RSA Versus NTRU

Abderrahmane NITAJ

Laboratoire de Mathématiques Nicolas Oresme
Université de Caen, France

Casablanca, 28 février 2011

الدار البيضاء

CONTENU

CONTENU

Utlisation

RSA

- Inventé en 1977 par Rivest+Shamir+Adleman.
- Commercialisé par EMC.
- Serveurs Web, Internet (SSL/TLS), PGP, Cartes de crédit, Paiement électronique, Téléphones portables, Microsoft, Apple Computer, Cisco Systems, Intel, Nokia, Sonny Ericson .

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.

Définitions

le module RSA

- p et q sont deux nombres premiers secrets de même taille.
- $N = pq$ est le module RSA.
  1. Moyenne sécurité : N de 1024 bits ($\approx$ 309 chiffres)
  2. haute sécurité : N de 2048 bits ($\approx$ 617 chiffres)

L'indicateur d'Euler

$\phi(N) = (p-1)(q-1)$.

les clés

- $e \in \mathbb{N}, 1 \leq e \leq \phi(N)$, est la clé publique.
- $d \in \mathbb{N}, 1 \leq d \leq \phi(N), \quad ed \equiv 1 \pmod{\phi(N)}$ est la clé secrète.

Définitions

le module RSA

- p et q sont deux nombres premiers secrets de même taille.
- $N = pq$ est le module RSA.
  1. Moyenne sécurité : N de 1024 bits ($\approx$ 309 chiffres)
  2. haute sécurité : N de 2048 bits ($\approx$ 617 chiffres)

L'indicateur d'Euler

$\phi(N) = (p-1)(q-1)$.

les clés

- $e \in \mathbb{N}, 1 \leq e \leq \phi(N)$, est la clé publique.
- $d \in \mathbb{N}, 1 \leq d \leq \phi(N), \quad ed \equiv 1 \pmod{\phi(N)}$ est la clé secrète.

Définitions

le module RSA

- p et q sont deux nombres premiers secrets de même taille.
- $N = pq$ est le module RSA.
  1. Moyenne sécurité : N de 1024 bits ($\approx$ 309 chiffres)
  2. haute sécurité : N de 2048 bits ($\approx$ 617 chiffres)

L'indicateur d'Euler

$\phi(N) = (p-1)(q-1)$.

les clés

- $e \in \mathbb{N}$, $1 \leq e \leq \phi(N)$, est la clé publique.
- $d \in \mathbb{N}$, $1 \leq d \leq \phi(N)$, $\quad ed \equiv 1 \pmod{\phi(N)}$ est la clé secrète.

Principe d'utilisation

B veut envoyer un message à A

1. **A** choisit deux nombres premiers p et q de même taille.
2. **A** calcule $N = pq$ et $\phi(N) = (p-1)(q-1)$.
3. **A** choisit $e \in \mathbb{N}$, $1 \leq e \leq \phi(N)$.
4. **A** calcule $d \in \mathbb{N}$, $1 \leq d \leq \phi(N)$ et $ed \equiv 1 \pmod{\phi(N)}$.
5. **A** publie N, e et garde p, q, d secrets.
6. **B** transforme son message en entier m avec $1 < m < N$.
7. **B** calcule $c \equiv m^e \pmod{N}$ et envoi c à **A**.
8. **A** calcule $c^d \equiv m \pmod{N}$ et retrouve le message m de **B**.

Preuve (basée sur le théorème d'Euler)

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \times m^{k\phi(N)} \equiv m \pmod{N}.$$

Principe d'utilisation

B veut envoyer un message à A

1. **A** choisit deux nombres premiers p et q de même taille.
2. **A** calcule $N = pq$ et $\phi(N) = (p-1)(q-1)$.
3. **A** choisit $e \in \mathbb{N}$, $1 \leq e \leq \phi(N)$.
4. **A** calcule $d \in \mathbb{N}$, $1 \leq d \leq \phi(N)$ et $ed \equiv 1 \pmod{\phi(N)}$.
5. **A** publie N, e et garde p, q, d secrets.
6. **B** transforme son message en entier m avec $1 < m < N$.
7. **B** calcule $c \equiv m^e \pmod{N}$ et envoi c à **A**.
8. **A** calcule $c^d \equiv m \pmod{N}$ et retrouve le message m de **B**.

Preuve (basée sur le théorème d'Euler)

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \times m^{k\phi(N)} \equiv m \pmod{N}.$$

Principe d'utilisation

B veut envoyer un message à A

1. **A** choisit deux nombres premiers p et q de même taille.
2. **A** calcule $N = pq$ et $\phi(N) = (p-1)(q-1)$.
3. **A** choisit $e \in \mathbb{N}$, $1 \leq e \leq \phi(N)$.
4. **A** calcule $d \in \mathbb{N}$, $1 \leq d \leq \phi(N)$ et $ed \equiv 1 \pmod{\phi(N)}$.
5. **A** publie N, e et garde p, q, d secrets.
6. **B** transforme son message en entier m avec $1 < m < N$.
7. **B** calcule $c \equiv m^e \pmod{N}$ et envoi c à **A**.
8. **A** calcule $c^d \equiv m \pmod{N}$ et retrouve le message m de **B**.

Preuve (basée sur le théorème d'Euler)

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \times m^{k\phi(N)} \equiv m \pmod{N}.$$

Principe d'utilisation

B veut envoyer un message à A

1. **A** choisit deux nombres premiers p et q de même taille.
2. **A** calcule $N = pq$ et $\phi(N) = (p-1)(q-1)$.
3. **A** choisit $e \in \mathbb{N}$, $1 \leq e \leq \phi(N)$.
4. **A** calcule $d \in \mathbb{N}$, $1 \leq d \leq \phi(N)$ et $ed \equiv 1 \pmod{\phi(N)}$.
5. **A** publie N, e et garde p, q, d secrets.
6. **B** transforme son message en entier m avec $1 < m < N$.
7. **B** calcule $c \equiv m^e \pmod{N}$ et envoi c à **A**.
8. **A** calcule $c^d \equiv m \pmod{N}$ et retrouve le message m de **B**.

Preuve (basée sur le théorème d'Euler)

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \times m^{k\phi(N)} \equiv m \pmod{N}.$$

Attaques sur RSA

1. Attaques élémentaires.
2. Factorisation du module $N = pq$.
3. Canaux cachés (Side Channels).
4. Attaques sur les clés.
5. Ordinateur quantique en 2???.

Attaques sur RSA

1. Attaques élémentaires.
2. Factorisation du module $N = pq$.
3. Canaux cachés (Side Channels).
4. Attaques sur les clés.
5. Ordinateur quantique en 2???.

Attaques sur RSA

1. Attaques élémentaires.
2. Factorisation du module $N = pq$.
3. Canaux cachés (Side Channels).
4. Attaques sur les clés.
5. Ordinateur quantique en 2???.

Attaques sur RSA

1. Attaques élémentaires.
2. Factorisation du module $N = pq$.
3. Canaux cachés (Side Channels).
4. Attaques sur les clés.
5. Ordinateur quantique en 2???.

Attaques sur RSA

1. Attaques élémentaires.
2. Factorisation du module $N = pq$.
3. Canaux cachés (Side Channels).
4. Attaques sur les clés.
5. Ordinateur quantique en 2???.

Attaques élémentaires

1. Module commun et plusieurs clés.
2. Plus de e modules pour la même clé ($e = 3$) et le même message.
3. Attaque cyclique (cycling attack).
4. Attaque de Fermat pour factoriser $N = pq$.

Contremesures

- Utiliser un seul module et une seule clé.
- Ne jamais envoyer le même message plus d'une fois avec le même module.
- Prendre p et q ayant une grande difference $|p - q| > N/2^{100}$.

Attaques élémentaires

1. Module commun et plusieurs clés.
2. Plus de e modules pour la même clé ($e = 3$) et le même message.
3. Attaque cyclique (cycling attack).
4. Attaque de Fermat pour factoriser $N = pq$.

Contremesures

- Utiliser un seul module et une seule clé.
- Ne jamais envoyer le même message plus d'une fois avec le même module.
- Prendre p et q ayant une grande difference $|p - q| > N/2^{100}$.

Factorisation du module $N = pq$

Algorithme de factorisation ECM

- ECM = Elliptic Curve Method.
- H.W. Lenstra, Jr. 1987.
- Complexité sous-exponentielle: $c > 0$,

$$\mathcal{O}\left(e^{c\sqrt{\log(p)\log\log p}}\right).$$

Algorithme de factorisation GNFS

- GNFS = General Number Field Sieve.
- J.M. Pollard, 1988.
- Complexité sous-exponentielle: $c < 2$,

$$\mathcal{O}\left(e^{(c+o(1))(\log(N))^{1/3}(\log\log(N))^{2/3}}\right).$$

Factorisation du module $N = pq$

Algorithme de factorisation ECM

- ECM = Elliptic Curve Method.
- H.W. Lenstra, Jr. 1987.
- Complexité sous-exponentielle: $c > 0$,

$$\mathcal{O}\left(e^{c\sqrt{\log(p)\log\log p}}\right).$$

Algorithme de factorisation GNFS

- GNFS = General Number Field Sieve.
- J.M. Pollard, 1988.
- Complexité sous-exponentielle: $c < 2$,

$$\mathcal{O}\left(e^{(c+o(1))(\log(N))^{1/3}(\log\log(N))^{2/3}}\right).$$

Factorisation du module $N = pq$

Factorisation du module par GNFS

1. RSA-576 (173 chiffres), 2003.
2. RSA-640 (192 chiffres), 2005.
3. RSA-768 (231 chiffres), 2010.

Les modules actuels de RSA

- RSA-1024 : module RSA de 1024 bits (308 chiffres).
- RSA-2048 : module RSA de 2048 bits (617 chiffres).
- Prendre p et q de tailles voisines.

Factorisation du module $N = pq$

Factorisation du module par GNFS

1. RSA-576 (173 chiffres), 2003.
2. RSA-640 (192 chiffres), 2005.
3. RSA-768 (231 chiffres), 2010.

Les modules actuels de RSA

- RSA-1024 : module RSA de 1024 bits (308 chiffres).
- RSA-2048 : module RSA de 2048 bits (617 chiffres).
- Prendre p et q de tailles voisines.

Canaux cachés (Side Channels)

Plusieurs sortes

1. Attaques temporelles (timing attacks), Kocher, 1995.
2. Analyse de la consommation (Power Analysis, Boneh et al., 1997).
3. Attaques par injection de fautes (Fault Analysis).
4. Attaques par champ électromagnetique (Elecrtomagnetic Analysis).

Canaux cachés (Side Channels)

Problème

Pour calculer $c \equiv m^d \pmod N$, on écrit $d = \overline{1 d_{k-2} \cdots d_0}$ en base 2 et on effectue l'algorithme

1. $c = m$.
2. Pour $i = k-2$ à 0 faire $c = c^2 \pmod N$.
3. Si $d_i = 1$ alors $c = m \cdot c \pmod N$.

Contre mesures

On peut utliser l'algorithme

1. $a_0 = m$.
2. Pour $i = k-2$ à 0 faire $a_0 = a_0^2 \pmod N$, $a_1 = m \cdot a_0 \pmod N$, $a_0 = a_{d_i}$.
3. Renvoyer a_0.

Canaux cachés (Side Channels)

Problème

Pour calculer $c \equiv m^d \pmod N$, on écrit $d = \overline{1 d_{k-2} \cdots d_0}$ en base 2 et on effectue l'algorithme

1. $c = m$.
2. Pour $i = k-2$ à 0 faire $c = c^2 \pmod N$.
3. Si $d_i = 1$ alors $c = m \cdot c \pmod N$.

Contre mesures

On peut utliser l'algorithme

1. $a_0 = m$.
2. Pour $i = k-2$ à 0 faire $a_0 = a_0^2 \pmod N$, $a_1 = m \cdot a_0 \pmod N$, $a_0 = a_{d_i}$.
3. Renvoyer a_0.

Attaques sur les clés

Equations vérifiées par les clés

1. $ed - k\phi(N) = 1$: Wiener, 1990, si $d < N^{1/4}$, on peut factoriser $N = pq$.
2. $ed - k\phi(N) = 1$: Boneh-Durfee, 1999, si $d < N^{0.292}$, on peut factoriser $N = pq$.
3. $ex - k\phi(N) = y$: Blömer-May, 2004, si x, $|y|$ et k sont assez petits.
4. Attaques avec des parties connues de la clé (Partial Key Exposure Attacks).

Contre mesures

- Prendre une clé privée aléatoire avec $d > \sqrt{N}$.

Attaques sur les clés

Equations vérifiées par les clés

1. $ed - k\phi(N) = 1$: Wiener, 1990, si $d < N^{1/4}$, on peut factoriser $N = pq$.
2. $ed - k\phi(N) = 1$: Boneh-Durfee, 1999, si $d < N^{0.292}$, on peut factoriser $N = pq$.
3. $ex - k\phi(N) = y$: Blömer-May, 2004, si x, $|y|$ et k sont assez petits.
4. Attaques avec des parties connues de la clé (Partial Key Exposure Attacks).

Contre mesures

- Prendre une clé privée aléatoire avec $d > \sqrt{N}$.

Ordinateur quantique en 2...?

Généralités

1. Deux états de base: $|0>$ et $|1>$.
2. Un qubit = $\alpha|0\rangle + \beta|1\rangle$, avec $\alpha, \beta \in \mathbb{C}$ et $|\alpha|^2 + |\beta|^2 = 1$.
3. La puissance est exponentielle en le nombre de qubits.
4. Le premier ordinateur quantique: 1996, IBM, avec 2 qubits.
5. 2007, D-Wave Systems, avec 16 qubits.
6. Shor, 1994: la factorisation et le logarithme discret en temps polynômial $\mathcal{O}\left(\log(N)^3\right)$.

Contre mesures

- Ne plus utiliser RSA tel qu'il est défini actuellement.

Ordinateur quantique en 2...?

Généralités

1. Deux états de base: $|0>$ et $|1>$.
2. Un qubit = $\alpha|0\rangle + \beta|1\rangle$, avec $\alpha, \beta \in \mathbb{C}$ et $|\alpha|^2 + |\beta|^2 = 1$.
3. La puissance est exponentielle en le nombre de qubits.
4. Le premier ordinateur quantique: 1996, IBM, avec 2 qubits.
5. 2007, D-Wave Systems, avec 16 qubits.
6. Shor, 1994: la factorisation et le logarithme discret en temps polynômial $\mathcal{O}\left(\log(N)^3\right)$.

Contre mesures

- Ne plus utiliser RSA tel qu'il est défini actuellement.

CONTENU

Les protocoles NTRU

NTRUEncrypt

- Proposé en 1996, puis en 2001, puis en 2005.
- Commercialisé par Security Innovation.
- Principal inconvénient jusqu'à 2005 : le déchiffrement n'est pas correct tout le temps.

NSS (Signature)

- Proposée en 2000.
- Cassée en 2001 (Gentry, Jonsson, Stern, Szydlo), $\Longrightarrow$ abandonnée.

NTRUSign (Signature)

- Proposée en 2001.
- Partiellement cassée en 2006 (Nguyen, Regev).

Les protocoles NTRU

NTRUEncrypt

- Proposé en 1996, puis en 2001, puis en 2005.
- Commercialisé par Security Innovation.
- Principal inconvénient jusqu'à 2005 : le déchiffrement n'est pas correct tout le temps.

NSS (Signature)

- Proposée en 2000.
- Cassée en 2001 (Gentry, Jonsson, Stern, Szydlo), $\Longrightarrow$ abandonnée.

NTRUSign (Signature)

- Proposée en 2001.
- Partiellement cassée en 2006 (Nguyen, Regev).

Les protocoles NTRU

NTRUEncrypt

- Proposé en 1996, puis en 2001, puis en 2005.
- Commercialisé par Security Innovation.
- Principal inconvénient jusqu'à 2005 : le déchiffrement n'est pas correct tout le temps.

NSS (Signature)

- Proposée en 2000.
- Cassée en 2001 (Gentry, Jonsson, Stern, Szydlo), $\Longrightarrow$ abandonnée.

NTRUSign (Signature)

- Proposée en 2001.
- Partiellement cassée en 2006 (Nguyen, Regev).

Introduction

NTRU

- NTRU=N-th degree TRUncated polynomial ring.
- Inventé en 1996 par J. Hoffstein, J. Pipher, J.H. Silverman.
- Breveté en 2000 par NTRU Cryptosystems, Inc.
- Partenariat: Microsoft, Intel, NXP, Texas Instruments, Authentec, Smart Card Alliance,...
- Utilisation: Cartes à puces, Téléphonie céllulaire, la radio identification (RFID).

Anneau $\mathcal{R} = \mathbb{Z}[X]/(X^N - 1)$

Eléments de $\mathcal{R}$

$f = (f_0, f_1, \cdots, f_{N-1}) = \sum_{i=0}^{N-1} f_i X^i,$
$g = (g_0, g_1, \cdots, g_{N-1}) = \sum_{i=0}^{N-1} g_i X^i,$

Somme

$f + g = (f_0 + g_0, f_1 + g_1, \cdots, f_{N-1} + g_{N-1}).$

Produit de convolution

$f * g = h = (h_0, h_1, \cdots, h_{N-1})$ avec

$$h_k = \sum_{i+j \equiv k \pmod N} f_i g_j$$

Anneau $\mathcal{R} = \mathbb{Z}[X]/(X^N - 1)$

Eléments de $\mathcal{R}$

$f = (f_0, f_1, \cdots, f_{N-1}) = \sum_{i=0}^{N-1} f_i X^i,$
$g = (g_0, g_1, \cdots, g_{N-1}) = \sum_{i=0}^{N-1} g_i X^i,$

Somme

$f + g = (f_0 + g_0, f_1 + g_1, \cdots, f_{N-1} + g_{N-1}).$

Produit de convolution

$f * g = h = (h_0, h_1, \cdots, h_{N-1})$ avec

$$h_k = \sum_{i+j \equiv k \pmod N} f_i g_j$$

Anneau $\mathcal{R} = \mathbb{Z}[X]/(X^N - 1)$

Eléments de $\mathcal{R}$

$f = (f_0, f_1, \cdots, f_{N-1}) = \sum_{i=0}^{N-1} f_i X^i,$
$g = (g_0, g_1, \cdots, g_{N-1}) = \sum_{i=0}^{N-1} g_i X^i,$

Somme

$f + g = (f_0 + g_0, f_1 + g_1, \cdots, f_{N-1} + g_{N-1}).$

Produit de convolution

$f * g = h = (h_0, h_1, \cdots, h_{N-1})$ avec

$$h_k = \sum_{i+j \equiv k \pmod N} f_i g_j$$

Anneau $\mathcal{R} = \mathbb{Z}[X]/(X^N - 1)$

Table du produit

$f = (f_0, f_1, \cdots, f_{N-1}), \quad g = (g_0, g_1, \cdots, g_{N-1}).$

$$h = f * g$$

	1	X	X^2	$\cdots$	X^{N-1}
	f_0g_0	f_0g_1	f_0g_2	$\cdots$	f_0g_{N-1}
$+$	f_1g_{N-1}	f_1g_0	f_1g_1	$\cdots$	f_1g_{N-2}
$+$	f_2g_{N-2}	f_2g_{N-1}	f_2g_0	$\cdots$	f_2g_{N-3}
$+$	f_3g_{N-3}	f_3g_{N-2}	f_3g_{N-1}	$\cdots$	f_3g_{N-4}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$+$	$f_{N-1}g_1$	$f_{N-1}g_2$	$f_{N-1}g_3$	$\cdots$	$f_{N-1}g_0$
$h =$	h_0	h_1	h_2	$\cdots$	h_{N-1}

Anneau $\mathcal{R} = \mathbb{Z}[X]/(X^N - 1)$

Table du produit

$f = (f_0, f_1, \cdots, f_{N-1}), \quad g = (g_0, g_1, \cdots, g_{N-1}).$

$$h = f * g$$

	1	X	X^2	$\cdots$	X^{N-1}
	f_0g_0	f_0g_1	f_0g_2	$\cdots$	f_0g_{N-1}
$+$	f_1g_{N-1}	f_1g_0	f_1g_1	$\cdots$	f_1g_{N-2}
$+$	f_2g_{N-2}	f_2g_{N-1}	f_2g_0	$\cdots$	f_2g_{N-3}
$+$	f_3g_{N-3}	f_3g_{N-2}	f_3g_{N-1}	$\cdots$	f_3g_{N-4}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$+$	$f_{N-1}g_1$	$f_{N-1}g_2$	$f_{N-1}g_3$	$\cdots$	$f_{N-1}g_0$
$h =$	h_0	h_1	h_2	$\cdots$	h_{N-1}

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 $h_k = \sum_{i+j \equiv k \pmod N} f_i g_j$.
- $p, q \in \mathbb{N}$ avec $p < q$ et $\gcd(p, q) = 1$.
- $\mathcal{R}_p = (\mathbb{Z}/p\mathbb{Z}[X])/(X^N - 1)$.
- $\mathcal{R}_q = (\mathbb{Z}/q\mathbb{Z}[X])/(X^N - 1)$.
- d_F, d_g, d_r des entiers fixés et

$$\mathcal{B}(d) = \left\{ f(X) \in \mathcal{R} \,\middle|\, f(X) = \sum_{i=1}^{d} X^{n_i} \right\}.$$

Notations

Espaces

- $\mathcal{L}_f \subset \mathcal{R}$, espace des clés privées f,

$$\mathcal{L}_f = \{1 + pF \mid F \in \mathcal{B}(d_F)\}.$$

- $\mathcal{L}_g = \mathcal{B}(d_g) \subset \mathcal{R}$, espace des clés privés g.
- $\mathcal{L}_m \subset \mathcal{R}$, espace des messages $m \in \mathbb{Z}_p[X]/(X^N - 1)$.
- $\mathcal{L}_r = \mathcal{B}(d_r) \subset \mathcal{R}$, espace des polynômes aléatoires.

Paramètres de NTRU

Version	N	p	q	d_F	d_g	d_r
Sécurité modérée	167	2	128	32	31	32
Sécurité Standard	251	2	128	72	71	72
Haute Sécurité	347	2	128	64	173	64
Très Haute Sécurité	503	2	256	420	251	170

Notations

Espaces

- $\mathcal{L}_f \subset \mathcal{R}$, espace des clés privées f,

$$\mathcal{L}_f = \{1 + pF \mid F \in \mathcal{B}(d_F)\}.$$

- $\mathcal{L}_g = \mathcal{B}(d_g) \subset \mathcal{R}$, espace des clés privés g.
- $\mathcal{L}_m \subset \mathcal{R}$, espace des messages $m \in \mathbb{Z}_p[X]/(X^N - 1)$.
- $\mathcal{L}_r = \mathcal{B}(d_r) \subset \mathcal{R}$, espace des polynômes aléatoires.

Paramètres de NTRU

Version	N	p	q	d_F	d_g	d_r
Sécurité modérée	167	2	128	32	31	32
Sécurité Standard	251	2	128	72	71	72
Haute Sécurité	347	2	128	64	173	64
Très Haute Sécurité	503	2	256	420	251	170

Principe d'utilisation

BUT

Une personne **B** veut envoyer un message M à une personne **A** en utilisant le cryptosystème NTRU.

Etape 1 : Préparation de A

1. **A** choisit le niveau de sécurité, donc $N, p, q, \mathbb{L}_f, \cdots$.
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 $f_p * f \equiv 1 \pmod{p}$.
4. **A** calcule $f_q \in \mathcal{R}_q$ tel que $f_q * f \equiv 1 \pmod{q}$.
5. **A** calcule $h \in \mathcal{R}_q$ avec $h \equiv f_q * g \pmod{q}$.
6. **A** publie sa clé publique (h, q, p, N).
7. La clé secrète de **A** est (f, f_p).

Principe d'utilisation

BUT

Une personne **B** veut envoyer un message M à une personne **A** en utilisant le cryptosystème NTRU.

Etape 1 : Préparation de A

1. **A** choisit le niveau de sécurité, donc $N, p, q, \mathbb{L}_f, \cdots$.
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 $f_p * f \equiv 1 \pmod{p}$.
4. **A** calcule $f_q \in \mathcal{R}_q$ tel que $f_q * f \equiv 1 \pmod{q}$.
5. **A** calcule $h \in \mathcal{R}_q$ avec $h \equiv f_q * g \pmod{q}$.
6. **A** publie sa clé publique (h, q, p, N).
7. La clé secrète de **A** est (f, f_p).

Principe d'utilisation

B veut envoyer un message M à A

Etape 1 : Préparation de A: Calcule $h \equiv f_q * g \pmod{q}$ et envoie (h, q, p, N).

Etape 2 : Chiffrement par B

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

Principe d'utilisation

B veut envoyer un message M à A

Etape 1 : Préparation de A: Calcule $h \equiv f_q * g \pmod{q}$ et envoie (h, q, p, N).

Etape 2 : Chiffrement par B

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

Principe d'utilisation

B veut envoyer un message M à A

Etape 1 : Préparation de A: Calcule $h \equiv f_q * g \pmod{q}$ et envoie (h, q, p, N)

Etape 2 : Chiffrement par B: Envoie $e \equiv p * r * h + m \pmod{q}$.

Etape 3 : 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 veut envoyer un message M à A

Etape 1 : Préparation de A: Calcule $h \equiv f_q * g \pmod{q}$ et envoie (h, q, p, N)

Etape 2 : Chiffrement par B: Envoie $e \equiv p * r * h + m \pmod{q}$.

Etape 3 : 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 veut envoyer 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. $f_p * a = f_p * p * r * g + f_p * m * f \pmod p = m \pmod p$

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.

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 $g * h^{-1} = f \pmod{q}$.
2. Méthode: Tester tous les $g \in \mathbb{L}_g$ jusqu'à ce que $g * h^{-1} \pmod{q}$ 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 $g * h^{-1} = f \pmod{q}$.
2. Méthode: Tester tous les $g \in \mathbb{L}_g$ jusqu'à ce que $g * h^{-1} \pmod{q}$ 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.

Attaques basées sur la réduction des réseaux

Coppersmith-Shamir, 1998, basée sur LLL

- 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:

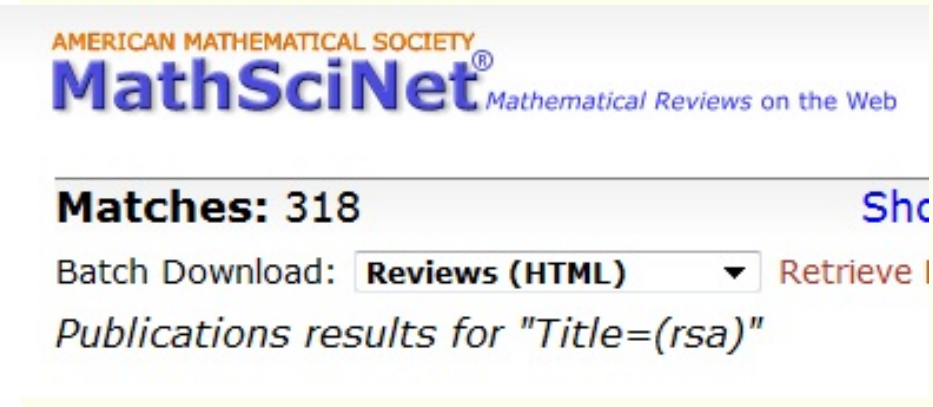
$$\left[\begin{array}{cccc|c} & & I_N & & 0_N \\ \hline h_0 & h_{N-1} & \cdots & h_1 & \\ h_1 & h_0 & \cdots & h_2 & \\ \vdots & \vdots & \ddots & \vdots & qI_N \\ h_{N-1} & h_{N-2} & \cdots & h_0 & \end{array}\right]$$

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

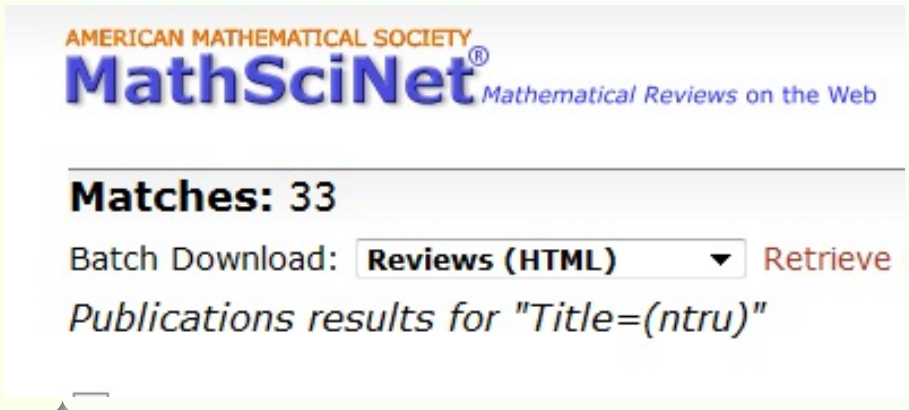
CONTENU

Comparaison

RSA dans Math Reviews



NTRU dans Math Reviews



Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique (n).
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010.
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

Comparaison

RSA

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^3)$.
- Taille de la clé publique (n).
- Standardizé dans une série de PKCS.
- Sécurité basée sur la factorisation.
- Vulnérable à l'ordinateur quantique.

NTRU

- Le chiffrement et le déchiffrement sont en $\mathcal{O}(n^2)$.
- Taille de la clé publique $\frac{N}{2}\log_2(N/4)$.
- Standardizé dans IEEE 1363.1-2008 et X9.98-2010
- Sécurité basée sur SVP ou CVP.
- Non vulnérable à l'ordinateur quantique.

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. $b_1, b_2, \cdots, b_n \in \mathbb{R}^n$, n vecteurs linéairement indépendants.
2. $\mathcal{B} = \{b_1, b_2, \cdots b_n\}$.
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. $\det(\mathcal{L}) = |\det(\mathcal{B})| = \text{volume}\,\{\sum_{i=1}^{n} \alpha_i v_i, \quad 0 \leq \alpha_i < 1\}$

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.
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 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 $B = \max_i(||b_i^{\text{init}}||)$.

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

CONTENU

Le futur

- Le présent est pour RSA, et très peu pour NTRU.
- Le futur est pour ... NTRU et un peu pour RSA.
- A moins que ECC soit le cryptosystème du futur.

Le futur

- Le présent est pour RSA, et très peu pour NTRU.
- Le futur est pour ... NTRU et un peu pour RSA.
- A moins que ECC soit le cryptosystème du futur.

Le futur

- Le présent est pour RSA, et très peu pour NTRU.
- Le futur est pour ... NTRU et un peu pour RSA.
- A moins que ECC soit le cryptosystème du futur.

Merci

شكرًا