LE CRYPTOSYSTEME NTRU
REALITES ET PERSPECTIVES

Abderrahmane NITAJ

Université de Caen
Département de Mathématiques
France

Oujda, 27 Avril 2007

CONTENU

CONTENU

les algorithmes NTRU

NTRUEncrypt

- Proposé en 1996.
- Le déchiffrement n'est pas garanti .

NSS

- Proposée en 2001.
- Cassée en 2001 (Gentry, Jonsonn, Stern, Szydlo).

NTRUSign

- Proposée en 2002.
- Partiellement cassée en 2006 (Nguyen, Regev).

les algorithmes NTRU

NTRUEncrypt

- Proposé en 1996.
- Le déchiffrement n'est pas garanti .

NSS

- Proposée en 2001.
- Cassée en 2001 (Gentry, Jonsonn, Stern, Szydlo).

NTRUSign

- Proposée en 2002.
- Partiellement cassée en 2006 (Nguyen, Regev).

les algorithmes NTRU

NTRUEncrypt

- Proposé en 1996.
- Le déchiffrement n'est pas garanti .

NSS

- Proposée en 2001.
- Cassée en 2001 (Gentry, Jonsonn, Stern, Szydlo).

NTRUSign

- Proposée en 2002.
- Partiellement cassée en 2006 (Nguyen, Regev).

Introduction

NTRU

- NTRU=Number Theorists aRe Us.
- NTRU=N-th degree TRUncated polynomial ring.
- Inventé et breveté en 1996 par J. Hoffstein, J. Pipher, J.H. Silverman.
- Investisseurs: Texas Instruments, Sonny Corpration, Macrovision,...
- Utilisation: Téléphone céllulaire, la radio identification (RFID).

Principe des couleurs

La coleur rouge

Toutes les notations rouges sont secrètes.

La couleur bleue ou noire

Toutes les notations bleues ou noires sont publiques.

Notations

Paramètres

- $N \in \mathbb{N}$, typiquement $N = 251, 347, 503$.
- $\mathcal{R} = \mathbb{Z}[X]/(X^N - 1)$, muni du produit de convolution :
  - $f(X) = f_0 + f_1 X + \cdots + f_{N-1} X^{N-1} \in \mathcal{R}$.
  - $f * g = h$ avec $h_k = \sum_{i+j \equiv k \pmod N} f_i g_j$.
- $p, q \in \mathbb{N}$ (ou $p \in \mathbb{Z}[X]$) pour fixer les anneaux $\mathbb{Z}/p\mathbb{Z}$ et $\mathbb{Z}/q\mathbb{Z}$ avec $p < q$ et $\gcd(p, q) = 1$.
- $\mathcal{R}_p = (\mathbb{Z}/p\mathbb{Z}[X])/(X^N - 1)$.
- $\mathcal{R}_q = (\mathbb{Z}/q\mathbb{Z}[X])/(X^N - 1)$.
- $\mathbb{L}_f$, $\mathbb{L}_g$, $\mathbb{L}_r$ sous-ensembles de $\mathcal{R}$. Leurs éléments possèdent respectivement d_f d_g, ou d_r coefficients égaux à 1 et le reste des coefficients est nul.
- $\mathbb{L}_m$ sous-ensemble de $\mathcal{R}$ à coefficients dans $\mathbb{Z}/p\mathbb{Z}$.

Paramètres

Sécurité	N	p	q	d_f	d_g	d_r
Moyenne	251	$3\ (2+X)$	128	72	71	72
Haute	347	$3\ (2+X)$	128	64	173	64
Très haute	503	$3\ (2+X)$	256	420	251	170

Principe d'utilisation

B veut envoyer un message m à A

Préparation de A

1. **A** choisit N, p, q.
2. **A** choisit $f \in \mathbb{L}_f$, et $g \in \mathbb{L}_g$.
3. **A** calcule $f_p \in \mathcal{R}_p$ tel que $f_p * f = 1$.
4. **A** calcule $f_q \in \mathcal{R}_q$ tel que $f_q * f = 1$.
5. **A** calcule $h \in \mathcal{R}_q$ avec $h = f_q * g$.
6. **A** publie sa clé publique (h, q, p, N).
7. La clé secrète de **A** est f (et f_p).

Principe d'utilisation

B veut envoyer un message m à A

Préparation de A: $\longrightarrow h$

Chiffrement par B

1. **B** transforme son message en un polynôme $m \in \mathcal{R}_p$.
2. **B** choisit un polynôme aléatoire $r \in \mathbb{L}_r$.
3. **B** calcule $e = p * r * h + m \pmod q$ et envoi e à **A**.

Principe d'utilisation

B veut envoyer un message m à A

Préparation de A: $\longrightarrow h$

Chiffrement par B: $\longrightarrow e$

Déchiffrement par A

1. **A** calcule $a = e * f \pmod{q}$.
2. **A** transforme les coefficients de a dans un intervalle $[A, A + q - 1]$, (A est un paramètre "pour le bon fonctionnement" de NTRU).
3. **A** calcule $f_p * a = m \pmod{p}$.

Principe d'utilisation

B a envoyé un message m à A

Preuves

1. $f_q * f = 1 \pmod q$, $\quad f_p * f = 1 \pmod p$,
2. $h = f_q * g$, $\quad e = p * r * h + m$.

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

3. Avec le "bon paramètre A",
   $p * r * g + m * f \in \mathbb{Z}[X]/(X^N - 1)$.
4. $f_p * a = f_p * p * r * g + f_p * m * f \pmod p = m \pmod p$

Equivalence des versions

RSA	NTRU
$n = 1024$	$N = 251$
$n = 2048$	$N = 347$
$n = 4096$	$N = 503$

Avantages et inconvénients

Avantages

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(N^2)$ contre $\mathcal{O}(n^3)$ pour RSA.
- Le chiffrement et le déchiffrement sont plus rapides que RSA : de 30 à 300 fois plus rapide.

Inconvénients

- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$ plus grande que celle de RSA (n).
- Le déchiffrement n'est pas conforme une fois tous les 2^{25} avec $N = 251$.

Avantages et inconvénients

Avantages

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(N^2)$ contre $\mathcal{O}(n^3)$ pour RSA.
- Le chiffrement et le déchiffrement sont plus rapides que RSA : de 30 à 300 fois plus rapide.

Inconvénients

- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$ plus grande que celle de RSA (n).
- Le déchiffrement n'est pas conforme une fois tous les 2^{25} avec $N = 251$.

Signatures

Signatures NTRU

1. NTRU Signature Scheme (NSS):
   - Proposée en 2000 .
   - Cassée en 2001 .
2. Revised NTRU Signature Scheme (R-NSS):
   - Proposée en 2001.
   - Cassée en 2002.
3. NTRUSign
   - Proposée en 2002 .
   - Cassée (Version non perturbée) en 2006.
4. Conclusion: Méfiance de la communauté scientifique.

La sécurité de NTRU

La factorisation des polynômes modulaires

- Rappel: la clé publique est $h = f_q * g \pmod{q}$.
- Hypothèse: Etant donné un polynôme $h \in \mathbb{Z}_q[X]/(X^N - 1)$, il est difficile de déterminer deux "petits" polynômes f et g tels que $f * h = g \pmod{q}$.

Le problème SVP: the Shortest Vector Problem

Soit $\mathcal{L}$ un réseau. Déterminer le plus petit vecteur non nul de $\mathcal{L}$.

Le problème CVP: the Closest Vector Problem

Soit $\mathcal{L}$ un réseau et $v_0 \notin \mathcal{L}^*$. Déterminer un vecteur $v \in \mathcal{L}$ proche de v_0.

La sécurité de NTRU

La factorisation des polynômes modulaires

- Rappel: la clé publique est $h = f_q * g \pmod{q}$.
- Hypothèse: Etant donné un polynôme $h \in \mathbb{Z}_q[X]/(X^N - 1)$, il est difficile de déterminer deux "petits" polynômes f et g tels que $f * h = g \pmod{q}$.

Le problème SVP: the Shortest Vector Problem

Soit $\mathcal{L}$ un réseau. Déterminer le plus petit vecteur non nul de $\mathcal{L}$.

Le problème CVP: the Closest Vector Problem

Soit $\mathcal{L}$ un réseau et $v_0 \notin \mathcal{L}^*$. Déterminer un vecteur $v \in \mathcal{L}$ proche de v_0.

La sécurité de NTRU

La factorisation des polynômes modulaires

- Rappel: la clé publique est $h = f_q * g \pmod{q}$.
- Hypothèse: Etant donné un polynôme $h \in \mathbb{Z}_q[X]/(X^N - 1)$, il est difficile de déterminer deux "petits" polynômes f et g tels que $f * h = g \pmod{q}$.

Le problème SVP: the Shortest Vector Problem

Soit $\mathcal{L}$ un réseau. Déterminer le plus petit vecteur non nul de $\mathcal{L}$.

Le problème CVP: the Closest Vector Problem

Soit $\mathcal{L}$ un réseau et $v_0 \notin \mathcal{L}^*$. Déterminer un vecteur $v \in \mathcal{L}$ proche de v_0.

CONTENU

Les réseaux

Le produit scalaire

1. $u = (u_1, u_2, \cdots, u_n) \in \mathbb{R}^n$, $v = (v_1, v_2, \cdots, v_n) \in \mathbb{R}^n$.
2. $(u, v) = \sum_{i=1}^{n} u_i v_i$.

La norme euclidienne

1. $u = (u_1, u_2, \cdots, u_n) \in \mathbb{R}^n$.
2. $||u|| = \sqrt{(u, u)} = \sqrt{\sum_{i=1}^{n} u_i^2}$.

Les réseaux

Définition

1. Soient $b_1, b_2, \cdots, b_n \in \mathbb{R}^n$, n vecteurs linéairement indépendants.
2. Soit $\mathcal{B} = \{b_1, b_2, \cdots b_n\}$.
3. Le réseau engendré par $\mathcal{B}$ est

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} \lambda_i b_i, \qquad \lambda_i \in \mathbb{Z} \right\}.$$

Caractéristiques

1. $\dim(\mathcal{L}) = n$.
2. $\det(\mathcal{L}) = |\det(\mathcal{B})| = \text{volume}\left\{\sum_{i=1}^{n} \alpha_i v_i, \quad 0 \leq \alpha_i < 1\right\}$

Vecteurs courts

Le problème SVP: the Shortest Vector Problem

1. Soit $\mathcal{L}$ un réseau. Déterminer un court vecteur non nul $v \in \mathcal{L} : ||v|| = \lambda_1(\mathcal{L}) = \inf\{||u||, \quad u \in \mathcal{L}^*\}$.
2. NP-dur, Ajtai, 1996 (pour des réductions polynômiales probabilistes).
3. Peut être résolu en pratique pour les dimensions $n \leq 30$.
4. A la base de certains cryptosystèmes :
   - Ajtai-Dwork (1996), cassé par Ngyuen-Stern (1996) pour les petits paramètres et non pratique pour les grands paramètres.
   - Cryptosystèmes "Sac à dos" (dès 1978), cassés totalement (terminus en 1997).
   - NTRU

Vecteurs courts

Le problème CVP: the Closest Vector Problem

1. Soit $\mathcal{L}$ un réseau et $v_0 \notin \mathcal{L}^*$. Déterminer un vecteur $v \in \mathcal{L}$ proche de v_0 ($||v - v_0|| \leq \lambda_1(\mathcal{L})$)
2. NP-dur, van Emde Boas, 1981.
3. Peut être résolu en pratique pour les dimensions $n \leq 30$.
4. A la base de certains cryptosystèmes :
   - Goldreich-Goldwasser-Halevi (GGH) (1996), partiellement cassé par Nguyen (1999), puis encore plus par Nguyen et Regev (2006).

Solution théorique

Le théorème de Minkowski

1. *Soit $\mathcal{L}$ un réseau de dimension n. Il existe un vecteur non nul $v \in \mathcal{L}$ tel que*

$$||v|| \leq \sqrt{n}\, det(\mathcal{L})^{1/n}.$$

2. Transformé en algorithme par Ajtai Kumar et Sivakumar en 2001, mais non pratique car le temps et l'espace sont en $2^{\mathcal{O}(n)}$.

L'algorithme LLL

1. LLL=Lenstra-Lenstra- Lovasz, 1982.
2. Nombreuses améliorations.

Théorème

- *Soit $\mathcal{L}$ un réseau. L'algorithme LLL produit une base $(b_1, b_2, \cdots, b_n)$ avec les propriétés*
  1. $||b_1|| \leq 2^{\frac{n-1}{4}} \det(\mathcal{L})^{1/m}$.
  2. Relation avec SVP: $||b_1|| \leq \left(\frac{4}{3}\right)^{\frac{n}{2}} \lambda_1(\mathcal{L})$.
  3. $\det(\mathcal{L}) \leq \prod_{i=1}^{n} ||b_i| \leq \left(\frac{4}{3}\right)^{\frac{n(n-1)}{4}} \det(\mathcal{L})$.
- Complexité : $\mathcal{O}\left(n^6 (\log B)^3\right)$, avec $B = \max_i(||b_i^{\text{init}}||)$.
- Pratique pour les dimensions $n \leq 1000$.

Applications de l'algorithme LLL

Cryptogaphie

1. Résolution du problème du sac à dos.
2. Attaque de RSA.
3. Attaque des signatures NTRU.

Autres applications

- Factorisation des polynômes.
- Factorisation des entiers (LLL intervient dans NFS).
- Equations diophantiennes.
- Courbes elliptiques.
- Théorie algébrique des nombres

L'algorithme BKZ

1. BKZ=Block Korkine- Zolatarev,
2. Schnor-Euchner, 1994
3. Plus adapté pour résoudre le problème SVP.
4. Sa complexité n'est pas prouvé polnômiale.

CONTENU

Attaques exhaustives

Attaque sur la clé publique

1. Principe: $h = f_q * g \pmod{q}$, donc $f * h = g \pmod{q}$.
2. Méthode: Tester tous les $f \in \mathbb{L}_f$ jusqu'à ce que $f * h \pmod{q}$ ait des petits coefficients.

Deuxième attaque sur la clé publique

1. Principe: $h = f_q * g \pmod{q}$, donc $g * h = f \pmod{q}$.
2. Méthode: Tester tous les $g \in \mathbb{L}_g$ jusqu'à ce que $g * h \pmod{q}$ ait des petits coefficients.

Attaques exhaustives

Attaque sur la clé publique

1. Principe: $h = f_q * g \pmod{q}$, donc $f * h = g \pmod{q}$.
2. Méthode: Tester tous les $f \in \mathbb{L}_f$ jusqu'à ce que $f * h \pmod{q}$ ait des petits coefficients.

Deuxième attaque sur la clé publique

1. Principe: $h = f_q * g \pmod{q}$, donc $g * h = f \pmod{q}$.
2. Méthode: Tester tous les $g \in \mathbb{L}_g$ jusqu'à ce que $g * h \pmod{q}$ ait des petits coefficients.

Attaques exhaustives

Attaque sur le message

1. Principe: $e = m + pr * h \pmod q$, donc $e - pr * h = m \pmod q$.
2. Méthode: Tester tous les $r \in \mathbb{L}_m$ jusqu'à ce que $e - pr * h \pmod q$ ait des petits coefficients.

Attaque par le milieu

1. Principe: $h = f_q * g \pmod q$, donc $f * h = g \pmod q$.
2. On pose $f = f_1 + f_2$.
3. Méthode: Tester tous les $(f_1, f_2) \in \mathbb{L}_f^2$ jusqu'à ce que $f_1 * h \pmod q$ et $f_1 * h \pmod q$ ait des coefficients presque identiques.

Attaques exhaustives

Attaque sur le message

1. Principe: $e = m + pr * h \pmod q$, donc $e - pr * h = m \pmod q$.
2. Méthode: Tester tous les $r \in \mathbb{L}_m$ jusqu'à ce que $e - pr * h \pmod q$ ait des petits coefficients.

Attaque par le milieu

1. Principe: $h = f_q * g \pmod q$, donc $f * h = g \pmod q$.
2. On pose $f = f_1 + f_2$.
3. Méthode: Tester tous les $(f_1, f_2) \in \mathbb{L}_f^2$ jusqu'à ce que $f_1 * h \pmod q$ et $f_1 * h \pmod q$ ait des coefficients presque identiques.

Attaques exhaustives

Attaque sur des transmissions multiples

Si on envoie k fois le même message m avec la même clé h.

1. Principe: Pour $1 \leq i \leq k$, on a $e_i = m + pr_i * h \pmod{q}$, donc $e_i - e_1 = p(r_i - r_1) * h$.
2. Méthode: Pour $2 \leq i \leq k$, on calcule $r_i - r_1 = (e_i - e_1) * p^{-1} * h^{-1} \pmod{q}$.
3. Ceci donne des des informations sur certains coefficients de r_1.
4. On fait une recherche exhaustive sur les coefficients manquants de r_1.
5. On calcule pour chaque r_1 la valeur du message $m = e_1 - pr_1 * h$.

Les attaques

Coppersmith-Shamir, 1998

- Principe: Puisque $h = f_q * g \pmod q$, alors $f * h = g \pmod q$ avec f et g "petits".
- Réseau: $\mathcal{L}_{CS} = \{(a, b) \in \mathbb{Z}^{2N}, \quad a * h = b \pmod q\}$.
- La matrice correspondante est de la forme:

$$\left[\begin{array}{cccc|c} & & I_N & & 0_N \\ \hline h_0 & h_{N-1} & \cdots & h_1 & \\ h_1 & h_0 & \cdots & h_2 & \\ \vdots & \vdots & \ddots & \vdots & qI_N \\ h_{N-1} & h_{N-2} & \cdots & h_0 & \end{array}\right]$$

- Réduire $\mathcal{L}_{CS}$ et déterminer (f, g) parmi les petites solutions (a, b).

Les attaques

May, 2000

C'est une variante de l'attaque de Coppersmith-Shamir

Gentry, Jonsonn, Stern, Szydlo, 2001

- Attaque contre la signature NSS.
- Défaut: Chaque signature sur un message donne des renseignements sur la clé privée.
- Contrefaçon: Toute personne peut signer n'importe quel message sans connaître la clé privée.
- Méthode basée sur la réduction de réseaux de dimension $2N$ (LLL et BKZ).

Les attaques

May, 2000

C'est une variante de l'attaque de Coppersmith-Shamir

Gentry, Jonsonn, Stern, Szydlo, 2001

- Attaque contre la signature NSS.
- Défaut: Chaque signature sur un message donne des renseignements sur la clé privée.
- Contrefaçon: Toute personne peut signer n'importe quel message sans connaître la clé privée.
- Méthode basée sur la réduction de réseaux de dimension $2N$ (LLL et BKZ).

Les attaques

Gentry-Szydlo 2001

- Attaque contre la version Revisée de NSS
- Méthode : Réduction de réseaux sous-jacents à R-NSS et résolution de SVP en temps polynômial.

Les attaques

Nguyen-Regev, 2006

- Attaque contre la signature NTRUSign avec $N = 251$.
- Défaut: Chaque signature sur un message donne des renseignements sur la clé privée.
- Problème sous-jacent: Connaissant un grand nombre de points d'un parallélépipède inconnu, comment déterminer ce parallélépipède.
- Méthode basée sur la résolution de problèmes d'optimisation.
- Avec 90000 (amélioré à 400) signatures délivrées avec la même clé privée, on peut retrouver cette clé.

CONTENU

Le présent et le futur

Réalités

- NTRU est encore au stade expérimental.
- Nombreuse versions.
- Faiblesse des signatures.
- Déchiffrement non toujours conforme.

Perspectives

- Chiffrement et déchiffrement rapides.
- Une alternative à RSA?
- Applications dans les puces radio RFID (Radio Frequency Identification).

Le présent et le futur

Réalités

- NTRU est encore au stade expérimental.
- Nombreuse versions.
- Faiblesse des signatures.
- Déchiffrement non toujours conforme.

Perspectives

- Chiffrement et déchiffrement rapides.
- Une alternative à RSA?
- Applications dans les puces radio RFID (Radio Frequency Identification).