Cryptographie à clef publique
Cours 8

Julien Lavauzelle

Université Paris 8

Master 1 mathématiques et applications – parcours ACC et CSSD

02/04/2021

Séance précédente

Vu à la **séance précédente** :

- Protocoles d'identification
- Signature de Schnorr
- Chiffrement basé sur l'identité : Cocks, Boneh-Franklin

Questions ?

Aujourd'hui

1. Cryptographie post-quantique

L'ordinateur quantique : un nouveau modèle de calcul
Une cryptographie post-quantique ?

2. Cryptographie basée sur les réseaux euclidiens

Fondements mathématiques
NTRU : un premier schéma de chiffrement
Chiffrement fondé sur le problème LWE

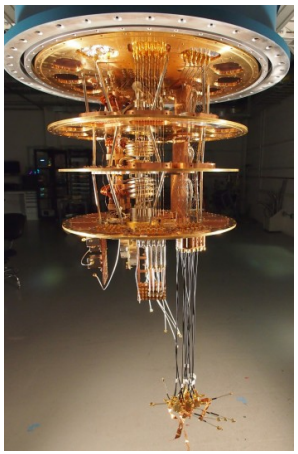
Plan

Plan

Le modèle de calcul quantique

Modèle de calcul **classique** : machine de Turing, opérations effectuées sur des séquences de bits (éléments de $\{0,1\}^n$) avec des opérateurs binaires.

À partir des années 80, introduction d'un nouveau modèle de calcul **quantique**



- $\rightarrow$ opérations sur des états quantiques (**qbits**) modélisés par des éléments $|x\rangle \in \mathbb{C}^{2^n}$,
- $\rightarrow$ avec des **opérateurs logiques quantiques** (au lieu des `xor`, `and`, etc.)
- $\rightarrow$ constructions récentes de processeurs quantiques (< 100 qbits en 2020)
- $\rightarrow$ toujours moins efficace que les machines classiques, mais en progression rapide

année	factorisation de...
2012	21
2016	200 099
2019	1 099 551 473 989

Idée informelle : alors qu'un bit porte l'information d'une **valeur** de $\{0,1\}$, un état quantique porte l'information d'une **distribution** de probabilité sur $\{0,1\}$.

Avantage : on peut effectuer un calcul identique sur une **superposition d'états** en temps constant.

$\rightarrow$ exemple : transformée de Fourier discrète essentiellement en temps $O(1)$.

Le modèle de calcul quantique

D'un point de vue des **classes de complexité**, on a

$$\mathsf{P} \subseteq \mathsf{BPP} \subseteq \mathsf{BQP} \subseteq \mathsf{PSPACE}\,.$$

Pour rappel :
P : problèmes décidables en temps polynomial de manière déterministe
BPP : problèmes décidables en temps polynomial avec probabilité $\geq 2/3$
BQP : problèmes décidables en temps polynomial quantique avec probabilité $\geq 2/3$
PSPACE : problèmes décidables en espace polynomial

On conjecture également que $\mathsf{BQP} \neq \mathsf{BPP}$, et que $\mathsf{BQP} \neq \mathsf{NP}$.

Remarque : le modèle quantique apporte aussi des contraintes. Par exemple, il n'existe pas de porte quantique qui permette de copier un état (*no-cloning theorem*).

Référence pour (beaucoup) plus de détails : notes de cours de R. de Wolf (CWI, Univ. Amsterdam)

`https://homepages.cwi.nl/~rdewolf/qcnotes.pdf`

Problèmes avec accélération quantique (1)

Problème de Deutsch-Josza. Soit $N = 2^n$.

Instance. Un vecteur $\boldsymbol{x} \in \{0,1\}^N$ qui est

1. ou bien constant ($\forall i, j,\ x_i = x_j$)
2. ou bien équilibré ($\sum_i x_i = N/2$).

But. Distinguer si $\boldsymbol{x}$ est constant ou équilibré.

- Modèle déterministe classique : $N/2 + 1$ opérations nécessaires,
- Il existe un algorithme quantique avec 1 **seule requête quantique** et $O(n)$ autres opérations.

Problème de Simon.

Instance. Une fonction $f : \{0,1\}^n \to \{0,1\}$ pour laquelle $\exists \boldsymbol{s} \in \{0,1\}^n$ tel que

$$\forall \boldsymbol{x} \in \{0,1\}^n,\ f(\boldsymbol{x} \oplus \boldsymbol{s}) = f(\boldsymbol{x})\,.$$

But. Trouver la période $\boldsymbol{s}$ de f.

- **Théorème** (Simon) : tout algorithme classique (même probabiliste) est en temps $\Omega(\sqrt{2^n})$
- Il existe un algorithme quantique avec $O(n)$ **requêtes quantiques** et $O(n^4)$ autres opérations.

Problèmes avec accélération quantique (2)

Un autre problème admet une accélération quantique moindre, mais avec beaucoup d'**impact en cryptographie**.

Problème de recherche. Soit $N = 2^n$.

Instance. Un vecteur $\boldsymbol{x} \in \{0,1\}^N$ différent de $\mathbf{0}$.

But. Trouver $i \in \{1,\ldots,N\}$ tel que $x_i = 1$.

Complexité de la résolution

- Dans le pire cas, un algorithme classique nécessite $\Omega(N)$ opérations.
- L'algorithme de **Grover** (1996) nécessite $O(\sqrt{N})$ requêtes quantiques.

Application. Essentiellement, cela réduit de moitié la complexité de l'attaque exhaustive sur les clés de chiffrement. Conséquences pratiques pour les chiffrements symétriques et les fonctions de hachage.

$\rightarrow$ exemple : AES-256 a une sécurité quantique $\leq$ 128 bits.

Plan

Cryptanalyse quantique

En 1994, Shor présente un algorithme qui permet de résoudre les problèmes de la factorisation et du logarithme discret, en **temps polynomial dans le modèle de calcul quantique**.

Algorithms for Quantum Computation : Discrete Logarithms and Factoring. P. Shor. FOCS. **1994**.

Idée très informelle : pour factoriser $n = pq$, on peut chercher l'ordre d'un élement $x \in (\mathbb{Z}/n\mathbb{Z})^\times$.

$$\begin{aligned} x^r \equiv 1 \mod n &\iff (x^{r/2} - 1)(x^{r/2} + 1) \equiv 0 \mod n \\ &\iff \text{pgcd}(x^{r/2} - 1, n) \neq 1 \text{ ou } \text{pgcd}(x^{r/2} + 1, n) \neq 1 \end{aligned}$$

La fonction $f_x : i \mapsto x^i \mod n$ admet alors une **période** r, que l'on peut chercher en adaptant l'algorithme de Simon.

Conséquence. Tous les cryptosystèmes dont la sécurité repose sur la difficulté de la factorisation (par conséquent, également sur la résiduosité quadratique) ou du logarithme discret sont **cassés dans un modèle de calcul quantique**.

On cherche donc de nouveaux systèmes « post-quantiques », c'est-à-dire qui se fondent sur des problèmes **difficiles dans un modèle de calcul quantique**.

L'appel à standardisation du NIST

En 2017, appel à **standardisation de primitives post-quantiques** du NIST (*National Institute of Standards and Technology*).

→ chiffrement & encapsulation de clé, signature

→ + de 60 propositions au premier tour

→ en 2020, dernier tour

type	finalistes	alternatifs
chiffrement	Classic McEliece, CRYSTALS-KYBER, NTRU, SABER	BIKE, FrodoKEM, HQC, NTRU Prime, SIKE
signature	CRYSTALS-DILITHIUM, FALCON, Rainbow	GeMSS, Picnic, SPHINCS+

→ jusqu'en 2024 : phase d'analyse, puis standardisation

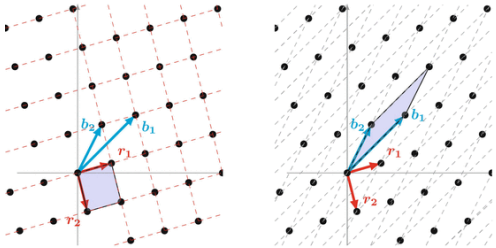
Processus de désignation public, détails ici :

`https://csrc.nist.gov/Projects/post-quantum-cryptography`

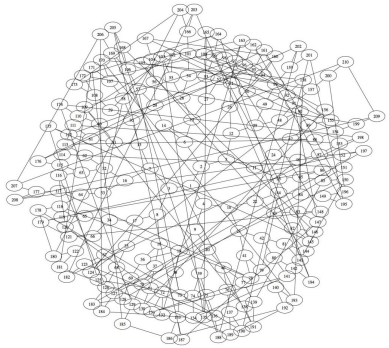
Familles de problèmes

Parmi les **propositions** retenues :

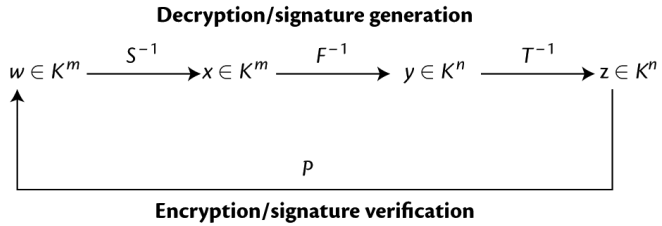
Réseaux euclidiens
(NTRU, CRYSTALS, FALCON, SABER...)



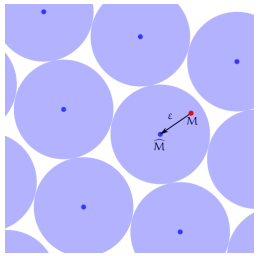
Graphes d'isogénies
(SIKE)



Systèmes polynomiaux multivariés
(Rainbow, ...)



Codes correcteurs
(Classic McEliece, ...)



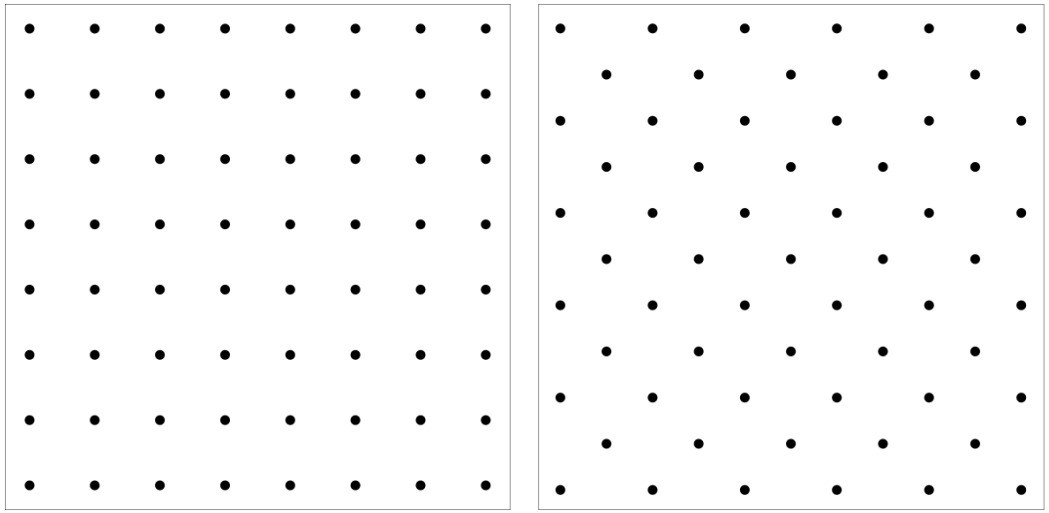
Plan

Plan

Définitions

Définition. Un **réseau euclidien** (*lattice* en anglais) de dimension n est un **sous-groupe discret** $\mathcal{L}$ de $(\mathbb{R}^n, +)$.

Deux réseaux dans $\mathbb{R}^2$:



Exemples.

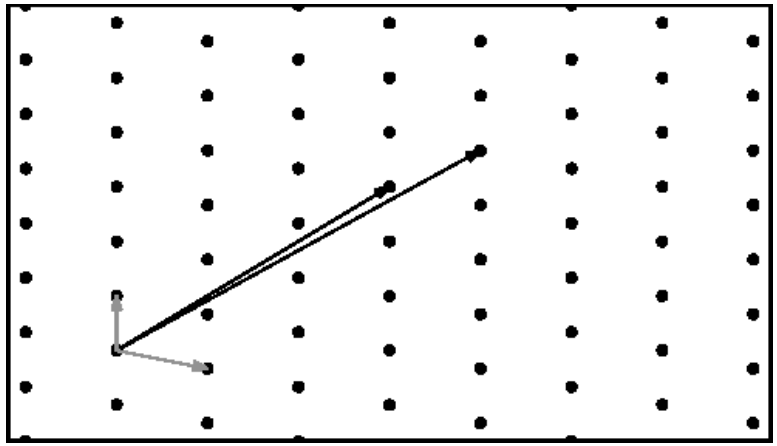
- L'ensemble $\mathbb{Z}^n$.
- Si $\mathcal{L}$ est un réseau, alors pour toute matrice $\boldsymbol{G} \in \mathrm{GL}_n(\mathbb{R})$,

$$\boldsymbol{G}\mathcal{L} := \{\boldsymbol{G}\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{L}\}$$

  est aussi un réseau.

Autres définitions.

- Une **base** d'un réseau est un ensemble de vecteurs libres engendrant le réseau.



deux bases (une bonne et une mauvaise) pour un même réseau

- Le **rang** du réseau $\mathcal{L}$ est le nombre d'éléments dans une base de $\mathcal{L}$. On supposera souvent que le rang vaut n.
- On munit les éléments de $\mathbb{Z}^n$ de la **norme** euclidienne : $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$.
- La **distance minimale** d'un réseau est $\lambda_1(\mathcal{L}) := \min_{\boldsymbol{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\boldsymbol{v}\|_2$.
- Si $\mathcal{L} \subseteq \mathbb{Z}^n$, le **réseau orthogonal** à $\mathcal{L}$, noté $\mathcal{L}^\perp$, est l'ensemble des élements $\boldsymbol{v} \in \mathbb{Z}^n$ tels que pour tout $\boldsymbol{x} \in \mathcal{L}$, on a $\sum_{i=1}^{n} x_i v_i = 0$.

Problèmes difficiles : vecteurs courts

En cryptographie à clef publique, on a besoin de **problèmes difficiles**. Les problèmes de recherche suivants sont NP-difficiles.

On note maintenant, et dans toute la suite, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$.

Problème SVP (*shortest vector problem*), problème du **vecteur le plus court**.

Instance. Une base **quelconque** $\boldsymbol{B}$ d'un réseau $\mathcal{L} \subseteq \mathbb{Z}_q^n$.

But. Trouver $\boldsymbol{v} \in \mathcal{L}$ tel que $\|\boldsymbol{v}\|_2 = \lambda_1(\mathcal{L})$.

Problème SVP_γ (*approximate shortest vector problem*), problème d'**approximation du vecteur le plus court**. Étant donné $\gamma(n) > 1$:

Instance. Une base **quelconque** $\boldsymbol{B}$ d'un réseau $\mathcal{L} \subseteq \mathbb{Z}_q^n$.

But. Trouver $\boldsymbol{v} \in \mathcal{L}$ tel que $\|\boldsymbol{v}\|_2 \leq \gamma(n) \cdot \lambda_1(\mathcal{L})$.

On peut également définir le **problème décisionnel** associé : le problème GapSVP_γ, qui consiste à **distinguer si un réseau possède un vecteur de norme** ≤ 1 **ou** $\geq \gamma(n)$.
$\rightarrow$ utile pour les réductions de sécurité (IND-CPA).

Vecteurs proches et problème SIS

Problème CVP (*closest vector problem*), problème du **vecteur le plus proche**.

Instance. Une base $\boldsymbol{B}$ d'un réseau $\mathcal{L} \subseteq \mathbb{Z}_q^n$, un vecteur $\boldsymbol{x} \in \mathbb{Z}_q^n$.

But. Trouver $\boldsymbol{v} \in \mathcal{L}$ tel que $\|\boldsymbol{v} - \boldsymbol{x}\|_2$ est le plus petit possible.

Problème SIS_β (*short integer solution*), problème de la « solution entière courte » de paramètre β.

Instance. m vecteurs $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{Z}_q^n$

But. Trouver $\boldsymbol{z} = (z_1, \ldots, z_m) \in \mathbb{Z}_q^m$ de poids $\|z\|_2 \leq \beta$ tel que

$$z_1 \boldsymbol{a}_1 + \cdots + z_m \boldsymbol{a}_m = \boldsymbol{0}\,.$$

Question (légitime). En quoi SIS est un problème de réseau ?

Soit $\mathcal{L}$ le réseau défini par la matrice

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \end{pmatrix}$$

On cherche un vecteur $\boldsymbol{z} \in \mathbb{Z}_q^m$ tel que $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{0}$. Autrement dit, on cherche un vecteur court dans le réseau orthogonal $\mathcal{L}^\perp$.

Réduction pire cas – cas moyen

Les problèmes de réseau sont également équipés de certaines réduction « pire cas – cas moyen ».

Par exemple :

Théorème de réduction pire cas – cas moyen (Ajtai 1996, expression informelle). Soit $\mathcal{L}$ un réseau quelconque de $\mathbb{Z}_q^n$ et $\beta \ll q$.
Supposons que l'on connaisse un algorithme qui résolve SIS_β avec bonne probabilité sur une instance aléatoire. Alors on peut résoudre $\mathsf{GapSVP}_{\beta\sqrt{n}}$ dans tout réseau de dimension n.

Conséquence. Cela a un intérêt fort en cryptographie, car les clés sont tirées aléatoirement.

On veut que le problème qui assure la sécurité d'une clé soit « presque aussi difficile » dans le cas moyen (tirage aléatoire) que dans le pire cas (clé de meilleure sécurité).

Quelques algorithmes pour les problème de réseaux

Pour attaquer les problèmes SVP, SIS, etc., on peut essayer de transformer la base en entrée en une meilleure base.

Algorithme LLL, pour Lenstra, Lenstra, Lovasz, publié en 1982.

Étant donnée une base quelconque $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ d'un réseau $\mathcal{L} \subseteq \mathbb{R}^n$, calcule une autre base (dite **base LLL-réduite**) de $\mathcal{L}$, qui est « **presque orthogonale** » et **assez courte**.

Propriétés.

1. L'algorithme LLL termine en temps polynomial, précisément $O(n^6 \log^3(\max \|\boldsymbol{b}_i\|_2))$.
2. Le vecteur le plus court en sortie de LLL a pour norme

$$\|\boldsymbol{b}_1\|_2 \leq 2^{(n-1)/2} \lambda_1(\mathcal{L}) \,.$$

Remarques. Cela reste exponentiellement loin de $\lambda_1(\mathcal{L})$.
$\Longrightarrow$ pas une solution optimale en temps polynomial pour SVP, SIS, etc.

Il existe d'autres bornes en fonction du volume du domaine fondamental du réseau.

Une autre série d'algorithmes, les algorithmes **BKZ** (initiés en 1987), utilise des projections sur des sous-réseaux pour obtenir des vecteurs plus courts.

- Pas d'analyse de complexité théorique, seulement expérimentale.
- Permet parfois d'obtenir des vecteurs plus courts que ceux de LLL.

Rem. : nous étudierons LLL dans le cours d'Algorithmes Arithmétiques II.

Plan

Mathématiques pour NTRU

Contexte mathématique. Soient N, p, q des entiers ($N = 401, p = 3, q = 2048$ en pratique).

Les éléments de l'anneau de polynômes $\mathcal{R} := \mathbb{Z}[X]/(X^N - 1)$ ont des représentants uniques sous la forme

$$A(X) = a_0 + a_1 X + \cdots + a_{N-1} X^{N-1}.$$

On note $\boldsymbol{a} := (a_0, \ldots, a_{N-1}) \in \mathbb{Z}^N$ le vecteur correspondant.

On définit $\boldsymbol{c} := \boldsymbol{a} \star \boldsymbol{b}$ comme le vecteur correspondant au produit $A(X)B(X)$:

$$A(X)B(X) = C(X) \in \mathcal{R} \iff \boldsymbol{a} \star \boldsymbol{b} = \boldsymbol{c} \in \mathbb{Z}^N$$

Remarque. On associe souvent au vecteur $\boldsymbol{a} \in \mathbb{Z}_q^n$ la matrice circulante

$$\boldsymbol{M}(\boldsymbol{a}) = \begin{pmatrix} a_0 & a_1 & & & a_{N-1} \\ a_{N-1} & a_0 & a_1 & & a_{N-2} \\ & & \ddots & & \\ & & & \ddots & \\ a_1 & a_2 & & a_{N-1} & a_0 \end{pmatrix}.$$

Intérêt : calculatoire + voir $\star$ comme un produit vecteur-matrice :

$$\boldsymbol{a} \star \boldsymbol{b} = \boldsymbol{a}\,\boldsymbol{M}(\boldsymbol{b})$$

Le système NTRU

Le cryptosystème **NTRUEncrypt**, souvent abrégé **NTRU** (*N-th degree Truncated polynomial Ring Units*) a été proposé en 1996 par Hoffstein, Pipher et Silverman.

Soit $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$. On choisit $S_q := \{-\frac{q-1}{2}, \dots, \frac{q-1}{2}\}$ pour représenter les éléments de $\mathbb{Z}_q$ (**représentation « centrée »**).
Pour $\boldsymbol{a} \in \mathbb{Z}_q^N$, on définit $\|\boldsymbol{a}\|_2 := \sqrt{\sum_{i=1}^N a_i^2}$ où les a_i sont pris dans $\{-\frac{q-1}{2}, \dots, \frac{q-1}{2}\}$.

Les valeurs de N, p et q sont des paramètres publics. On note $\mathcal{R} = \mathbb{Z}_q[X]/(X^N - 1)$.

NTRU : GÉNÉRATION DE CLEFS

1. Tirer $U(X)$ et $V(X)$ deux polynômes dans $\mathcal{R}$, dont les coefficients sont tirés uniformément dans $S_p = \{-\frac{p-1}{2}, \dots, \frac{p-1}{2}\}$.
2. Calculer $F(X) = 1 + pU(X)$ et $G(X) = pV(X)$.
3. Si $F(X)$ n'est pas inversible mod $X^N - 1$, recommencer pour U et F les étapes 1. et 2.
4. Calculer $F(X)^{-1} \mod (X^N - 1)$ et $H(X) = G(X)F(X)^{-1} \mod (X^N - 1)$.
5. La clé publique est $\boldsymbol{h} \in \mathbb{Z}_q^N$, la clé privée est $\boldsymbol{f} \in \mathbb{Z}_q^N$.

Remarque. Les polynômes U et V sont tirés uniformément dans « $\mathcal{R} \mod p$ ».

Le système NTRU

L'espace des **clairs** est $\mathcal{M} = S_p^N$ avec $S_p = \{-\frac{p-1}{2}, \dots, \frac{p-1}{2}\}$. L'espace des **chiffrés** est $\mathbb{Z}_q^N$.

NTRU : CHIFFREMENT

Pour chiffrer $\boldsymbol{m} \in S_p^N$:

1. Tirer uniformément $\boldsymbol{r} \in S_p^N$.
2. Calculer et retourner $\boldsymbol{y} = \boldsymbol{r} \star \boldsymbol{h} + \boldsymbol{m} \mod q$.

NTRU : DÉCHIFFREMENT

1. Calculer $\boldsymbol{a} = \boldsymbol{y} \star \boldsymbol{f} \mod q$.
2. Retourner $\boldsymbol{m}' = \boldsymbol{a} \mod p$ (en écrivant les coefficients dans S_p au lieu de $[0, p-1]$).

Exercice. Quel est le facteur d'expansion (ratio entre la taille du clair et celle du chiffré) dans NTRU ?

Le système NTRU : exemple

Exemple pour $N = 23$, $p = 3$ et $q = 31$ (pris dans *Cryptography : theory and practice*, Stinson & Paterson)

Génération de clefs. On choisit

$$U(X) = X^{18} - X^9 + X^8 - X^4 - X^2, \text{ donc } F(X) = 3X^{18} - 3X^9 + 3X^8 - 3X^4 - 3X^2 + 1$$

et

$$V(X) = X^{17} + X^{12} + X^9 + X^3 - X, \text{ donc } G(X) = 3X^{17} + 3X^{12} + 3X^9 + 3X^3 - 3X\,.$$

Cela donne (du coeff. le plus grand au plus petit) :

$$\boldsymbol{h} = [-13, -15, 0, 12, -14, 0, 8, -14, -6, 14, -3, 7, -5, -14, 3, 10, 5, -8, 0, 0, 1, 8]$$

Chiffrement de $\boldsymbol{m}$ tel que $M(X) = X^{15} - X^{12} + X^7 - 1$. La clé publique est $\boldsymbol{h}$.
On choisit $\boldsymbol{r}$ tel que $R(X) = X^{19} + X^{10} + X^6 - X^2$. Le chiffré est alors $\boldsymbol{y} = \boldsymbol{r} \star \boldsymbol{h} + \boldsymbol{m} \mod q$, soit

$$\boldsymbol{y} = [5, -15, 4, 8, 10, -15, 6, 8, -8, 3, -10, -7, -1, -9, 12, -14, 15, -10, 15, -14, -5, -15, -3]$$

Déchiffrement de $\boldsymbol{y}$. On calcule $\boldsymbol{a} = \boldsymbol{f} \star \boldsymbol{y} \mod q$:

$$\boldsymbol{a} = [6, 3, -6, -3, 0, -3, 0, \mathbf{7}, 0, 6, \mathbf{-1}, -9, 3, 3, 0, \mathbf{-5}, 0, 0, 6, 3, 6, -3, \mathbf{5}]$$

Puis, on réduit $\boldsymbol{a}$ modulo $p = 3$ et on obtient la réprésentation de M.

Le système NTRU : validité

Résumé.

1. $\mathtt{sk} = f = 1 + pu$,
   $\mathtt{pk} = g \star (f^{-1} \mod q) = (pv) \star (f^{-1} \mod q)$ où u et v sont à coeff. dans S_p.
2. Chiffrement $y = r \star h + m \mod q$ avec r aléatoire à coeff. dans S_p.
3. Déchiffrement $m' = (f \star y \mod q) \mod p$.

Validité. On a

$$f \star y \equiv f \star (r \star h + m) \equiv r \star (f \star h) + f \star m \equiv r \star g + f \star m \mod q$$

Si tous les coefficients de $r \star g + f \star m$ sont dans $\{-\frac{q-1}{2}, \ldots, \frac{q-1}{2}\}$, alors on a

$$f \star y = r \star g + f \star m\,.$$

Rappelons-nous que $f = 1 + pu$ et $g = pv$, donc

$$f \star y \equiv r \star (pv) + (1 + pu) \star m \equiv m + p(v + u \star m) \equiv m \mod p\,.$$

Conséquence. Afin que le déchiffement fonctionne, on choisit N, p et q de sorte que les coefficients a_i de $r \star g + f \star m$ soient tels que $|a_i| \leq \frac{q-1}{2}$ avec très bonne probabilité.

Le système NTRU

Questions/exercice.

1. Quelle est la valeur maximale de $\|\boldsymbol{u}\|_2$ pour $\boldsymbol{u} \in \mathbb{Z}_q^N$?
2. Si $\boldsymbol{u} \in \mathbb{Z}_q^N$ est uniforme, que vaut typiquement $\|\boldsymbol{u}\|_2$?
3. Calculer une borne supérieure sur $\|\boldsymbol{f}\|_2$ en fonction de p et N.
4. Peut-on majorer $\|\boldsymbol{h}\|_2$ de manière similaire ?
5. En considérant que $\boldsymbol{r}$ et $\boldsymbol{g}$ ont chacun $N/3$ coordonnées nulles, quelle valeur maximale peut atteindre un coefficient de $\boldsymbol{r} \star \boldsymbol{g}$?
6. Comparer avec la valeur de $q/2$ pour les choix $N = 401$, $q = 2048$ et $p = 3$.

Le système NTRU : sécurité

Observation. Attaquer la clé de NTRU s'apparente à résoudre un problème CVP.

Soit $\mathcal{L}$ le réseau (dans $\mathbb{Z}^{2N}$) engendré par la matrice

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{M}(\boldsymbol{h})^\top & q\boldsymbol{I} \end{pmatrix} \in \mathbb{Z}^{2N \times 2N}$$

On a $\mathcal{L} = \{(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{Z}^{2N}, \boldsymbol{a} \star \boldsymbol{h} \equiv \boldsymbol{b} \mod q\}$. Comme $\boldsymbol{f} \star \boldsymbol{h} \equiv \boldsymbol{g} \mod q$, on obtient $(\boldsymbol{f}, \boldsymbol{g}) \in \mathcal{L}$.

Par ailleurs, en moyenne on a

$$\|(\boldsymbol{f}, \boldsymbol{g})\|_2 \simeq \sqrt{\frac{N}{3}\left(1^2 + (-1)^2 + p^2 + (-p)^2\right)} = \sqrt{\frac{2N(1+p^2)}{3}} = 2\sqrt{\frac{5N}{3}} \quad \text{pour } p = 3\,.$$

Dans un réseau de dimension $2N$, un vecteur aléatoire aurait une norme $q\sqrt{\frac{N}{6}}$.

Si $q \gg p^2$, le vecteur $(\boldsymbol{f}, \boldsymbol{g})$ est **particulièrement court** dans le réseau $\mathcal{L}$. On peut le chercher avec un algorithme qui résout CVP.

Conséquence. Si CVP est facile en dimension $2N$, alors on peut attaquer le chiffrement NTRU. Pas de preuve pour la réciproque...

En pratique

Pour obtenir une sécurité IND-CPA, voire IND-CCA2, il faut incorporer des **fonctions de hachage**.

Dans la proposition NTRU sélectionnée pour standardisation au NIST, et pour une sécurité « quantique » de 128 bits :

- $N = 509$ et $q = 2048$,
- clés publique/privée de taille 699/935 octets (compressées),
- chiffrés de taille 699 octets,
- chiffrement et déchiffrement rapides ($< 2\,000\,000$ cycles).

Remarque. Il existe une signature fondée sur NTRU, intitulée **Falcon** et sélectionnée pour standardisation au NIST.

Plan

Le problème LWE

Problème LWE (*learning with errors*). Soient q un nombre premier, n un entier, et $\mathcal{E}$ une variable aléatoire de distribution $\pi_{\mathcal{E}}$, à valeurs dans $\mathbb{Z}_q$.

Instance. Une séquence de m échantillons $(\boldsymbol{a}_i, b_i) \in \mathbb{Z}_q^{n+1}$, où

- chaque $\boldsymbol{a}_i$ est choisi uniformément dans $\mathbb{Z}_q^n$,
- $b_i = e_i + \sum_{j=1}^{n} a_{i,j} s_j \mod q$, avec e_i tiré selon $\mathcal{E}$
- les $s_j \in \mathbb{Z}_q$ sont quelconques.

But. Trouver $\boldsymbol{s}$.

Remarque. Si l'on note $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$ la matrice des $(a_{i,j})$, alors le problème LWE s'apparente à

$$\text{trouver } \boldsymbol{s} \text{ tel que } \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \equiv \boldsymbol{b} \mod q, \quad \text{avec } \boldsymbol{e} \sim \mathcal{E}^m.$$

Difficulté. Il existe des paramètres pour lesquels le problème LWE est difficile :

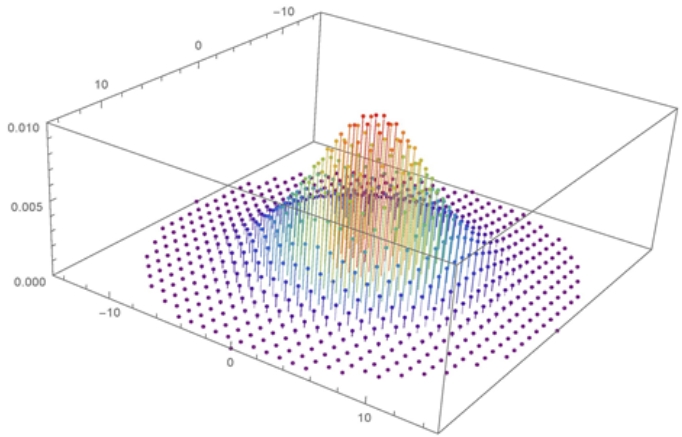
- m « petit » devant n,
- distribution $\mathcal{E}$ centrée en 0

Cryptosystème de Regev

Paramètres publics.

- Des entiers n et m, un nombre premier q,
- Une variable aléatoire $\mathcal{E}$ sur $\mathbb{Z}_q$, telle que de petites valeurs sont très probables.

En pratique, on utilise une Gaussienne discrète centrée d'écart type σ petit comparé à q.



REGEV : GÉNÉRATION DE CLEFS

1. Tirer $\boldsymbol{s}$ uniformément dans $\mathbb{Z}_q^n$.
2. Tirer uniformément une matrice $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$.
3. Tirer $\boldsymbol{e} = (e_1, \ldots, e_m)^\top$, où les e_i sont tirés selon $\mathcal{E}$.
4. Calculer $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \mod q$.
5. La clé publique est $(\boldsymbol{A}, \boldsymbol{b})$, la clé privée est $\boldsymbol{s}$.

Chiffrement de Regev

L'espace des **clairs** est $\{0,1\}$. L'espace des **chiffrés** est $\mathbb{Z}_q^{n+1}$. On note $\langle \cdot, \cdot \rangle$ le produit scalaire.

Regev : chiffrement

Pour chiffrer **un bit** $x \in \{0,1\}$.

1. Définir $\boldsymbol{r} = (r_1, \ldots, r_m)$, où les r_i sont tirés uniformément dans $\{0,1\}$
2. Si $x = 0$, alors définir $y = (\boldsymbol{rA}, \langle \boldsymbol{r}, \boldsymbol{b} \rangle)$
3. Si $x = 1$, alors définir $y = (\boldsymbol{rA}, \lfloor \frac{q}{2} \rfloor + \langle \boldsymbol{r}, \boldsymbol{b} \rangle)$
4. Retourner y.

Regev : déchiffrement

Pour déchiffrer $(\boldsymbol{u}, v)$.

1. Calculer $z = v - \sum_{j=1}^{n} u_j s_j$.
2. Si $|z| < \left| \lfloor \frac{q}{2} \rfloor - z \right|$, retourner $x' = 0$.
3. Sinon, retourner $x' = 1$.

Chiffrement de Regev

Validité. Que vaut $z = v - \sum_{j=1}^{n} u_j s_j$?
En notation matricielle, si les vecteurs sont notés comme des vecteurs colonnes :

$$\begin{aligned} z = v - \boldsymbol{u}^\top \cdot \boldsymbol{s} &= \left(\boldsymbol{r}^\top \cdot \boldsymbol{b} + x \lfloor \tfrac{q}{2} \rfloor\right) - \left(\boldsymbol{r}^\top \cdot \boldsymbol{A}\right) \cdot \boldsymbol{s} \\ &= \boldsymbol{r}^\top \cdot (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{s}) + x \lfloor \tfrac{q}{2} \rfloor = \boldsymbol{r}^\top \cdot \boldsymbol{e} + x \lfloor \tfrac{q}{2} \rfloor \end{aligned}$$

Comme $\mathcal{E}$ suit une loi gaussienne discrète centrée en 0 et de petite variance, on a

$$\boldsymbol{r}^\top \cdot \boldsymbol{e} \ll q\,.$$

Ainsi, on retrouve x suivant si z est proche de 0 ou de $\lfloor \frac{q}{2} \rfloor$.

Remarque. C'est donc un déchiffrement probabiliste.

Sécurité. La sécurité du chiffrement de Regev repose sur deux points :

1. impossibilité de distinguer $\boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$ d'un élément uniforme de $\mathbb{Z}_q^m$ (variante décisionnelle du problème LWE)
2. impossibilité de distinguer $(\boldsymbol{r}^\top \boldsymbol{A}, \boldsymbol{r}^\top \boldsymbol{b})$ d'un élément uniforme de $\mathbb{Z}_q^{n+1}$. Un théorème (*leftover hash lemma*) l'assure dès lors que m n'est pas trop grand, typiquement $m \ll n \log_2 q$.

Chiffrement de Regev : efficacité et variantes

Constat. Le chiffrement de Regev est peu pratique : un seul bit $x \in \{0,1\}$ est chiffré en $(n+1)\log q$ bits : $(\boldsymbol{u}, v) \in \mathbb{F}_q^{n+1}$.

On peut gagner en efficacité en

- remplaçant les vecteurs $\boldsymbol{s}$, $\boldsymbol{e}$ et $\boldsymbol{b}$ par des matrices à $k \simeq n$ colonnes ;
- parallélisant le chiffrement (même $\boldsymbol{r}$ pour plusieurs bits x).

Néanmoins, cela augmente la taille des clés.

Variantes. Il existe des variantes avec des réseaux **structurés**. L'idée est de remplacer la matrice aléatoire uniforme $\boldsymbol{A}$ par une matrice toujours aléatoire mais plus structurée, afin d'obtenir des **clés plus courtes** et des **calculs plus efficaces**.

- On peut prendre $\boldsymbol{A}$ circulante, comme $\boldsymbol{M}(\boldsymbol{a})$ vue précédemment : il suffit alors de stocker $\boldsymbol{a}$ (clé plus courte). Cela correspond à considérer l'anneau de polynômes $\mathbb{F}_q[x]/(x^n - 1)$.
- On peut choisir d'autres anneaux de polynômes $\mathbb{F}_q[x]/(\Phi(x))$: ce sont les familles de chiffrements « Ring-LWE » et « Module-LWE ».

Par exemple, le cryptosystème **CRYSTALS-Kyber** a été retenu par le NIST, et propose des tailles de clé ≤ 2 ko pour une sécurité de 128 bits.

`https://pq-crystals.org/kyber/index.shtml`

Fin de la partie cours

Questions ?