Cryptographie RSA

Introduction

Opérations

Attaques

Introduction

- **Historique:**
  - **Rivest Shamir Adleman** ou **RSA** est un algorithme asymétrique de cryptographie à clé publique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet.
  - Cet algorithme est fondé sur l'utilisation d'une paire de clés composée d'une clé publique pour chiffrer et d'une clé privée pour déchiffrer des données confidentielles

Introduction

- Fonctionnement général
  - *Alice* crée la paire de clés (public et privé), envoie sa clé public à Bob.
  - Bob chiffre message M avec ce clé, renvoyer message chiffré C à Alice.
  - Alice déchiffre C avec sa clé privée

Opérations

- Génération des clés
  - Choisir *p* et *q*, deux nombres premiers distincts
  - *n=pq*
  - Calculer l'indicatrice d'Euler de *n* : $\varphi(n) = (p-1)(q-1)$
  - Choisir un entier *e*, tel que $1 < e < \varphi(n)$ et premier avec , $\varphi(n)$ appelé « exposant de chiffrement ».
  - Calculer d : $de \equiv 1 \mod \varphi(n)$

$de \equiv 1 \mod \varphi(n) \iff$ de – k$\varphi(n)$ = 1

Extended Euclidean Algorithm : $ax + by = \gcd(a,b)$

  - (n,e) clé public
  - (n,d) clé privé

Opérations

- Encryptions

$$c \equiv m^e \text{ mod n}$$

- Décryptions

$$m \equiv c^d \text{mod n}$$

Car $c^d \equiv (m^e)^d \equiv m^{ed} \text{ mod n}$

$$ed = 1 + k\varphi(n).$$

$$m^{ed} \equiv m^{1+k\varphi(n)} \equiv m(m^k)^{\varphi(n)} \equiv m \text{ mod n}$$

Opérations

$$m(m^k)^{\varphi(n)} \stackrel{?}{\equiv} m \mod n$$

- Si m premier avec n, d'après le théorème d'Euler

$$a^{\varphi(n)} \equiv 1 \mod n$$

$$(m^k)^{\varphi(n)} \equiv 1 \mod n \Rightarrow m \times 1 \equiv m \mod n$$

Opérations

- Si m n'est pas premier avec n, d'après le théorème Chinese remainder,

$$x \equiv a_1 \mod n_1$$
$$x \equiv a_2 \mod n_2$$
.....
$$x \equiv a_k \mod n_K$$

Du coup, toutes les solutions *x* pour ce systeme sont congruentes modulo le produit $N = n1n2\ldots nk$

D'après ce théorème on peut séparer à deux:

$$m(m^k)^{(p-1)(q-1)} \equiv m \mod p$$
$$m(m^k)^{(p-1)(q-1)} \equiv m \mod q$$

Opérations

Supposons m n'est pas premier avec p, donc m est un multiple de p

$$m(m^k)^{(p-1)(q-1)} \equiv 0 \equiv m \mod p$$

Donc m est premier avec q, d'après le théorème d'Euler

$$m(m^{k(p-1)})^{(q-1)} \equiv m \mod q$$

Grâce au théorème Chinese remainder

$$m(m^k)^{(p-1)(q-1)} \equiv m \mod p \times q$$

Exemple

1. Choisir deux entier premier p = 7 et q = 19
2. $n = pq$; n = 7*19 = 133
3. $\varphi(n) = (p-1)(q-1)$; $\varphi(n)$= (7 - 1)(19 - 1) = 108
4. Choisir e > 1 premier avec 108; e = 5
5. Calculer d par $de \equiv 1 \mod \varphi(n)$

   d = 65 car $5 \cdot 65 = 325 = 1 + 3 \cdot 108$

Donc la clé public est (n=133, e=5);

la clé privé est (n=133, d=65)

Ensuite on prend un message m=6.

Exemple

- Chiffrer

$$c = 6^5 \bmod 133 = 62$$

- Déchiffrer

$$m = 62^{65} \bmod 133 = 6$$

On utilise la méthode ‘modulo exponentiation’

- $m = C^d \% n$
  $= 62^{65} \% 133$
  $= 62 * 62^{64} \% 133$
  $= 62 * (62^2)^{32} \% 133$
  $= 62 * 3844^{32} \% 133$
  $= 62 * (3844 \% 133)^{32} \% 133$
  $= 62 * 120^{32} \% 133$

Exemple

- $= 62 * 36^{16} \% 133$
  $= 62 * 99^{8} \% 133$
  $= 62 * 92^{4} \% 133$
  $= 62 * 85^{2} \% 133$
  $= 62 * 43 \% 133$
  $= 2666 \% 133$
  $= 6$

Attaques

- Mathématique Attaques: Factorisation un Grand nombre
  - Idée: retrouver p et q en factoriser le modulo N
  - Méthode Fermat
  - Méthode Euler
  - Méthode Pollard's Rho
- Implémentation Attaque: Obtention de physique implémentation d'un système cryptographie
  - Timing Attack

Mathématiques Attaques

- **Méthode Fermat**

$$n = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2$$

  - Choisir k, le plus petit entier tell que

$$k^2 > n$$

  - Augmenter k un par un jusqu'à ce qu'il existe un entier h

$$(k+g)^2 - n = h^2$$

  - Et n = (k+g+h)(k+g-h)

Mathématiques Attaques

- Méthode Pollard's Rho
  - Choisir r, s plus petit que N
  - Si pgcd(r-s,N) # 1, pgcd(r-s,N) est un diviseur de N
  - Sinon, choisir une autre couple (r,s)
  - En effet, a la prochaine itération r = f(r), s = (f(s))
  - En général f(x)= $x^2 + a$

Mathématiques Attaques

- **Résultats d'implémentions en Java**

Dell Inspiron D6000 processor 1.6 Ghz chips Intel Pentium M

Number of bit (N)	Method Rho	Fermat
10	0	31
20	94	219
25	200	13375
25	198	3375
25	188	765
32	282	899515
40	8000	
40	8456	

Implémentation Attaque

- Timing Attack :
  - But: retrouver la clé privé d
  - Algorithme de 'Square and Multiply' C^d mod n. d= $\sum_{i=0}^{t-1} d_i.2^{t-1-i}$

$$d = d_o d_1 .. d_{t-1}$$

    - z = C
    - for j = 1 to t-1 do
      - $z \equiv z^2 \bmod n$ (1)
      - if $d_j == 1$ then
    - $z \equiv z.C \bmod n$ (2)
      - endif
    - endfor
    - return z

Implémentation Attaque

- Attaques
  - Choisir 2 messages E F, tel que $E^3 < n$

    $F^2 < n < F^3$
  - Si $d_j = 1$ nous devons faire $E^2.E$ mod n et $F^2.F$ mod n .

    Et, encore $F^3 - n$
  - Si d_j =0 nous devons faire E^2 mod n et F^2 mod n
  - Conclusion, si temp(E) < temp(F) le bit est 1,

    0 dans le cas contraire.

Implémentation Attaque

- The messages E et F
  - Problème: difficile de décider le temps différent
  - En effet, on doit baser sur les probabilités
  - Choisir 2 séries E1..Ek, F1..Fk, calculer le temps moyen, et si temp(Emoy) < temp(Fmoy)+e c'est le bit 1
  - e peut être déterminée par expériences. En réalité le nombre k de messages est de taille 20000

Implémentation Attaque

- Résultats en Java (Dell Inspiron D6000 processor 1.6 Ghz chips Intel Pentium M)

Length of private key	Number of bits error
62	2
62	*4*
62	*5*
62	*6*
125	*44*
125	*30*
125	*36*
125	*30*
254	*50*
256	*60*
256	*49*
255	*52*
266	*48*

Implémentation Attaque

- Réalité

Key size	Result			
	without error correction		with error correction	
	sample size	speed	sample size	speed
64	1 500–6 500	> 20 bits/s	1 500–4 500	> 20 bits/s
128	12 000–20 000	2 bits/s	6 000–10 000	4 bits/s
256	70 000–80 000	1 bit/4s	40 000–50 000	1 bit/2s
512	±350 000–400 000	1 bit/65s	200 000 – 300 000	1 bit/37s

‘Document Pratical Timing Attack’

Conclusion

- Fort algorithme peut être détruit par une attaque ‘simple’