$$

Donc, l'expression après avoir substitué et simplifié γ_{NTRU} est :

$$\frac{4\cdot(\log q - 2\log\gamma_{\mathrm{uSVP}} + \log\gamma_{\mathrm{HSVP}} + 9 + \frac{3}{2}\log d)\cdot d\log d}{\log^2 q + \log^2 d}$$

Ainsi, la complexité finale de l'algorithme qui résout γ_{uSVP}-wc-uSVP$_{\text{mod}}$ est en temps

$$\mathsf{poly}\Bigg(\text{taille en bits de l'entrée}, \exp\left(\frac{d\log(d)}{\log(2q/q_0)}\right), \exp\left(\frac{4\cdot(\log q - 2\log\gamma_{\text{uSVP}} + \log\gamma_{\text{HSVP}} + 9 + \frac{3}{2}\log d)\cdot d\log d}{\log^2 q + \log^2 d}\right)\Bigg).$$

□

Rappelons que $q \geqslant \mathsf{poly}(d, \gamma^+)$ et $\gamma_{\text{HSVP}} \geqslant 2^{O(\sqrt{d})}$. En substituant dans l'expression précédente, nous avons un algorithme qui tourne en temps

$$\mathsf{poly}(\text{taille en bits de l'entrée}, \log d, 4\cdot\log d).$$

Le théorème que nous présentons n'est vrai que si les deux régimes de NTRU coïncident et cela s'observe dans la quantité cachée dans le petit o de 4.4.9. Pour illustrer nos intentions, nous utilisons également les algorithmes développés par Wessel van Woerden, disponibles sur NTRU_FATIGUE. L'idée était d'utiliser son algorithme pour obtenir plus de détails sur le petit o en développant plus en détail la quantité $gh(\beta)$ qui se retrouve cachée dans 4.4.9.

Nous avons rencontré un problème en cherchant à détailler les expressions, à savoir que les équations devenaient trop lourdes à résoudre. Cependant, nous pensons qu'en passant plus de temps à étudier les algorithmes de Wessel van Woerden, nous pourrons conclure si notre résultat est possible ou non.

Références

[BGPM22] Katharina Boudgoust, Erell Gachon, and Alice Pellet-Mary. Some easy instances of ideal-svp and implications on the partial vandermonde knapsack problem. In *Advances in Cryptology – CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Proceedings*, number 13508 in Lecture Notes in Computer Science, pages 480–509. Springer, 2022. 42nd Annual International Cryptology Conference, CRYPTO 2022 ; Conference date : 15-08-2022 Through 18-08-2022.

[CDW21] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Mildly Short Vectors in Cyclotomic Ideal Lattices in Quantum Polynomial Time. *Journal of the ACM (JACM)*, 68(2) :1–26, January 2021.

[Cha20] Iván Blanco Chacón. On the rlwe/plwe equivalence for cyclotomic number fields, 2020.

[DvW21] Léo Ducas and Wessel van Woerden. Ntru fatigue : How stretched is overstretched ? Cryptology ePrint Archive, Paper 2021/999, 2021. `https://eprint.iacr.org/2021/999`.

[FPMS22] Joël Felderhoff, Alice Pellet-Mary, and Damien Stehlé. On module unique-svp and ntru. Cryptology ePrint Archive, Paper 2022/1203, 2022. `https://eprint.iacr.org/2022/1203`.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru : A ring-based public key cryptosystem. In *International Workshop on Ant Colony Optimization and Swarm Intelligence*, 1998.

[HPS11] Guillaume Hanrot, Xavier Pujol, and Damien Stehleacute ;. Analyzing blockwise lattice algorithms using dynamical systems. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference*, volume 6841 of *Lecture Notes in Computer Science*, page 441. Springer, 2011.

[PMS21] Alice Pellet-Mary and Damien Stehlé. On the hardness of the ntru problem. Cryptology ePrint Archive, Paper 2021/821, 2021. `https://eprint.iacr.org/2021/821`.