LE CRYPTOSYSTÈME NTRU

Abderrahmane Nitaj

Janvier 2003

Abstract. NTRU est nouveau cryptosystème à clé publique dont la sécurité n'est basée ni sur difficulté de la factorisation des entiers, ni sur celle de la résolution du logarithme discret mais sur la difficulté de trouver un vecteur court dans un réseau de grande dimension. Dans ce rapport, on présente ce cryptosystème et les différentes attaques pour le casser.

Contents

1991 *Mathematics Subject Classification.* 94A60, 11T06.
Key words and phrases. Cryptosystèmes, NTRU, Réseaux.

1. Introduction.

J. Hoffstein, J. Pipher et J.H. Silverman [3] de l'université de Brown (USA) ont proposé en 1996 un cryptosystème qu'ils ont appelé NTRU, mais ce n'est qu'en 2000 qu'ils ont déposé leur découverte et ont reçu des investissements (38 millions de \$ de la part de Sony, de TI, Greylock,....). Ils ont choisi le nom NTRU en hommage a leur groupe de travail "Number Theory Research Unit", mais comme préfère le dire Silverman, cela rime avec "Number Theorists R Us".

Ce cryptosystème permet de crypter, décrypter, authentifier, signer et engendrer des clés.

Chaque système utilisé en cryptographie est basé sur la difficulté de résoudre un problème. Ainsi, RSA [8] est lié à la factorisation des nombres, et El Gamal [2] est lié au problème du logarithme discret. Le nouveau cryptosystème NTRU est basé sur la difficulté de trouver le plus court vecteur d'un réseau de grande dimension.

Les cryptosystèmes basés sur des variations du problème du Sac à Dos ont été cassés à l'aide des techniques de réduction des réseaux. Contrairement à ces cryptosystèmes qui "craignent la réduction des réseaux", le nouveau système de cryptographie NTRU est basé sur l'idée que la détermination d'un vecteur très courts dans un réseau n'est pas facile.

Les opérations nécessaires pour la fabrication d'un cryptosystème basé sur NTRU sont beaucoup plus simples que celles utilisées pour d'autres systèmes tels RSA, El Gamal et ECC. Ces derniers utilisent en fait l'exponentiation, qui est très couteuse en opérations, alors que NTRU n'en nécessite aucune.

2. Les produits de convolution.

Soit N un entier. On pose $R = \mathbb{Z}[X]/(X^N - 1)$. Soient F et G deux éléments de R définis par :

$$F(X) = \sum_{i=0}^{N-1} F_i X^i, \qquad G(X) = \sum_{i=0}^{N-1} G_i X^i.$$

Soient α et β deux entiers. La somme de αF et βG est défini par

$$\alpha F + \beta G = \sum_{i=0}^{N-1} (\alpha F_i + \beta G_i) X^i.$$

Le produit de F et G dans R est défini par

$$F * G = H = \sum_{i=0}^{N-1} H_i X^i$$

avec

$$H_i = \sum_{k=0}^{i} F_k G_{i-k} + \sum_{k=i+1}^{N-1} F_k G_{N+i-k} = \sum_{k+l \equiv i \pmod{N}} F_k G_l.$$

Ceci peut s'écrire aussi :

$$\begin{pmatrix} H_0 \\ H_1 \\ \vdots \\ H_{N-1} \end{pmatrix} = \begin{pmatrix} F_0 & F_1 & \cdots & F_{N-1} \\ F_1 & F_2 & \cdots & F_0 \\ \vdots & \vdots & \ddots & \vdots \\ F_{N-1} & F_0 & \cdots & F_{N-2} \end{pmatrix} \begin{pmatrix} G_0 \\ G_{N-1} \\ \vdots \\ G_1 \end{pmatrix}$$

Le produit de convolution $*$ vérifie, en particulier, pour des polynômes F, G et U et des entiers α et β :

$$\begin{aligned} F * G &= G * F, \\ F * (G * U) &= (F * G) * U, \\ F * (\alpha G + \beta U) &= \alpha F * G + \beta F * U, \end{aligned}$$

Si p est un entier, on convient de poser $F \pmod p$ avec la convention

$$F \pmod p = \sum_{i=0}^{N-1} (F_i \pmod p) X^i.$$

Le calcul direct de $F*G$ a une complexité en $O(N^2)$. Cependant, la transformée de Fourier rapide (FFT) requiert uniquement une complexité en $O(N \log N)$ lorsque N est une puissance de 2.

3. Description de NTRU.

3.1. Le problème.

Brigitte veut envoyer un message m sous forme crypté e à Ali.

Pour cela, Brigitte doit utiliser les paramètres de NTRU et la clé publique de Ali. Ali et Brigitte vont utiliser le système NTRU. Ils doivent utiliser les paramètres de NTRU composés de trois nombres entiers N, p et q, non nécessairement premiers, avec $(p, q) = 1$ et q est très grand par rapport à p et quatre sous-ensembles $\mathbb{L}_m$, $\mathbb{L}_f$, $\mathbb{L}_g$ et $\mathbb{L}_\phi$ de $\mathbb{Z}[X]/(X^N - 1)$.

3.2. Ali fabrique sa clé publique.

Ali choisit deux polynômes secrets $f \in \mathbb{L}_f$ et $g \in \mathbb{L}_g$ et calcule les inverses f_p^{-1} et f_q^{-1} de f respectivement dans $\mathbb{Z}_p[X]/(X^N - 1)$ et $\mathbb{Z}_q[X]/(X^N - 1)$. Ainsi f_q^{-1} et f_p^{-1} sont deux polynômes qui vérifient :

$$\begin{aligned} f_p^{-1} * f &\equiv 1 \pmod p \\ f_q^{-1} * f &\equiv 1 \pmod q. \end{aligned}$$

Ali calcule ensuite le polynôme :

$$h \equiv f_q^{-1} * g \pmod q.$$

La clé publique d'Ali est alors h. Ses clés privées sont f et g (et donc aussi f_p^{-1} et f_q^{-1}).

3.3. Brigitte envoie un message.

Brigitte prend un message clair et le transforme de tel sorte que $m \in \mathbb{L}_m$ et choisit un polynôme $\phi \in \mathbb{L}_\phi$ et transforme le message clair m en un message crypté e :

$$e \equiv p\phi * h + m \pmod q.$$

Alors Brigitte envoie le message e à Ali et détruit son polynôme ϕ.

3.4. Ali décrypte le message de Brigitte.

En recevant le message secret e de Brigitte, Ali utilise sa clé secrète f et calcule dans $\mathbb{Z}[X]/(X^N - 1)$:

$$a \equiv f * e \pmod q,$$

en s'arrageant pour que les coefficients de a soient dans l'intervalle $[-q/2, q/2]$.

S'ils sont tous dans $]-q/2, q/2[$ alors il calcule :

$$b = f_p^{-1} * a.$$

en s'arrageant pour que les coefficients de b soient dans l'intervalle $[-p/2, p/2]$. Alors Ali sait maintenant que le message clair de Brigitte est $b = m$.

3.5. Explications.

Si les coefficients de a sont tous dans $]-q/2, q/2[$, alors il vérifie :

$$\begin{aligned} a \equiv f * e &\equiv f * (p\phi * h + m) \pmod q \\ &\equiv p\phi * f * h + f * m \pmod q \\ &\equiv p\phi * f * (f_q^{-1} * g) + f * m \pmod q \\ &\equiv p\phi * g + f * m \pmod q. \end{aligned}$$

De même, le polynôme b vérifie :

$$\begin{aligned} b \equiv f_p^{-1} * a &\equiv f_p^{-1} * (p\phi * g + f * m) \pmod p \\ &\equiv f_p^{-1} * f * m \pmod p \\ &\equiv m \pmod p. \end{aligned}$$

3.6 Conditions pour le décryptage.

Le décryptage peut ne pas fonctionner dans certains cas. Ceci est du au fait qu'une équivalence modulo q peut être fausse modulo p et inversement. Un exemple numérique est :

$$1 \equiv 17 \pmod{16}, \text{ mais } 1 \not\equiv 17 \pmod 5.$$

Soient u et v deux entiers. Si $u \equiv v \pmod q$ et $u \equiv v \pmod p$, alors $u - v \equiv 0 \pmod q$ et $u - v \equiv 0 \pmod p$. Comme $(p, q) = 1$, alors $u - v \equiv 0 \pmod{pq}$. Ainsi Pour que l'équivalence $u \equiv v \pmod q$ se transforme en $u \equiv v \pmod p$, il faut que $u - v$ soit un multiple de p et un multiple de q.

Supposons maintenant que u et v sont dans l'intervalle $[-q/2, q/2]$ et que $u \equiv v \pmod q$. Alors la valeur exacte de $u - v$ est :

$$u - v = \begin{cases} -q & \text{si} \quad u = -v = -\frac{q}{2}, \\ q & \text{si} \quad u = -v = \frac{q}{2}, \\ 0 & \text{sinon,} \end{cases}$$

Ainsi, pour que $u \equiv v \pmod p$, il faut que u et $-v$ ne soient pas égaux à $\pm q/2$.

Dans le procéssus de décryptage, on a

$$a \equiv p\phi * g + f * m \pmod q.$$

Pour que cette équivalence se transmette modulo p, il faut que tous les coefficients de a coincident avec les coefficients dans le calcul exact (dans $\mathbb{Z}[X]/(X^N - 1)$) de $a_0 = p\phi * g + f * m$, en excluant $\pm q/2$. Alors dans ce cas, b est le message clair m.

Comme ni Ali ni Brigitte ne connaissent f, g et ϕ en même temps, alors le polynôme a_0 ne peut être explicitement calculé. On peut cependant donner condition pour que le décryptage aboutisse.

On pose

$$a_0 = \sum_{i=0}^{N-1} \alpha_i X^i,$$

et

$$\bar{a_0} = \frac{1}{N} \sum_{i=0}^{N-1} \alpha_i, \quad |a_0|_\infty = \max_i \alpha_i - \min_i \alpha_i, \quad |a_0|_{2'} = \left(\sum_{i=0}^{N-1} \alpha_i^2 - N\bar{a_0}^2 \right)^{1/2}.$$

Pour le décryptage donne le message m, il faut $|a_0|_\infty < q$, donc $|p\phi * g + f * m|_\infty < q$. Comme

$$|p\phi * g + f * m|_\infty \leq |p\phi * g|_\infty + |f * m|_\infty,$$

on peut se contenter de prendre

$$|p\phi * g|_\infty \leq \frac{q}{4} \text{ et } |f * m|_\infty \leq \frac{q}{4}.$$

Or, il existe [3] deux constantes γ_1 et γ_2 dépendant de N tels que pour deux polynômes F et G on ait:

$$\gamma_1 |F|_{2'} |G|_{2'} \leq |F * G|_\infty \leq \gamma_2 |F|_{2'} |G|_{2'}.$$

Ainsi, on obtient :

$$|\phi|_{2'} |g|_{2'} \leq \frac{q}{4p\gamma_2}, \qquad |f|_{2'} |m|_{2'} \leq \frac{q}{4\gamma_2}.$$

Comme

$$|\phi|_{2'} = \sqrt{2d_\phi}, \quad |g|_{2'} = \sqrt{2d_g}, \quad |f|_{2'} = \sqrt{2d_f - 1 - N^{-1}}, \quad |m|_{2'} \leq \frac{p}{2}\sqrt{N},$$

alors on peut exprimer d_f, d_g et d_ϕ en fonction de p, q et N. Ceci permet à NTRU de choisir les paramètres d_f, d_g et $d_p hi$.

4. Choix des paramètres par NTRU.

Soient d_1 et d_2 deux nombres entiers. On pose

$$\mathbb{L}(d_1, d_2) = \{F \in \mathbb{Z}[X]/(X^N - 1),\ F \text{ a} \begin{cases} d_1 & \text{coefficients } 1, \\ d_2 & \text{coefficients } -1, \\ N-(d_1+d_2) & \text{coefficients } 0, \end{cases} \}$$

Pour que le décryptage se passe bien, NTRU [3] suggère les paramètres suivants:

$$\mathbb{L}_m = \mathbb{Z}_p[X]/(X^N - 1),$$

et pour des entiers d_f, d_g et d_ϕ,

$$\begin{aligned} \mathbb{L}_f &= \mathbb{L}(d_f, d_f - 1), \\ \mathbb{L}_g &= \mathbb{L}(d_g, d_g), \\ \mathbb{L}_\phi &= \mathbb{L}(d_\phi, d_\phi), \end{aligned}$$

La sécurité du système NTRU dépend non seulement de N, p et q mais aussi des entiers d_f, d_g et d_ϕ. NTRU [3] préconise les valeurs suivantes, qui ont été choisies expérimentalement de telle sorte que la probabilité pour que le message m de Brigitte ne soit décodable par Ali est inférieure à 5.10^{-5} :

Sécurité	N	p, q	d_f	d_g	d_ϕ
Modérée	107	$3, 64$	15	12	5
Haute	167	$3, 128$	61	20	18
Très Haute	503	$3, 256$	216	72	55

5. Les attaques

5.1. Attaques forcées contre la clé.

La clé publique f d'Ali peut être découverte en essayant tous les polynômes F de $\mathbb{L}_f$ et en regardant si $F * h \pmod q$ a des coefficents assez petits. En effet si

$$F * h \equiv F * f_q^{-1} * g \pmod q,$$

a des coefficents assez petits, on peut recouvrir f en prenant $F^{-1} \pmod q$. Comme chaque polynôme de $\mathbb{L}_f$ a d_f termes égaux à 1, et $d_f - 1$ termes égaux à -1, alors le nombre de choix possible pour F est

$$\#\mathbb{L}_f = \binom{N}{d_f}\binom{N - d_f}{d_f - 1}.$$

De même la clé publique d'Ali peut être recouverte en essayant tous les polynomes G de $\mathbb{L}_g$ et en regardant si $G * h^{-1} \pmod q$ a des coefficents assez petits. Le nombre de choix possible pour les polynomes G est alors :

$$\#\mathbb{L}_g = \binom{N}{d_g}\binom{N - d_g}{d_g}.$$

5.2. Attaques forcées contre le message.
Le message clair de Brigitte vérifie $m = e - p\phi * h \pmod q$. L'attaquant connait e, p et h. Il peut donc tester tous les polynômes Φ de $\mathbb{L}_\phi$ et regarder si $e - p\Phi * h \pmod q$ a des coefficents assez petits. Le nombre de choix possible pour Φ est alors :

$$\#\mathbb{L}_\phi = \binom{N}{d_\phi}\binom{N-d_\phi}{d_\phi}.$$

Les paramètres d_f, d_g et d_ϕ de NTRU vérifient $d_\phi < d_g < d_f$, le tableau suivant donne un aperçu de la taille de $\#\mathbb{L}_p hi$:

Sécurité	N	d_ϕ	$\#\mathbb{L}_\phi$
Modéré	107	5	$0.6.10^{26}$
Haute	167	18	$0.6.10^{44}$
Très Haute	503	55	$0.3.10^{100}$

5.3. Attaques forcées avec rencontre au milieu contre la clé.
La clé publique d'Ali f peut être découverte en essayant tous les polynômes F_1 et F_2 dans $\mathbb{L}_f$ tels que $F_1 * h \pmod q$ et $-F_2 * h \pmod q$ ont des coefficients voisins. Ainsi le polynôme $F = F_1 + F_2$ et tel que $F * h$ a de petits coefficients et on se trouve donc dans le cas d'une attaque forcée, décrite ci-dessus. Avec cette méthode [12], le nombre de choix des polynômes F_1 et F_2 se limite à $\sqrt{\#\mathbb{L}_g}$.

5.4. Attaques forcées avec rencontre au milieu contre le message.
De façon similaire, l'attaquant peut chercher des polynômes Phi_1 et Phi_2 tels que $e - p\Phi_1 * h \pmod q$ et $-(e - p\Phi_2 * h) \pmod q$ ont des coefficients voisins. En prenant alors $\Phi = \Phi_1 + \Phi_2$, on se trouve dans le cas d'une attque forcée. Le nombre de choix des polynômes Φ_1 et Φ_2 se limite alors [12] à $\sqrt{\#\mathbb{L}_\phi}$.

5.5. Attaques en cas de multiples transmissions.
Si Brigitte envoie le même message m sous formes cryptées e_i, $i = 1, \cdots, r$, avec la même clé h, mais avec des polynômes différents ϕ_1, ...,ϕ_r, alors un attaquant peut facilement recouvrir le message m. Comme

$$e_i \equiv p\phi_i * h + m \pmod q, \quad , i = 1, \cdots, r,$$

alors :

$$\phi_i - \phi_1 = (e_i - e_1) * p^{-1} h^{-1} \pmod q, i = 2, \cdots, r.$$

Comme chaque coefficient de $\phi_i - \phi_1$ est dans $\{-2, -1, 0, 1, 2\}$, alors il recouvre l'expression exacte de $\phi_i - \phi_1$. Posons :

$$\phi_i = \sum_{k=0}^{N-1} \alpha_{i,k} X^k,$$

$$\phi_i - \phi_1 = \sum_{k=0}^{N-1} \beta_{i,k} X^k.$$

Alors pour tout $i = 2, \cdots, r$ et tout $k = 0, \cdots N$, on a

$$\beta_{i,k} = \alpha_{i,k} - \alpha_{1,k}.$$

Si $\beta_{i,k} = -2$, alors $\alpha_{i,k} = -1$ et $\alpha_{1,k} = 1$. Si $\beta_{i,k} = 2$, alors $\alpha_{i,k} = 1$ et $\alpha_{1,k} = -1$.

Comme la probabilité que $\beta_{i,k} = -2, 2$ est $2/9$, alors ceci permet donc de trouver $2/9$ des coefficients de ϕ_1. En prenant tous les polynômes ϕ_i, l'attaquant peut ainsi recouvrir une grande partie de ϕ_1, et à l'aide d'une attaque forcée, il peut trouver les coefficients manquants et compléter ϕ_1 et donc exprimer le message clair m de Brigitte :

$$m \equiv e_1 - p\phi_1 * h \pmod q.$$

Une façon de rendre la transmission sûre du même message m en l'envoyant r fois avec la même clé h est la suivante. Brigitte prend $2r$ polynômes ϕ_i et Φ_i, $i = 1, \cdots, r$ dans $\mathbb{L}_\phi$ et envoie les r couples (e_i, E_i) avec :

$$e_i = p\phi_i * h + \Phi_i, \qquad E_i = p\Phi_i * e_i + m.$$

Alors Ali commence par calculer Φ_i par sa clé secrète f à partir de e_i, puis recouvrir m à partir de E_i.

6. Attaques basées sur les réseaux

6.1. Attaques contre la clé publique.

La clé publique d'Ali est calculée avec la formule $h \equiv f_q^{-1} * g \mod q$ et donc $f * h \equiv g \pmod q$.

Soit α un nombre réel assez petit. Soient u et v deux éléments de $\mathbb{Z}_q[X]/(X^N - 1)$ avec $u = \sum_{i=0}^{N-1} u_i X^i$ et $v = \sum_{i=0}^{N-1} v_i X^i$. On considère l'ensemble

$$L_h = \{(\alpha u_0, \cdots \alpha u_{N-1}, v_0, \ldots, v_{N-1}) = (\alpha u, v), \quad u * h = v \pmod q)\}.$$

Cette ensemble définit en fait un réseau de $\mathbb{R}^{2N}$ de dimension $2N$. D'autre part, le couple $(\alpha f, g)$ est un élément de L_h. Les normes euclidiennes de αf et g sont:

$$||\alpha f||_2 = |\alpha|\sqrt{2d_f - 1}, \quad ||g||_2 = \sqrt{2d_g},$$

et donc la norme de $(\alpha f, g)$ est

$$||(\alpha f, g)||_2 = \sqrt{(2d_f - 1)\alpha^2 + 2d_g},$$

qui peut être rendue très petite relativement à la dimension $2N$ de L_h. Ceci signifie que le couple $(\alpha f, g)$ est un vecteur de petite longueur dans L_h. Si on peut donc trouver une base de L_h et des vecteurs assez courts, on peut alors déterminer les clés secrètes f et g par identification.

Soit $(\alpha u, v)$ un élément de L_h. Alors $u * h \equiv v \pmod q$ et donc il existe un vecteur F de $\mathbb{Z}_q[X]/(X^N - 1)$ tel que $v = u * h + qF$, ce qui s'écrit :

$$(v_0, v_1, \cdots, v_{N-1}) = (u_0, u_1, \cdots, u_{N-1}) \begin{pmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{pmatrix}$$
$$+ q\,(F_0, F_1, \cdots, F_{N-1})$$

On peut donc écrire

$$(\alpha u, v) = (u, F) M_\alpha,$$

où M_α est la matrice de Coppersmith et Shamir [1] :

$$\left(\begin{array}{cccc|cccc} \alpha & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & \alpha & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right)$$

Les colonnes $(b_0, b_1, \cdots, b_{2N-1})$ de cette matrice forment une base de L_h.

On peut donc chercher le vecteur le plus court $(\alpha u, v)$ et en déduire les clés secrètes f et g. Le vecteur le plus court d'un réseaux de dimension n et de déterminant D est de la forme

$$\beta D^{1/n} \sqrt{\frac{n}{2\pi e}}, \quad \beta \in \left[1, \sqrt{2}\right].$$

Avec $n = 2N$ et $D = \alpha^N q^N$, cette valeur prend la forme :

$$\beta \sqrt{\frac{N \alpha q}{\pi e}} = \beta s.$$

On peut alors, en variant α, chercher un vecteur $(\alpha u, v)$ de norme minimal par rapport à s. Comme on veut en plus $u = f$ et $v = g$, alors

$$||(\alpha u, v)||_2 = ||(\alpha f, g)||_2 = \sqrt{\alpha^2 ||f||_2^2 + ||g||_2^2} = \sqrt{\alpha^2 (2d_f - 1) + 2d_g}.$$

Rappelons que d_f et d_g sont publiques et que α est une valeur variable. Le problème se réduit donc à maximiser la valeur

$$\frac{s}{||(\alpha u, v)||_2} = \frac{s}{\sqrt{\alpha^2 ||f||_2^2 + ||g||_2^2}},$$

ce qui revient en élevant au carré et en retirant les valeurs fixes N et q et la constante πe à maximiser la valeur

$$\frac{\alpha}{\alpha^2 ||f||_2^2 + ||g||_2^2}.$$

la valeur maximale est alors atteinte avec $\alpha = ||g||_2/||f||_2$. Une autre façon d'aborder ce problème est de minimiser la valeur

$$c_h = \frac{||(\alpha u, v)||_2}{s}.$$

Avec $u = f$, $v = g$ et $\alpha = ||g||_2/||f||_2$, on obtient :

$$c_h = \sqrt{\frac{2\pi e ||f||_2 ||g||_2}{Nq}}.$$

Ainsi, si c_h est très petit, l'algorithme LLL peut donc aboutir rapidement aux clés f et g. NTRU a donc intérêt à choisir d_f et d_g pour que c_h soit assez grand. Numériquement, pour le cas $N = 503$ (haute sécurité), on a $c_h = 0,18$. Ceci n'est pas très grand mais la dimension du réseau est $2N = 1006$. NTRU estime qu'un ordinateur avec un Pentium 200 Mhz doit fonctionner pendant $6,2.10^{27}$ années pour déterminer le plus court vecteur.

6.2. Attaques contre le message.

Le message codé e de Brigitte est calculé avec la formule $e \equiv p\phi * h + m \pmod q$, et donc $e - m \equiv p\phi * h \pmod q$. Pour un réel α, on considère maintenant le réseau de dimension $2N$:

$$L_e = \{(\alpha u_0, \cdots \alpha u_{N-1}, v_0, \ldots, v_{N-1}) = (\alpha u, v), \quad e - m \equiv p\phi * h \pmod q\}.$$

Ainsi $(\alpha m, \phi)$ est un élément de L_e et comme $||m||_2$ et $||\phi||_2 = \sqrt{2d_p hi}$ sont petits relativement à N, la détermination d'un vecteur de L_e de petite taille peut aboutir à la découverte de m et ϕ. De façon similaire, la valeur $\alpha = ||\phi||_2/||m||_2$ maximise les chances de découverte de m et ϕ. Ceci aboutit à une valeur c_e analogue de c_h :

$$c_e = \sqrt{\frac{2\pi e ||m||_2 ||\phi||_2}{Nq}}.$$

Ainsi un message est vulnérable si la valeur de c_e est très petite. Comme les coefficients de m sont compris entre $-p/2$ et $p/2$, alors

$$||m||_2 \leq \frac{1}{2} p\sqrt{N},$$

et donc c_e dépend essentiellement de d_ϕ. Alors NTRU doit choisir d_ϕ avec c_e assez grand.

NTRU a expérimenté la version Schnorr-Euchner de LLL [9] avec une machine 200 Mhz et a conclu que le temps nécessaire pour recouvrir le message m ou les clés secrètes f et g est au moins exponentiel en N. Le temps nécessaire est de 9 jours dans le cas Modéré, 380 années dans le cas Haute Sécurité et $6.2.10^{27}$ années dans le cas Très Haute Sécurité.

7. La signature NSS.

7.1. Description.

La signature NSS (NTRU Signature Scheme) fut proposée en 2001 pour authentifier les messages. Tout comme le système NTRU, elle repose sur un certain nombre de paramètres.

- N est un nombre entier fixé. Les opérations ont lieu dans $\mathbb{Z}[X]/(X^N-1)$, muni du produit de convolution.
- p est un entier non nécessairement premier.
- q est un entier vérifiant $(p,q)=1$, q est très grand par rapport à p.
- $D_{\min}$ est une borne inférieure fixée.
- $D_{\max}$ est une borne supérieur fixée.
- $L_f = L(d_f, d_f)$ est l'ensemble des clés privées, où d_f est un entier donné.
- $L_f = L(d_f, d_f)$ est l'ensemble des clés privées, où d_f est un entier donné.
- $L_g = L(d_g, d_g)$ est l'ensemble des clés privées, où d_g est un entier donné.
- $L_{w_2} = L(d_{w_2}, d_{w_2})$ est l'ensemble de polynômes intervenant dans la création de la signature, où d_w est un entier donné.
- $L_m = L(d_m, d_m)$ est l'ensemble des messages, où d_m est un entier donné.
- f_0 et g_0 sont deux polynômes fixés dans $\mathbb{Z}[X]/(X^N-1)$.
- Dev est une fonction définie de la façon suivante:

Si F et G sont deux polynômes de $\mathbb{Z}[X]/(X^N-1)$, on pose $\bar{F} \equiv F \pmod q \pmod p$, et $\bar{G} \equiv G \pmod q \pmod p$, avec des coefficients compris dans $]-q/2, q/2]$ et $]-p/2, p/2]$. Alors, en posant $\bar{F} = \sum_{i=0}^{N-1} F_i X^i$ et $\bar{G} = \sum_{i=0}^{N-1} G_i X^i$, on a

$$\mathrm{Dev}(F,G) = \#\{i : F_i \neq G_i\}.$$

Problème : Brigitte veut envoyer un message m et une signature s à Ali.

7.1.1. Brigitte fabrique sa clé de vérification publique NSS.

Brigitte choisit deux polynômes secrets $f_1 \in \mathbb{L}_f$ et $g_1 \in \mathbb{L}_g$ et calcule

$$f = f_0 + pf_1, \quad g = g_0 + pg_1.$$

Elle calcule ensuite l'inverse f_q^{-1} de f dans $\mathbb{Z}_q[X]/(X^N-1)$. Ainsi f_q^{-1} est un polynôme qui vérifie :

$$f_q^{-1} * f \equiv 1 \pmod q.$$

Brigitte calcule ensuite le polynôme :

$$h \equiv f_q^{-1} * g \pmod q.$$

La clé de vérification publique de Brigitte est alors h. Ses clés privées sont f et g (et donc aussi f_q^{-1}).

7.1.2. Brigitte fabrique sa signature.

Brigitte transforme le message $m \in L_m$ en un message e avec la clé publique NTRU d'Ali. Brigitte choisit un polynôme $w_2 \in \mathbb{L}_{w_2}$ et un autre $w_1 \in \mathbb{L}_w$ et calcule

$$w \equiv m + w_1 + pw_2 \pmod q.$$

Elle calcule ensuite

$$s \equiv f * w \pmod q.$$

Le message signé de Brigitte est donc (e, s).

7.1.3. Ali vérifie la signature.

En recevant le couple (e, s) de Brigitte, Ali calcule le message m à l'aide de ses clés privées relatives à NTRU. Il calcule ensuite

$$t \equiv h * s \pmod q.$$

Il vérifie ensuite si $s \neq 0$ et si les deux conditions suivantes sont satisfaites:

(i) $D_{\min} \leq \mathrm{Dev}(s, f_0 * m) \leq D_{\max}$,

(ii) $D_{\min} \leq \mathrm{Dev}(t, g_0 * m) \leq D_{\max}$.

Si toutes ces conditions sont satisfaites, alors Ali accepte la signature et le message.

7.2. Attaques contres la signature.

Stern [11] a montré récemment comment forger une signature s' a tout message m de telle sorte que le couple (m, s') convienne pour la signature NSS. Ceci a obligé les auteurs de NSS de changer de procédures et ont ainsi présenté une nouvelle méthode appelée NTRUSign [10].

8. Conclusion-Challenges.

La création de clés, le codage et le décodage avec le cryptosystème NTRU utilisent uniquement des produits, des sommes et des inverses de polynômes dans $\mathbb{Z}[X]/(X^N - 1)$ modulo des entiers petits p et q. Le plus coûteux est certainement le calcul de l'inverse d'un polynôme modulo p ou q. L'utilisation des transformées de Fourier rapides (FFT) permet ceci en un temps en $O(qN \log N)$.

Le système NTRU est très performant par rapport à ses concurents essentiellement pour deux raisons:

1. Les opérations nécessaires pour crypter et décrypter sont basées sur des opérations arithmétiques (addition) manipulant des nombres raisonnablement petits, contrairement aux autres systèmes de cryptographie qui sont basés pour la plupart sur l'exponentiation (RSA [8] et El Gamal [2]).

Le tableau suivant [6] donne la rapidité des temps d'execution des systèmes NTRU et RSA pour des niveaux de sécurité équivalents.

	Modérés NTRU 167 RSA 512	Hauts NTRU 263 RSA 1024	Très Hauts NTRU 503 RSA 2048
Clé	55.3	146	
Codage	2.4	3.9	4.7
Décodage	16.8	74.4	178.2

2. Les méthodes utilisées pour casser NTRU sont basées sur la réduction des réseaux par l'algorithme LLL. Mais cet algorithme ne fonctionne pas en parallèle sur des machines différentes. Ceci donne un avantage de NTRU par rapport à RSA et ECC, pour lesquels les attaques les peuvent être pratiquées sur des machines parallèles.

Les paramètres du système NTRU évoluent très rapidement selon les besoins de son utilisation. Par exemple, la valeur de p peut être considérée comme un polynôme.

La compagnie NTRU propose des challenges [5] pour décrypter des messages pour les trois niveaux de difficultés :

Sécurité	N	p	q	Somme
Haute	251	$1+X$	128	25.100$
Haute	347	$1+X$	128	34.700$
Très Haute	503	$1+X$	256	50.300$

References

1. D. Coppersmith, A. Shamir, *Lattice attacks on NTRU Proc. of Eurocypt '97,*, Springer-Verlag,, Lecture Notes in Computer Science **1233,** (197), 52–61.
2. Taher El Gamal, *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory, **IT–31(4)** (1985), 515–534.
3. J. Hoffstein, J. Pipher, J.H. Silverman, *NTRU: A new high speed public key cryptosystem*, in Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, Lecture Notes in Computer Sciences 1423 (J.P. Buhler, ed.), Springer-Verlag, Berlin, 1988,, 267–288.
4. A.K. Lenstra, H.W. Lenstra, L. Lovász, *Factoring polynomials with rational coefficients*, Math. Annalen **261** (1982), 515–534.
5. NTRU Challenges, `<http://www.ntru.com/cryptolab/challenges.htm>`.
6. NTRU Learning Center, `<http://www.ntru.com/>`.
7. NTRU Technical Center, `<http://www.ntru.com/cryptolab/>`.
8. R.L. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public key cryptosystems*, Communications of the ACM **21** (1978), 120–126.
9. C.P. Schnorr, M. Euchner, *Lattice basis reduction: improved practical algorithms and solving subset sum problems*, Mathematical Programing **66** (1994), 181-199.
10. J. Hoffstein, N. Howgrave-Graham, J.Pipher, J. Silverman, W. Whyte, *NTRUSign: Digital Signatures Using the NTRU Lattice*, preprint (2002), `<http://www.ntru.com/cryptolab/>`.
11. J. Stern, *Preliminary research annoucement : Cryptanalysis of the NTRU signature shceme (NSS)*, preprint (2002), `<http://www.di.ens.fr/ stern/nss.html>`.
12. J.H. Silverman, *A Meet-In-The-Middle Attack on an NTRU Private Key*, preprint (2002), `<http://www.ntru.com/cryptolab/tech_notes.htm>`.