CLES FAIBES POUR LE CRYPTOSYSTEME RSA

Abderrahmane NITAJ

Université de Caen
Département de Mathématiques
France

Oujda, 26 Avril 2007

CONTENU

CONTENU

Introduction

RSA

- RSA=Rabin+Shamir+Adleman.
- Inventé et breveté en 1977.
- Passé dans le domaine publique en 2000.
- Serveurs Web, Cartes de crédit, Paiement électronique, Téléphone portable.

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.
- $d \in \mathbb{N}$, $1 \leq d \leq \phi(N)$, $ed \equiv 1 \pmod{\phi(N)}$ 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 $1 < m < N$.
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)

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

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 $1 < m < N$.
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)

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

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 $1 < m < N$.
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)

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

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 $1 < m < N$.
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)

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

Nombre de nombres premiers

le module RSA-1024

- p et q sont deux nombres premiers $2^{511} < p, q < 2^{512}$.
- $N = pq$ est le module RSA.

Théorème des nombres premiers

- Le nombre de nombres premiers inférieurs à n est :

$$\pi(n) \approx \frac{n}{\log(n)}.$$

Pour RSA-1024

$$\pi(2^{512}) - \pi(2^{511}) \approx 1.88 \times 10^{151}$$

Nombre de nombres premiers

le module RSA-1024

- p et q sont deux nombres premiers $2^{511} < p, q < 2^{512}$.
- $N = pq$ est le module RSA.

Théorème des nombres premiers

- Le nombre de nombres premiers inérieurs à n est :

$$\pi(n) \approx \frac{n}{\log(n)}.$$

Pour RSA-1024

$$\pi(2^{512}) - \pi(2^{511}) \approx 1.88 \times 10^{151}$$

Nombre de nombres premiers

le module RSA-1024

- p et q sont deux nombres premiers $2^{511} < p, q < 2^{512}$.
- $N = pq$ est le module RSA.

Théorème des nombres premiers

- Le nombre de nombres premiers inérieurs à n est :

$$\pi(n) \approx \frac{n}{\log(n)}.$$

Pour RSA-1024

$$\pi(2^{512}) - \pi(2^{511}) \approx 1.88 \times 10^{151}$$

Factorisation du module par GNFS

GNFS

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

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

Pour le module RSA-1024

$$\text{Complexité} \approx \left(3.8 \times 10^{13}\right)^{(c+o(1))}.$$

Factorisation du module par GNFS

GNFS

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

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

Pour le module RSA-1024

$$\text{Complexité} \approx \left(3.8 \times 10^{13}\right)^{(c+o(1))}.$$

Le record-les défis

RSA-200 par GNFS

- RSA-200 : module RSA de 663 bits (200 chiffres).
- F. Bahr, M. Boehm, J. Franke et T. Kleinjung.
- le 9 mai 2005, Bonn, Allemagne.
- 80 PCs (2.2 GHz Opteron CPU).
- 10 mois.

Le défi RSA-2048

- RSA-2048 : module RSA de 2048 bits (617 chiffres).
- 200 000 US$.

Le record-les défis

RSA-200 par GNFS

- RSA-200 : module RSA de 663 bits (200 chiffres).
- F. Bahr, M. Boehm, J. Franke et T. Kleinjung.
- le 9 mai 2005, Bonn, Allemagne.
- 80 PCs (2.2 GHz Opteron CPU).
- 10 mois.

Le défi RSA-2048

- RSA-2048 : module RSA de 2048 bits (617 chiffres).
- 200 000 US$.

CONTENU

Clés faibles-Clés contraintes

Définition: Clé faible

e est une clé faible si à partir de e, on peut factoriser le module N en temps polynômial.

Définition: Clé contrainte

e est une clé contrainte s'il existe une fonction $F(p, q)$ qui vérifie:

1. e est en relation avec $F(p, q)$.
2. A partir de $F(p, q)$, on peut calculer p ou q en temps polynômial.

Clés faibles-Clés contraintes

Définition: Clé faible

e est une clé faible si à partir de e, on peut factoriser le module N en temps polynômial.

Définition: Clé contrainte

e est une clé contrainte s'il existe une fonction $F(p, q)$ qui vérifie:

1. e est en relation avec $F(p, q)$.
2. A partir de $F(p, q)$, on peut calculer p ou q en temps polynômial.

Exemples

Example (Wiener, 1990, (Boneh-Durfee, 2000))

- Relation $ed - k\phi(N) = 1$.
- Les clés faibles e avec $d < \frac{1}{3}N^{\frac{1}{4}}$ $\quad (d < N^{0.292})$.
- Clés contraintes avec $F(p, q) = \phi(N)$.

Example (Blömer-May, 2004)

- Relation $ex + y = k\phi(N)$.
- Les clés faibles e avec $x < N^{\frac{1}{4}}$, $|y| < N^{-\frac{3}{4}}ex$.
- Clés contraintes avec $F(p, q) = \phi(N)$.

Exemples

Example (Wiener, 1990, (Boneh-Durfee, 2000))

- Relation $ed - k\phi(N) = 1$.
- Les clés faibles e avec $d < \frac{1}{3}N^{\frac{1}{4}}$ $\quad (d < N^{0.292})$.
- Clés contraintes avec $F(p, q) = \phi(N)$.

Example (Blömer-May, 2004)

- Relation $ex + y = k\phi(N)$.
- Les clés faibles e avec $x < N^{\frac{1}{4}}$, $|y| < N^{-\frac{3}{4}}ex$.
- Clés contraintes avec $F(p, q) = \phi(N)$.

Les approximatios diophantiennes

Problème

$\theta \in \mathbb{R}^+$. Comment trouver $a, b \in \mathbb{N}$ tels que $\frac{a}{b} \approx \theta$?

Solution:

L'algorithme des fractions continues.

Expansion:

$$\theta = [a_0, a_1, a_2, a_3 \cdots] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\cdots}}}}$$

L'algorithme:

- $r_0 = \theta$, $a_0 = [r_0]$.
- $n \geq 1, \quad r_n = \frac{1}{r_{n-1} - a_{n-1}}, \quad a_n = [r_n]$,
- $[a_0, a_1, \cdots, a_s] = \frac{a}{b}$.
- $\frac{a}{b}$ est une convergente de θ.

Les approximatios diophantiennes

Problème

$\theta \in \mathbb{R}^+$. Comment trouver $a, b \in \mathbb{N}$ tels que $\frac{a}{b} \approx \theta$?

Solution:

L'algorithme des fractions continues.

Expansion:

$$\theta = [a_0, a_1, a_2, a_3 \cdots] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\cdots}}}}$$

L'algorithme:

- $r_0 = \theta$, $a_0 = [r_0]$.
- $n \geq 1, \quad r_n = \frac{1}{r_{n-1} - a_{n-1}}, \quad a_n = [r_n]$,
- $[a_0, a_1, \cdots, a_s] = \frac{a}{b}$.
- $\frac{a}{b}$ est une convergente de θ.

Les approximatios diophantiennes inverses

Problème

$\theta \in \mathbb{R}^+$, $a, b \in \mathbb{N}$. Peut-on savoir si $\frac{a}{b}$ est une réduite θ?

Solution: Le théorème de Legendre

Si $\left|\frac{a}{b} - \theta\right| < \frac{1}{2b^2}$, alors $\frac{a}{b}$ est une convergente de θ.

Les approximatios diophantiennes inverses

Problème

$\theta \in \mathbb{R}^+$, $a, b \in \mathbb{N}$. Peut-on savoir si $\frac{a}{b}$ est une réduite θ?

Solution: Le théorème de Legendre

Si $\left|\frac{a}{b} - \theta\right| < \frac{1}{2b^2}$, alors $\frac{a}{b}$ est une convergente de θ.

Les réseaux

Définition

1. $\mathcal{B} = \{v_1, v_2, \cdots v_m\} \in \mathbb{R}^m$, ensemble de m vecteurs linéairement indépendants. Le réseau engendré par $\mathcal{B}$ est

$$\mathcal{L} = \mathbb{Z}.v_1 \oplus \mathbb{Z}.v_2 \oplus \cdots \oplus \mathbb{Z}.v_m$$

2. Soit $v = \sum_{i=1}^{m} x_i v_i \in \mathcal{L}$. La norme de v est $||v|| = \sqrt{\sum_{i=1}^{m} x_i^2}$.

Caractéristiques

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

L'algorithme LLL

Problème

$\mathcal{L}$ est un réseau engendré par une base $\mathcal{B}$. Comment trouver $v \in \mathcal{L}$ tel que $||v||$ est assez petite?

Solution théorique: le théorème de Minkowski

Il existe un vecteur $v \in \mathcal{L}$ tel que $||v|| \leq \sqrt{m}\det(\mathcal{L})^{1/m}$.

Solution pratique: l'algorithme LLL

- Lenstra-Lenstra-Lovasz, 1982.
- Il existe un vecteur $v \in \mathcal{L}$ telq que $||v|| \leq 2^{\frac{m-1}{4}}\det(\mathcal{L})^{1/m}$.
- La complexité de l'algorithme LLL est polynômiale en m.

L'algorithme LLL

Problème

$\mathcal{L}$ est un réseau engendré par une base $\mathcal{B}$. Comment trouver $v \in \mathcal{L}$ tel que $||v||$ est assez petite?

Solution théorique: le théorème de Minkowski

Il existe un vecteur $v \in \mathcal{L}$ tel que $||v|| \leq \sqrt{m}\det(\mathcal{L})^{1/m}$.

Solution pratique: l'algorithme LLL

- Lenstra-Lenstra-Lovasz, 1982.
- Il existe un vecteur $v \in \mathcal{L}$ telq que $||v|| \leq 2^{\frac{m-1}{4}}\det(\mathcal{L})^{1/m}$.
- La complexité de l'algorithme LLL est polynômiale en m.

L'algorithme LLL

Problème

$\mathcal{L}$ est un réseau engendré par une base $\mathcal{B}$. Comment trouver $v \in \mathcal{L}$ tel que $||v||$ est assez petite?

Solution théorique: le théorème de Minkowski

Il existe un vecteur $v \in \mathcal{L}$ tel que $||v|| \leq \sqrt{m}\det(\mathcal{L})^{1/m}$.

Solution pratique: l'algorithme LLL

- Lenstra-Lenstra-Lovasz, 1982.
- Il existe un vecteur $v \in \mathcal{L}$ telq que $||v|| \leq 2^{\frac{m-1}{4}}\det(\mathcal{L})^{1/m}$.
- La complexité de l'algorithme LLL est polynômiale en m.

Deux théorèmes de Coppersmith

Théorème 1:

$f(x) \in \mathbb{Z}[x]$ est un polynôme de degré d. Si $f(x) \equiv 0 \pmod{N}$ a une solution x_0 avec $|x_0| < N^{1/d}$, alors on peut calculer x_0 en temps polynômial en $(\log N, d)$.

Théorème 2:

Soit $N = pq$ un module RSA. Si on connait une approximation $\bar{P}$ de p vérifiant $|\bar{P} - p| < N^{1/4}$, alors on peut calculer p en temps polynômial en $\log N$.

Deux théorèmes de Coppersmith

Théorème 1:

$f(x) \in \mathbb{Z}[x]$ est un polynôme de degré d. Si $f(x) \equiv 0 \pmod{N}$ a une solution x_0 avec $|x_0| < N^{1/d}$, alors on peut calculer x_0 en temps polynômial en $(\log N, d)$.

Théorème 2:

Soit $N = pq$ un module RSA. Si on connait une approximation $\bar{P}$ de p vérifiant $|\bar{P} - p| < N^{1/4}$, alors on peut calculer p en temps polynômial en $\log N$.

L'attaque de Wienner, 1990

Méthode

1. L'équation RSA $ed - k\phi(N) = 1$.
2. $\frac{k}{d} \approx \frac{e}{\phi(N)} \approx \frac{e}{N}$.
3. Si $d < \frac{1}{3}N^{\frac{1}{4}}$, alors $\frac{k}{d}$ est une réduite de $\frac{e}{N}$.
4. Clés faibles contraintes avec $\phi(N)$.

Variante

On peut appliquer la méthode avec $ed - kF(p,q) = 1$ et

$F(p,q) = (p+1)(q+1), (p-1)(q+1), (p+1)(q-1), N-(p+q), \cdots$

L'attaque de Wienner, 1990

Méthode

1. L'équation RSA $ed - k\phi(N) = 1$.
2. $\frac{k}{d} \approx \frac{e}{\phi(N)} \approx \frac{e}{N}$.
3. Si $d < \frac{1}{3}N^{\frac{1}{4}}$, alors $\frac{k}{d}$ est une réduite de $\frac{e}{N}$.
4. Clés faibles contraintes avec $\phi(N)$.

Variante

On peut appliquer la méthode avec $ed - kF(p,q) = 1$ et

$F(p,q) = (p+1)(q+1), (p-1)(q+1), (p+1)(q-1), N-(p+q), \cdots$

L'attaque de Boneh et Durfee, 2000

Méthode

1. L'équation RSA $ed - k\phi(N) = 1$.
2. Variante $k\left(\frac{N+1}{2} - \frac{p+q}{2}\right) + 1 \equiv 0 \pmod{e}$.
3. Utilisation de l'algorithme LLL.
4. Si $d < N^{0.292}$, alors on peut factoriser N.
5. Clés faibles contraintes avec $\phi(N)$.

L'attaque de Blömer et May, 2004

Méthode

1. L'équation $ex + y = k\phi(N)$.
2. $\frac{k}{x} \approx \frac{e}{\phi(N)} \approx \frac{e}{N}$.
3. Si $x < \frac{1}{3}N^{\frac{1}{4}}$ et $|y| < N^{-\frac{3}{4}}ex$, alors $\frac{k}{x}$ est une réduite de $\frac{e}{N}$.
4. $\phi(N) \approx \frac{ex}{k}$.
5. On détermine une approximation de p et on utlise le théorème de Coppersmith.
6. Clés faibles contraintes avec $\phi(N)$.

Clés contaraintes par $p(p-u)$

Méthode

1. L'équation $eY - p(q-u)X = Z$.
2. $\frac{X}{Y} \approx \frac{e}{p(q-u)} \approx \frac{e}{N}$.
3. Si X, Y et $|Z|$ sont "petits", alors $\frac{X}{Y}$ est une réduite de $\frac{e}{N}$.
4. $pu \approx N - \frac{eX}{Y}$.
5. On détermine pu en utilisant le théorème de Coppersmith, puis $p = \gcd(pu, N)$.

Clés contaraintes par $(p+1)(q-1)$

Méthode

1. L'équation $eY^m - (p+1)(q-1)X^m = Z$.
2. $\frac{X}{Y} \approx \frac{e^{1/m}}{((p+1)(q-1))^{1/m}} \approx \frac{e^{1/m}}{N^{1/m}}$.
3. Si X, Y et $|Z|$ sont "petits", alors $\frac{X}{Y}$ est une réduite de $\frac{e^{1/m}}{N^{1/m}}$.
4. $p - q \approx N - 1 - \frac{eX^m}{Y^m}$.
5. On détermine p en utilisant le théorème de Coppersmith.

Une utilisation de ECM (1)

Méthode

1. L'équation $eX + \phi(N)Y = NZ$.
2. Transformation $eX - N(Z - Y) = (p + q - 1)Y$.
3. $\frac{X}{Z - Y} \approx \frac{e}{N}$.
4. Si $|X|, |Y|$ et $|Z|$ sont "petits", alors $\frac{X}{Z - Y}$ est une réduite de $\frac{e}{N}$.
5. Reste à résoudre $(p + q - 1)|Y| = |eY - N(Z - Y)|$.

Une utilisation de ECM (2)

ECM

1. ECM=The Elliptic Curve Method for factoring.
2. H.W. Lenstra, 1985.
3. ECM permet de déterminer les petits facteurs premiers d'un entier n.
4. Complexité en $\mathcal{O}\left(\exp\left(\left(\sqrt{2}+o(1)\right)\sqrt{\log p \log\log p}\right)\right)$.
5. Records : B. Dodson, 2006, $p|10^{381}+1$ avec $p \approx 10^{67}$.

Une utilisation de ECM (3)

Résolution de $(p+q-1)|Y| = |eY - N(Z-Y)|$

1. Si tous les diviseurs de $|Y|$ sont inférieurs à 10^{40}, avec ECM, on trouve

$$M = |eY - N(Z-Y)| = p_1^{r_1} p_2^{r_2} \cdots p_s^{r_s} M', \quad p_i < 10^{40}.$$

2. Alors $|Y| = {p_1}^{x_1} {p_2}^{x_2} \cdots {p_s}^{x_s}$.
3. De plus, si $q < p < 2q$, alors $2\sqrt{N} < p+q < \frac{3\sqrt{2}}{2}\sqrt{N}$.
4. Avec $D_1 = \frac{M}{\frac{3\sqrt{2}}{2}\sqrt{N}}$ et $D_2 = \frac{M}{2\sqrt{N}}$, alors

$$D_1 < |Y| = {p_1}^{x_1} {p_2}^{x_2} \cdots {p_s}^{x_s} < D_2.$$

Une utilisation de ECM (4)

Résolution de $D_1 < |Y| = {p_1}^{x_1} {p_2}^{x_2} \cdots {p_s}^{x_s} < D_2$

1. Résoudre
   $\log D_1 < x_1 \log p_1 + x_2 \log p_2 + \cdots + x_s \log p_s < \log D_2$.
   - L'algorithme LLL (entier, de Weger, 1987).
   - L'algorithme PSLQ (Bailey-Ferguson, 1992).
2. Pour chaque solution $(x_1, x_2, \cdots, x_s)$, calculer
   $D = {p_1}^{x_1} {p_2}^{x_2} \cdots {p_s}^{x_s}$.
3. Tester si $p + q = \frac{M}{D} + 1$.

CONTENU

Le présent et le futur

Réalités

- Domine actuellement le marché.
- Aucune attaque ne remet en cause la sécurité de RSA.
- Forte confiance dans le milieu scientifique.

Précautions

- Générer des facteurs premiers aléatoires.
- Prendre des clés aléatoires.
- Vérifier que les clés publiques ne sont pas contraintes.
- Utiliser RSA 1024 jusqu'à 2011.
- Utiliser RSA 2048 à partir de 2012.