La cryptographie du futur

Abderrahmane Nitaj
Laboratoire de Mathématiques Nicolas Oresme
Université de Caen, France
nitaj@math.unicaen.fr
http://www.math.unicaen.fr/~nitaj

Résumé

Sans nous rendre compte, la cryptographie fait déjà partie de notre vie quotidienne. Notre carte bancaire, nos emails quotidiens, certains dvd, les télévisions par satellite,.... fonctionnent de façon plus ou moins sûre grâce à la cryptographie. Dans le futur, la cryptographie est appelée à jouer un rôle encore plus important, non seulement dans les nouvelles technologies, mais aussi dans la vie privée et publique : le vote électronique, les télécommunications, le payement électronique ... sont appelés à se développer de façon importante, et les moyens cryptographiques doivent suivre cette évolution.

On présente ici les principes fondamentaux de la cryptographie, un peu de son histoire, son présent et son développement dans le futur. On donne en particulier un aperçu des principaux procédés cryptographiques utilisés massivement actuellement, tels que RSA, El Gamal, ECC ainsi qu'un aperçu des procédés appelés à jouer un rôle important dans le futur, notamment NTRU et LWE.

قَالَ المُتَنَبّي

وَلَم أَرَ في عُيُوبِ النَّاسِ عيبًا كَنَقصِ القَادرينَ عَلَى التَّمَامِ

وَقَالَ ايضًا

فَاطلُبِ العِزَّ في لظَى وَدعِ الذُّ لَّ وَلَو كَانَ في جِنَانِ الخُلُودِ

1 Introduction

La cryptographie est l'art de chiffrer et de déchiffrer les textes par des moyens définis par les entités qui communiquent entre elles. La cryptanalyse est l'art de déchiffrer les textes chiffrés par des entités qui interceptent les communications, autre que les destinataires. Il est facile de se rendre compte de l'importance de la cryptographie dans le monde moderne, mais son invention remonte à plus de 4000 ans. En effet, les premiers symboles hiéroglyphiques représentaient des textes chiffrés et seuls quelques scribes pouvaient les déchiffrer. Une fois l'écriture hiéroglyphique s'est répondue, l'aspect cryptographique a disparu. Ce principe s'est répété dans plusieurs autres civilisations. En Chine par exemple, l'écriture, qui était très peu répandu, pouvait être considérée comme une technique cryptographique. Les chinois avaient aussi développé la technique de la stéganographie qui consiste à dissimuler les messages. On retrouve d'autres aspects de la cryptographie dans d'autres civilisations antiques : indienne, perse, mésopotamienne, greque et romaine. Les techniques cryptographiques romaines par exemple étaient basées sur le décalage de l'alphabet. Cette technique cryptographique est appelée *chiffre de César*.

La cryptanalyse, science du déchiffrement, naquit avec l'essor de la civilisation arabo-musulmane. Le nom dominant de cette époque est certainement Al Kindi (801-873) ابُو يُوسُف يَعقُوب إِبنُ إِسحَاق الكِندي qui travailla pour Al Mamoun المَامُون à la Maison de la Sagesse, Baitu alhikmata بَيتُ الحِكمَةَ. La technique de cryptanalyse inventée par Al Kindi est basée sur l'analyse des fréquences et fut utilisée jusqu'à la fin de la première guerre mondiale en 1918.

En 1918, les allemands Arthur Scherbius et Richard Ritter inventèrent une machine à chiffrer électrique et automatique : Enigma. Pendant une longue période, les cryptographes polonais, français, anglais et allemands bataillèrent pour casser ou renforcer Enigma. Ce ne fut qu'en 1940 qu'Alan Turing a pu inventer une machine pour déchiffrer les textes chiffrés par Enigma.

Aujourd'hui, la cryptographie est utilisée dans un grand nombre de produits. On la trouve ainsi dans les votes électroniques, les signatures électroniques, le paiement par cartes bancaires, le courrier électronique, les bases de données, les cartes à puces, les portes monnaies électroniques, les décodeurs numériques, les achats électroniques, le téléphones cellulaires...

2 Le présent

Dans les processus cryptographiques modernes, on distingues deux types de cryptographie, la cryptographie à clé secrète et la cryptographie à clé publique. Dans la cryptographie à clé secrète, le chiffrement est symétrique et utilise la même clé pour chiffrer et déchiffrer. Dans la cryptographie à clé publique, inventée par Diffie et Hellman [3], le chiffrement est asymétrique et utilise une clé pour chiffrer et une clé différente pour déchiffrer. Ces deux clés sont généralement liées entre elles par des formules plus ou moins complexes. L'inconvénient de la cryptographie à clé secrète est que les entités doivent échanger les clés. De plus, chaque entité doit avoir échangé autant de clés que de partenaires. Pour cette raison, la cryptographie à clé publique joue un rôle important dans les communications modernes car une seule clé, rendu publique par une entité, peut servir pour tous ses partenaires. Le principal inconvénient de la cryptographie à clé publique vient du fait que les temps de calculs deviennent importants. Pour pallier à cet inconvénient, il est possible d'utiliser une cryptographie hybride, qui consiste à utliser la cryptographie à clé secrète en échangeant les clés par la cryptographie à clé publique.

Comme exemples de systèmes symétriques, on peut citer DES (Data Encryption Standard), remplacé en 2000 par Rijndael, le standard AES (Advanced Encryption Standard) de NIST (National Institute of Standards and Technology).

Parmi les exemples de systèmes asymétriques, on peut citer RSA, El Gamal, ECC et NTRU.

- Le cryptosystème RSA a été inventé en 1977 par R. Rivest, A. Shamir et L. Adleman [17]. La sécurité de RSA est basée sur la difficulté de factoriser les grands nombres entiers du type $N = pq$ où p et q sont des nombres premiers, et sur la difficulté d'extraire la racine n-ième d'un entier modulo un grand nombre entier dont la factorisation est inconnue :
  Etant donné des nombres entiers n, N et a, déterminer un entier x tel que
  $$x^n \equiv a \pmod{N}.$$
- Le cryptosystème El Gamal a été inventé par T. El Gamal [4] en 1985. La sécurité de ce cryptosystème est basée sur le problème du logarithme discret :
  Etant donné deux nombres entiers g et a et un nombre premier p,
  déterminer un entier x tel que $g^x \equiv a \pmod{p}$.
- L'idée des cryptosystèmes de type ECC (Elliptic Curve Cryptography) à été introduite de façon indépendante en 1985 par N. Koblitz [7] et V.S. Miller [12]. La sécurité de ces cryptosystèmes est basée sur le problème du logarithme discret elliptique :
  Etant donné deux points P et Q d'une courbe elliptique E, déterminer un entier n tel que $nP = Q$.
- Le cryptosystème NTRU a été inventé entre 1996 et 1998 par J.H. Silverman,

J. Hoffstein et J. Pipher [5]. La sécurité de NTRU est basée sur le problème du plus court vecteur non nul d'un réseau (SVP) :
Etant donnée une base B d'un réseau L, déterminer un vecteur non nul de L, le plus court possible pour la norme euclidienne.

2.1 RSA

Depuis son invention en 1977, RSA est devenu un cryptosystème central en cryptographie.

2.1.1 Générations de clés

Le module RSA est un nombre entier $N = pq$ qui est le produit de deux nombres premiers p et q. La fonction indicatrice d'Euler est définie par $\varphi(N) = (p-1)(q-1)$. Soit e un entier vérifiant $3 \leq e < \varphi(N)$ et $\gcd(e, \varphi(N)) = 1$. On peut donc déterminer un entier d qui vérifie

$$ed \equiv 1 \pmod{\varphi(N)}.$$

L'entier e est appelé exposant publique et d est l'exposant privé correspondant. La clé publique est (N, e), alors que la clé privée est (N, d).

2.1.2 Chiffrement d'un message

Pour chiffrer un message clair M, on le transforme en un nombre entier positif m avec $m < \varphi(N)$ et on calcule l'entier chiffré c en utilisant la clé publique (N, e) :

$$c \equiv m^e \pmod{N}.$$

Le message chiffré c peut alors être transmis.

2.1.3 Déchiffrement d'un message

Pour déchiffrer un message chiffré c et retrouver le message clair m, il suffit de calculer

$$m \equiv c^d \pmod{N}.$$

L'exactitude du déchiffrement est basée sur la propriété suivante :

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\varphi(N)} \equiv m\left(m^{\varphi(N)}\right)^k \equiv m \pmod{N},$$

où k est l'unique entier vérifiant $ed = 1 + k\varphi(N)$. La dernière égalité provient du théorème suivant.

Théorème 2.1 (Euler)**.** *Soit N un nombre entier et $\varphi(N)$ l'indicateur d'Euler. Si a est un entier premier avec N, alors*

$$a^{\phi(N)} \equiv 1 \pmod{N}.$$

2.2 El Gamal

Inventé en 1985 par T. El Gamal [4], le cryptosystème El Gamal a pour univers un groupe fini.

2.2.1 Générations de clés

Soient p un grand nombre premier et g un générateur du groupe fini $\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z}^*$. Soit a un nombre entier vérifiant $2 \leq a \leq p-1$. On pose

$$b \equiv g^a \pmod{p}.$$

La clé publique est le triplet (p, g, b), tandis que la clé privée est (p, g, a).

2.2.2 Chiffrement d'un message

Pour chiffrer un message clair M, on le transforme en un nombre entier positif m avec $m < p-1$. Soit k un nombre entier aléatoire avec $2 \leq k \leq p-2$. On calcule alors les quantités γ et δ en utilisant la clé publique (p, g, b) par :

$$\gamma \equiv g^k \pmod{p}, \qquad \delta \equiv mb^k \pmod{p}.$$

Le message chiffré est le couple (γ, δ) et peut alors être transmis.

2.2.3 Déchiffrement d'un message

Pour déchiffrer un message chiffré (γ, δ) et retrouver le message clair m, il suffit d'utiliser la clé privée (p, g, a) et de calculer

$$m \equiv \gamma^{-a}\delta \pmod{p}.$$

L'exactitude du déchiffrement est basée sur la propriété suivante :

$$\gamma^{-a}\delta \equiv \left(g^k\right)^{-a} mb^k \equiv \left(g^k\right)^{-a} m \left(g^a\right)^k \equiv m \pmod{p}.$$

2.3 ECC

La théorie des courbes elliptiques est très ancienne mais son utilisation en cryptographie date de 1985 et a été proposée indépendamment par N. Koblitz [7] et V.S. Miller [12]. L'idée était de transformer les cryptosystèmes utilisant des groupes finis du type $\mathbb{F}_p^*$ en cryptosystèmes utilisant l'arithmétique des courbes elliptiques. Pour une introduction aux courbes elliptiques et leur utilisation en cryptographie, on peut consulter par exemple [11].

Soit K un corps. Par exemple, $K = \mathbb{Q}$, $K = \mathbb{R}$, $K = \mathbb{C}$, $K = \mathbb{F}_p$ ou $K = \mathbb{F}_{p^r}$, où p est un nombre premier et $r \geq 1$. Une courbe elliptique définie sur K peut être caractérisée par son equation Weierstrass.

Définition 2.2. *Une courbe elliptique E définie sur K est une courbe dont l'équation de Weierstrass est*

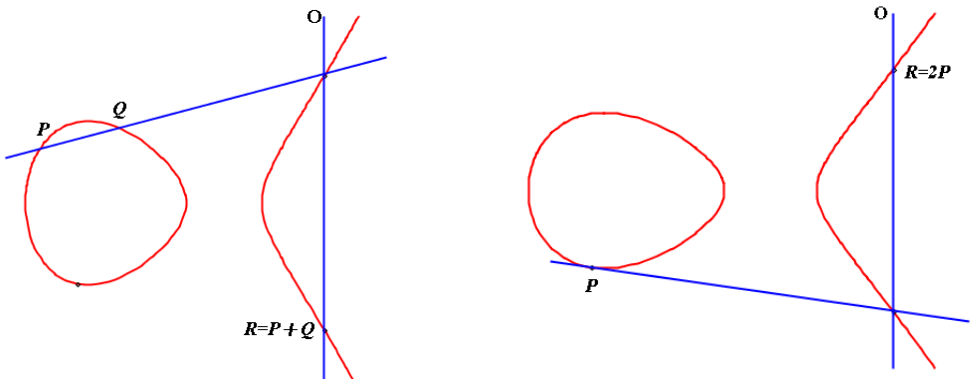
$$E: \quad y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

en plus du point à l'infini $\mathcal{O} = (0:1:0)$. L'union de $\{\mathcal{O}\}$ et de l'ensemble des points $P = (x, y) \in K^2$ de la courbe E est noté $E(K)$.

Une particularité importante de $E(K)$ est sa structure de groupe additif. Tout d'abord, l'opposé d'un point $P(x, y)$ est le point $-P$ défini par

$$-P = (-x, -y - a_1x - a_3).$$

La somme de deux point P et Q est un point R obtenu en utilisant la droite qui passe par P et Q si $P \neq Q$ ou la tangente à la courbe en P si $P = Q$ comme indiqué dans les figures ci-dessous.



En utilisant les coordonnées des points, le doublement et l'addition sont définies de façons explicites.

- **Doublement.** Soit $P = (x, y)$ un point de $E(K)$.
  1. Si $2y + a_1x + a_3 = 0$, alors $2P = \mathcal{O}$.
  2. Si $2y + a_1x + a_3 = 0$, alors $2P = (x_3, y_3)$ avec

$$\begin{aligned} x_3 &= -2x - a_2 + a_1\lambda + \lambda^2, \\ y_3 &= -y - (x_3 - x)\lambda - a_1x_3 - a_3, \end{aligned}$$

  où

$$\lambda = \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}.$$

- **Somme.** Soient $P = (x_1, y_1)$ et $Q = (x_2, y_2)$ deux points de $E(K)$ avec $P \neq Q$.
  1. Si $x_1 = x_2$, alors $P + Q = \mathcal{O}$.
  2. Si $x_1 \neq x_2$, alors $P + Q = (x_3, y_3)$ avec

$$\begin{aligned} x_3 &= -x_1 - x_2 - a_2 + a_1\lambda + \lambda^2, \\ y_3 &= -y_1 - (x_3 - x_1)\lambda - a_1x_3 - a_3. \end{aligned}$$

  où

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}.$$

Certains protocoles cryptographiques peuvent être alors adaptés avec les courbes elliptiques.

2.3.1 Echange de clés de Diffie et Hellman

Si deux personnes A et B veulent échanger une clé, ils peuvent procéder comme suivant.

- A et B s'accordent pour choisir une courbe elliptique E définie sur $\mathbb{F}_p$ et un point $P \in E(K)$ ayant un grand ordre, c'est à dire que le plus petit entier non nul n vérifiant $nP = \mathcal{O}$ est assez grand.
- A choisit un entier k_A avec $2 \leq k_A \leq n-1$ et calcule $Q_A = k_AP = (x_A, y_A)$. A envoie alors Q_A à B.
- B choisit un entier k_B avec $2 \leq k_B \leq n-1$ et calcule $Q_B = k_BP = (x_B, y_B)$. B envoie alors Q_B à A.
- En recevant $Q_B = k_BP = (x_B, y_B)$, A calcule alors k_AQ_B.
- En recevant $Q_A = k_AP = (x_A, y_A)$, B calcule alors k_BQ_A.
- La clé commune est alors $k_AQ_B = k_Ak_BP = k_Bk_AP = k_BQ_A$.

2.3.2 El Gamal avec les courbes elliptiques

Il est très facile d'adapter le cryptosystème El Gamal en utilisant les courbes elliptiques. On suppose qu'une personne A veut envoyer un message à une autre personne B.

- Les personnes A et B s'accordent sur une une courbe elliptique E définie sur $K = \mathbb{F}_p$ et un point $P \in E(K)$ ayant un grand ordre, c'est à dire que le plus petit entier non nul n vérifiant $nP = \mathcal{O}$ est assez grand.
- A choisit un entier k_A avec $2 \leq k_A \leq n-1$ et calcule $Q_A = k_A P = (x_A, y_A)$. La clé publique de A est $k_A P$ et sa clé secrète est k_A.
- B choisit un entier k_B avec $2 \leq k_B \leq n-1$ et calcule $Q_B = k_B P = (x_B, y_B)$. La clé publique de B est $k_B P$ et sa clé secrète est k_B.
- Pour envoyer le message M à B, A doit le transformer en un point m de $E(K)$. En utilisant la clé publique $k_B P$ de B, A calcule ensuite $m + k_A(k_B P)$ et envoie ce message chiffré à B.
- En recevant $m + k_A(k_B P)$, B calcule $m + k_A(k_B P) - k_B(k_A P) = m$ en utlisant la clé publique $k_A P$ de A.

3 Le futur

Actuellement, les cryptosystèmes à clés publiques les plus dominants sont RSA, El Gamal, et dans une moindre mesure ECC. Dans le futur, et dans l'hypothèse d'un développement rapide des ordinateurs quantiques, ces cryptosystèmes sont appelés à être abandonnés. En effet, avec un ordinateur quantique, l'algorithme de Shor [18] permet de résoudre le problème du logarithme discret aussi bien elliptique que sur les corps finis et permet aussi de factoriser un nombre entier en temps polynomial. Apparemment, le cryptosystème NTRU résiste aux ordinateurs quantiques. Il en résulte que NTRU est certainement l'un des plus grands systèmes du futur. D'autres cryptosystèmes qui peuvent jouer un rôle important dans le futur sont basés eux aussi sur les réseaux, en particulier en utilisant le problème du Learning With Errors (LWE) [16], ou encore des cryptosystèmes basés sur les codes comme McEliece [10] et Niederreiter [13], ainsi que le classique Rinjdael (AES). Les cryptosystèmes qui résistent aux ordinateurs quantiques sont appelés "*post quantum cryptosystems*".

3.1 NTRU

Le cryptosystème NTRU à été présenté en 1996 et publié en 1998 [5]. Son domaine de calculs est l'anneau des polynômes $\mathbb{Z}[X]/(X^N - 1)$ où N est un nombre entier. Il se compose de deux protocoles. Le protocole de chiffrement-déchiffrement

NTRUEncrypt et le protocole de signature NTRUSign. NTRU est aujourd'hui considéré comme fiable par le standard IEEE P1363.1 [6].

3.1.1 Définitions

Le cryptosystème NTRU utilise trois nombres entiers, N, p et q. Les opérations de NTRU ont pour domaine l'anneau des polynômes $\mathcal{P}$ avec :

$$\mathcal{P} = \mathbb{Z}[X]/\left(X^N - 1\right).$$

Les polynômes de $\mathcal{P}$ peuvent être représentés sous la forme

$$f = (f_0, f_1, \cdots, f_{N-1}) = \sum_{i=0}^{N-1} f_i X^i.$$

L'addition de deux polynômes $f, g \in \mathcal{P}$ se fait terme à terme de même degrés, alors que la multiplication se fait par un produit de convolution noté $*$. Si $f * g = h$ avec $f = \sum_{i=0}^{N-1} f_i X^i$ et $g = \sum_{i=0}^{N-1} g_i X^i$, alors $h = \sum_{i=0}^{N-1} h_i X^i$ avec pour tout $0 \leq k \leq N-1$,

$$h_k = \sum_{i+j \equiv k \mod N} f_i g_j = \sum_{i=0}^{k} f_i g_{k-i} + \sum_{i=k+1}^{N-1} f_i g_{N+k-i}.$$

Soient $p, q \in \mathcal{P}$ deux polynômes premiers entre eux. Généralement, q est de la forme $q = 2^l$ avec $l = \lfloor \log_2 N \rfloor$ et $p = 2$ ou $p = 3$ ou dans certaines versions $p = 2 + X$. Soit $\mathbb{Z}_q$ l'anneau des entiers modulo q, c'est à dire $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. Dans NTRU, $\mathbb{Z}_q$ est représenté par l'intervalle $\left[-\frac{q}{2}, \frac{q}{2}\right[$. En réduisant $\mathcal{P}$ modulo q, on obtient l'anneau

$$\mathcal{P}_q = \mathbb{Z}_q[X]/(X^N - 1).$$

Ainsi un polynôme $f = (f_0, f_1, \cdots, f_{N-1}) \in \mathcal{P}$ sera représenté par le polynôme

$$\pi_q(f) = (f_0 \pmod q, f_1 \pmod q, \cdots, f_{N-1} \pmod q) \in \mathcal{P}_q.$$

Un polynôme $f \in \mathcal{P}_q$ est dit inversible modulo q s'il existe un polynôme f_q dans $\mathcal{P}_q$ tel que $f * f_q = f_q * f \equiv 1 \pmod q$. De façon similaire, pour un polynôme $p \in \mathcal{P}$, on définit l'anneau $\mathcal{P}_p$ par

$$\mathcal{P}_p = \mathbb{Z}[X]/\left(X^N - 1, p\right),$$

obtenu en réduisant $\mathcal{P}$ modulo p. Si p est un entier, alors $\mathcal{P}_p = \mathbb{Z}_p[X]/\left(X^N - 1\right)$.

3.1.2 Paramètres de NTRU

Soit d un entier positif. On pose

$$\mathcal{B}(d) = \left\{ f \in \mathbb{Z}_2[X]/\left(X^N - 1\right) \mid f = \sum_{i=1}^{d} X^{n_i},\ 0 \leq n_1 < \cdots < n_d \leq N \right\}.$$

Ainsi, les éléments de $\mathcal{B}(d)$ ont exactement d coefficients égaux à 1 et le reste des coefficients est nul. De même, soient d_1 et d_2 deux entiers positifs. On pose

$$\mathcal{T}(d_1, d_2) = \left\{ f \in \mathbb{Z}_3[X]/\left(X^N - 1\right) \mid f = \sum_{i=1}^{d_1} X^{n_i} - \sum_{j=1}^{d_2} X^{m_j},\ n_i \neq m_j \right\}.$$

Autrement dit, $\mathcal{T}(d_1, d_2)$ est l'ensemble des polynômes ayant exactement d_1 coefficients égaux à 1, d_2 coefficients égaux à -1 et le reste des coefficients est nul.

Les paramètres de NTRU sont les suivants.

- Un nombre entier N. Ce nombre doit être premier et assez grand.
- Un nombre entier q, généralement de la forme $q = 2^l$ avec $l = \lfloor \log_2(N) \rfloor$.
- Un paramètre p qui est soit un nombre entier p premier avec q soit un polynôme $p = \alpha + \beta X \in \mathbb{Z}[X]$, inversible modulo q. Généralement $p = 2 + X$ ou $p = 3$.
- Un ensemble $\mathcal{L}_f \subset \mathcal{P}$ de polynômes f.
- Un ensemble $\mathcal{L}_g \subset \mathcal{P}$ de polynômes g.
- Un ensemble $\mathcal{L}_m \subset \mathcal{P}$ de messages secrets.
- Un ensemble $\mathcal{L}_r \subset \mathcal{P}$ de polynômes auxiliaires secrets.

Suivant les versions de NTRU, on rencontrer les paramètres ci-dessus.

Version	p	q	$\mathcal{L}_f$	$\mathcal{L}_g$	$\mathcal{L}_r$	$\mathcal{L}_m$
1998	3	2^k	$\mathcal{T}(d_f, d_f - 1)$	$\mathcal{T}(d_g, d_g)$	$\mathcal{T}(d_r, d_r)$	$\mathcal{T}(d_m, d_m)$
2001	$2 + X$	2^k	$1 + p * F$	$\mathcal{B}(d_g)$	$\mathcal{B}(d_r)$	$\mathcal{B}(d_m)$
2005	3	2^k	$1 + p * F$	$\mathcal{B}(d_g)$	$\mathcal{B}(d_r)$	$\mathcal{B}(d_m)$

Les nombres entiers d_f, d_g, d_r et d_m sont fixés pour chaque version. Actuellement, ces paramètres sont les suivants (voir [5]) :

Sécurité	N	p	q	d_f	d_g	d_r
Moyenne	251	3	128	72	71	72
Haute	347	3	128	64	173	64
Très haute	503	3	256	420	251	170

3.1.3 Utilisation de NTRU

Si une personne B souhaite envoyer un message M à une personne A, alors le processus peut se faire en trois étapes :

1. A doit générer une clé publique h et des clés privées f et f_p.
2. B doit transformer le message clair M en un message chiffré e et envoyer le message chiffré e à A.
3. A doit déchiffrer le message reçu e et retrouver le message clair M.

3.1.4 Générations de clés

Pour commencer, la personne A doit préparer ses paramètres.

Génération des clés :

1. A choisit aléatoirement un polynôme $f \in \mathcal{L}_f$.
2. A calcule l'inverse f_q de f dans $\mathcal{P}_q$, c'est à dire $f * f_q \equiv 1 \pmod{q}$.
3. A calcule l'inverse f_p de f dans $\mathcal{P}_p$, c'est à dire $f * f_p \equiv 1 \pmod{p}$.
4. A choisit aléatoirement un polynôme $g \in \mathcal{L}_g$.
5. A calcule $h \equiv p * g * f_q \pmod{q}$ dans $\mathcal{P}_q$.

La clé publique est h et la clé secrète est (f, f_p).

3.1.5 Chiffrement

Pour chiffrer un message M, la personne B doit le transformer en un polynôme $m \in \mathcal{L}_m$ et ensuite le transformer en un polynôme chiffré e avec la procédure suivante.

Chiffrement :

1. B choisit aléatoirement un polynôme $r \in \mathcal{L}_r$.
2. B calcule $e \equiv r * h + m \pmod{q}$ dans $\mathcal{P}_q$.

Le message chiffré est alors e et peut être envoyé à A.

3.1.6 Déchiffrement

Pour déchiffrer un message $e \in \mathcal{P}_q$, la personne A doit utiliser ses clés secrètes f et f_p.

Déchiffrement :

1. A calcule $a \equiv f * e \pmod{q}$ dans $\mathcal{P}_q$ avec des coefficients dans l'intervalle $\left[-\frac{q}{2}, \frac{q}{2}\right[$.
2. A calcule $m \equiv f_p * a \pmod{p}$ dans $\mathcal{P}_p$.

L'exactitude du processus est dans la propriété suivante.

$$\begin{aligned} a &\equiv f * e \pmod{q} \\ &\equiv f * (r * h + m) \pmod{q} \\ &\equiv f * r * (p * g * f_q) + f * m \pmod{q} \\ &\equiv p * r * g * f * f_q + f * m \pmod{q} \\ &\equiv p * r * g + f * m \pmod{q}. \end{aligned}$$

Alors

$$f_p * a \equiv f_p * (p * r * g + f * m) \equiv f_p * p * r * g + f_p * f * m \equiv m \pmod{p}.$$

En faite, avec une faible probabilité, on peut obtenir un message différent de l'original, mais ce défaut fut finalement corrigé en adaptant les paramètres de départ.

3.1.7 La sécurité de NTRU

La sécurité de NTRU est basée sur deux problèmes difficiles, SVP et CVP.

Soit m un entier positif. On considère l'ensemble $\mathbb{R}^m$ des vecteurs $u = (u_1, \cdots, u_m)$, muni du produit scalaire et de la norme euclidienne.

Définition 3.1. *Soient $u = (u_1, \cdots, u_m)$ et $v = (v_1, \cdots, v_m)$ deux vecteurs de $\mathbb{R}^m$. Le produit scalaire de u et v est*

$$\langle u, v \rangle = \sum_{i=1}^{m} u_i v_i.$$

Définition 3.2. *Soit $u = (u_1, \cdots, u_m)$ un vecteur de $\mathbb{R}^m$. La norme euclidienne de u est*

$$||u|| = \sqrt{\langle u, u \rangle} = \sqrt{\sum_{i=1}^{m} u_i^2}$$

Soit n un entier positif avec $n \leq m$. Soient $(b_1, \cdots, b_n)$ des vecteurs linéairement indépendants de $\mathbb{R}^m$. Soit B la matrice dont les colonnes sont composées des coordonnées des vecteurs $b_1, \cdots, b_n$.

Définition 3.3. *Le réseau engendré par la famille $(b_1, \cdots, b_n)$ (ou par B) est l'ensemble*

$$L = \sum_{i=1}^{n} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{n} x_i b_i \mid x_i \in \mathbb{Z} \right\}.$$

L'entier n est la dimension de L.

Un réseau L admet plusieurs bases. Le passage entre deux bases se fait à l'aide d'une matrice carrée à coefficients dans $\mathbb{Z}$, de déterminant ± 1. Parmi toutes les bases possibles, certaines ont de meilleures propriétés que d'autres. La recherche d'une bonne base est un problème NP-complet. En 1982, Lenstra, Lenstra et Lovasz [9] ont proposé l'algorithme LLL qui détermine une base avec de très bonnes propriétés. L'algorithme LLL est très efficace puisque sa complexité est polynomiale. On remarque en particulier que l'algorithme LLL produit des bases avec des vecteurs assez courts, ce qui peut apporter une réponse partiellement satisfaisante aux deux problèmes (NP-durs) sur lesquels repose la sécurité de NTRU.

Problème SVP, Shortest Vector Problem (problème du plus court vecteur) : *Etant donné une base B d'un réseau L, trouver un vecteur non nul de L le plus court possible pour la norme euclidienne.*

Problème CVP, Closest Vector Problem (problème du plus proche vecteur) : *Etant donné une base B d'un réseau L et un vecteur v_0, trouver un vecteur de $v \in L$ le plus proche possible de v_0 pour la norme euclidienne.*

3.2 Learning With Errors (LWE)

En 2005, Regev [16] a introduit un problème appelé "Learning With Errors" (LWE). Il s'agit de résoudre un système de presque-équations.

Soient p un nombre premier. On pose $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$. Soit $n \geq 1$ un nombre entier. Le produit de deux vecteurs $s = (s_1, \cdots, s_n) \in \mathbb{Z}_p^n$ et $a_i = ((a_i)_1, \cdots, (a_i)_n) \in \mathbb{Z}_p^n$ est par définition :

$$\langle s, a_i \rangle = \sum_{j=1}^{n} s_j \, (a_i)_j \, .$$

Soit χ une distribution de probabilité $\chi \ : \ \mathbb{Z}_p \to \mathbb{R}^+$ sur $\mathbb{Z}_p$. Soit m un entier et $e_1, \cdots, e_m$ des valeurs de $\mathbb{Z}_p$ choisis indépendamment selon χ. On considère le

système de m équation avec erreurs d'inconnues s

$$\begin{aligned} \langle s, a_1 \rangle + e_1 &= b_1, \\ \langle s, a_2 \rangle + e_2 &= b_2, \\ \vdots &= \vdots \\ \langle s, a_m \rangle + e_m &= b_m. \end{aligned}$$

Le problème LWE est de déterminer s connaissant les vecteurs $(a_1, \cdots, a_m) \in \left(\mathbb{Z}_p^n\right)^m$ et $(b_1, \cdots, b_m) \in \mathbb{Z}_p^m$. Regev a montré que ce problème est difficile et peut donc servir à la construction de cryptosystèmes.

Un cryptosystème à base de LWE (Regev 2005)

- **Paramètres :** Choisir un nombre premier p, deux paramètres m et n, et une distribution de probablité χ.
- **La clé privée :** Choisir aléatoirement un clé privée $s \in \mathbb{Z}_p^n$.
- **La clé publique :**
  1. Choisir m vecteurs $a_i \in \mathbb{Z}_p^n$, $i = 1, \cdots, m$, de façon indépendante.
  2. Choisir des entiers $e_1, \cdots, e_m \in \mathbb{Z}$ de façon indépendante relativement à χ.
  3. Pour $i = 1, \cdots, m$, calculer $b_i = \langle s, a_i \rangle + e_i$.
  4. Former la clé publique $((a_1, b_1), \cdots, (a_m, b_m))$.
- **Chiffrement :** Le chiffrement se fait bit par bit.
  1. Choisir aléatoirement un sous-ensemble S parmi les 2^m sous-ensembles de l'intervalle $[1, m]$.
  2. Si le bit est 0, calculer $\left(\sum_{i \in S} a_i \pmod p, \sum_{i \in S} b_i \pmod p\right)$.
  3. Si le bit est 1, calculer $\left(\sum_{i \in S} a_i \pmod p, \left\lfloor \frac{p}{2} \right\rfloor + \sum_{i \in S} b_i \pmod p\right)$ où $\lfloor x \rfloor$ désigne la partie entière de x.
- **Déchiffrement :** Pour déchiffrer une paire $(a, b) \in \mathbb{Z}_p^n \times \mathbb{Z}$.
  1. Calculer $c \equiv b - \langle s, a \rangle \pmod p$.
  2. Si c est plus proche de 0 que de $\left\lfloor \frac{p}{2} \right\rfloor$, alors le bit est 0, sinon le bit est 1.

Regev a montré que casser ce cryptosystème revient à résoudre le problème SVP à l'aide d'un algorithme quantique à temps polynomial.

4 Conclusion

Un certain nombre de cryptosystèmes basés sur des problèmes issus de la théorie des réseaux commencent à émerger. Ces cryptosystèmes sont appelés à jouer

un rôle de plus en plus déterminant dans le futur et la cryptographie telle qu'elle est pratiquée aujourd'hui et qui est basée sur RSA et El Gamal sera dans l'obligation de céder la place. En effet, la cryptographie du présent ne résistera pas aux ordinateurs quantiques contrairement aux nouveaux cryptosystèmes émergents tels que NTRU et les cryptosystèmes basés sur LWE.

Références

[1] H. Cohen, A Course in Computational Number Theory, Graduate Texts in Mathematics, Springer, 1993.

[2] D. Coppersmith and A. Shamir, Lattice attacks on NTRU. In Advances in cryptology—EUROCRYPT '97, volume 1233 of Lecture Notes in Comput. Sci., pages 52–61. Springer, Berlin, 1997.

[3] W. Diffie, E. Hellman, New Directions in Cryptography, IEEE Trans- actions on Information Theory, 22, 5 (1976), pp. 644–654.

[4] T. El Gamal, A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Transactions on Information Theory IT-31, 496-473,1976.

[5] J. Hoffstein, J. Pipher, and J. H. Silverman, NTRU : A Ring Based Public Key Cryptosystem in Algorithmic Number Theory. Lecture Notes in Computer Science 1423, Springer-Verlag, pages 267–288, 1998.

[6] IEEE P1363.1 Public-Key Cryptographic Techniques Based on Hard Problems over Lattices, June 2003. IEEE.

[7] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation 48, 1987, pp. 203–209

[8] N. Koblitz, A Course in Number Theory and Cryptography, Springer-Verlag, New York, 1994.

[9] A.K. Lenstra, H.W. Lenstra and L. Lovasz, Factoring polynomials with rational coefficients, Mathematische Annalen, Vol. 261, 513—534, 1982.

[10] McEliece, R.J. : A public-key cryptosystem based on algebraic coding theory. The Deep Space Network Progress Report 42–44, Jet Propulsion Laboratory, Pasadena, CA, (1978), 114–116.

[11] A. Menezes, Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston, 1993.

[12] V.S. Miller, Use of elliptic curves in cryptography, CRYPTO 85, 1985.

[13] H. Niederreiter. Knapsack-Type Cryptosystems and Algebraic Coding Theory. Problems of Control and Information Theory (Problemy Upravlenija i Teorii Informacii) 15, 1986, pages 159–166.

[14] NTRU Inc. http://www.ntru.com/

[15] NTRU Inc. http://www.ntru.com/cryptolab/faqs.htm#six

[16] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, STOC 2005, ACM (2005) p. 84–93.

[17] R. Rivest, A. Shamir, L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, Vol. 21 (2), 120—126 (1978)

[18] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Computing 26, pp. 1484-1509 (1997).