

Cryptographie avancée

Anne Garcia-Sánchez

M2i M1 - CCI Avignon

24 octobre 2024

Limites du chiffrement symétrique

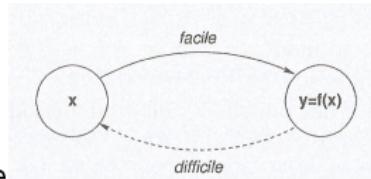
chiffrements sûrs, rapides

Problème: comment échanger la clé en toute sécurité?

Idée géniale 1



1976: Diffie et Hellman

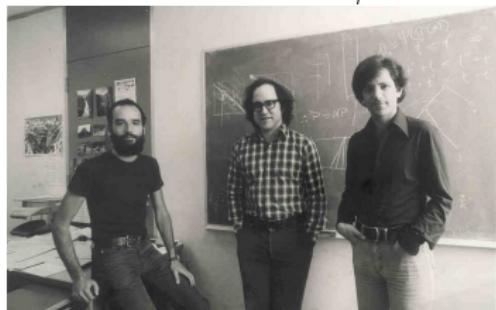


utilisation de fonctions à sens unique à trappe

clé publique permet de chiffrer mais pas de déchiffrer à moins de connaître la trappe - clé privée

Idée géniale 2

1978: Ronald Rivest, Adi Shamir et Leonard Adleman



premier exemple de fonction à sens unique (supposée) à trappe:
fonction *RSA*

s'appuie sur problème de factorisation des entiers

notions: nombres premiers, puissance modulaire, inverse modulaire

Chiffrement RSA

clé publique: deux entiers **N** et **e**

clé privée: un entier **d**

- chiffrement de l'entier m : calcul du chiffré $c \equiv m^e \pmod{N}$

avec m et c positifs et inférieurs à N

- déchiffrement du chiffré : $m \equiv c^d \pmod{N}$

ordres de grandeur

taille minimale de N : 2048 bits

e strictement supérieur à 2^{16} c'est à dire 65536

Protocole de chiffrement *RSA* simplifié

- **Génération des clés.**

On tire aléatoirement deux nombres premiers p et q distincts et de même longueur binaire.

On calcule $N = pq$ et $\varphi(N) = (p - 1)(q - 1)$.

On choisit un exposant public e premier avec $\varphi(N)$.

On calcule d l'**inverse de e modulo $\varphi(N)$** .

La clé publique est le couple (N, e) .

La clé secrète est l'entier d .

- **Chiffrement** du message m : calcul du chiffré $c = m^e \bmod N$
- **Déchiffrement** de c : calcul de $c^d \bmod N$ donne m

Un exemple simple avec des petits nombres

- choix de deux nombres premiers $p = 5, q = 11$
- $N = 5 \times 11 = 55$
- $\varphi(N) = (5 - 1) \times (11 - 1) = 4 \times 10 = 40$
- on choisit $e = 3$ premier avec 40
- on calcule $d = 27$, l'inverse de 3 modulo 40

clé publique: $(N, e) = (55, 3)$

clé privée: $(N, d) = (55, 27)$

- Chiffrement de $M = 20$
 $20^3 \equiv 25 \pmod{55} \rightarrow$ le chiffré est $C = 25$
- Déchiffrement de $C = 25$
 $25^{27} \equiv 20 \pmod{55} \rightarrow$ on retrouve $M = 20$

Inverse modulaire

L'inverse modulaire d'un entier relatif a modulo n est un entier u satisfaisant l'équation :

$$a \times u \equiv 1 \pmod{n}$$

u peut être noté a^{-1}

L'inverse de a modulo n existe si et seulement si a et n sont premiers entre eux c'est à dire qu'ils n'ont pas de diviseur commun autre que 1.

Inverse modulaire

Exemple: cherchons l'inverse modulaire de 2 modulo 7.

$$2 \times ?? \equiv 1 \pmod{7}$$

Inverse modulaire

$$2 \times 4 \equiv 1 \pmod{7}$$

```
>>> pow(2, -1, 7)
4
>>>
```

Comment calculer l'inverse de a modulo b ?

Identité de Bézout:

Soient a et b deux entiers relatifs.

Il existe deux entiers relatifs u et v tels que

$$a \times u + b \times v = PGCD(a, b).$$

u et v : coefficients de Bézout

algorithme d'Euclide → calcul de $PGCD(a, b)$

algorithme d'Euclide étendu → calcul de u, v et $PGCD(a, b)$

a et b premiers entre eux donc $PGCD(a, b) = 1$

$$a \times u + b \times v = 1 \text{ donc } a \times u = 1 - b \times v \text{ donc } a \times u \equiv 1 \pmod{b}$$

donc u est l'inverse de a modulo b

Algorithme d'Euclide

a et b entiers avec $a > b$, alors $\text{PGCD}(a, b) = \text{PGCD}(b, a \bmod b)$

Exemple: calcul du PGCD de 120 et 23

$$120 = 23 \times 5 + 5$$

$$23 = 5 \times 4 + 3$$

$$5 = 3 \times 1 + 2$$

$$3 = 2 \times 1 + 1$$

$$2 = 1 \times 2 + 0$$

Algorithme d'Euclide étendu

Cherchons l'inverse de 23 modulo 120

$$120 = 23 \times 5 + 5$$

$$23 = 5 \times 4 + 3$$

$$5 = 3 \times 1 + 2$$

$$3 = 2 \times 1 + 1$$

$$2 = 1 \times 2 + 0$$

r		=	u	x	a	+	v	x	b
120		=	1	x	120	+	0	x	23
23		=	0	x	120	+	1	x	23
5	= 120 - 5 x 23	=	1	x	120	+	-5	x	23
3	= 23 - 4 x 5 = 1 x 23	- 4 x (1 x 120 - 5 x 23)	=	-4	x	120	+	21	x 23
2	= 5 - 1 x 3 = (1 x 120 - 5 x 23)	- 1 x (-4 x 120 + 21 x 23)	=	5	x	120	+	-26	x 23
1	= 3 - 1 x 2 = (-4 x 120 + 21 x 23)	- 1 x (5 x 120 - 26 x 23)	=	-9	x	120	+	47	x 23

$$\text{donc } 1 = -9 \times 120 + 47 \times 23 \quad \text{ou} \quad 23 \times 47 = 1 + 9 \times 120$$

donc l'inverse de 23 modulo 120 est 47

Algorithme d'Euclide étendu

$$120 = 23 \times 5 + 5$$

$$23 = 5 \times 4 + 3$$

$$5 = 3 \times 1 + 2$$

$$3 = 2 \times 1 + 1$$

$$2 = 1 \times 2 + 0$$

r		=	u	x	a	+	v	x	b
120		=	1	x	120	+	0	x	23
23		=	0	x	120	+	1	x	23
5	= 120 - 5 × 23	=	1	x	120	+	-5	x	23
3	= 23 - 4 × 5 = 1 × 23	- 4 × (1 × 120 - 5 × 23)	=	-4	x	120	+	21	x 23
2	= 5 - 1 × 3 = (1 × 120 - 5 × 23)	- 1 × (-4 × 120 + 21 × 23)	=	5	x	120	+	-26	x 23
1	= 3 - 1 × 2 = (-4 × 120 + 21 × 23)	- 1 × (5 × 120 - 26 × 23)	=	-9	x	120	+	47	x 23

$$r_{i+1} = r_{i-1} - (r_{i-1} // r_i) r_i$$

$$u_{i+1} = u_{i-1} - (r_{i-1} // r_i) u_i$$

$$v_{i+1} = v_{i-1} - (r_{i-1} // r_i) v_i$$

Algorithme d'Euclide étendu: calcul des r_i

$$120 = 23 \times 5 + 5$$

quotients ↓ restes ↓
 r_{i-1}

$$23 = 5 \times 4 + 3$$

$$5 = 3 \times 1 + 2$$

r_{i-1} r_i $r_{i-1} // r_i$ r_{i+1}

$$3 = 2 \times 1 + 1$$

$$r_{i+1} = r_{i-1} - (r_{i-1} // r_i) r_i$$

Algorithme d'Euclide étendu: calcul des u_i

$$120 = 23 \times 5 + 5$$

$$23 = 5 \times 4 + 3$$

$$5 = 3 \times 1 + 2$$

$$3 = 2 \times 1 + 1$$

r	=	u	x	a	+	v	x	b
120	=	1	x	120	+	0	x	23
23	=	0	x	120	+	1	x	23
5	=	1	x	120	+	-5	x	23
3	=	-4	x	120	+	21	x	23
2	=	5	x	120	+	-26	x	23
1	=	-9	x	120	+	47	x	23

$$5 = 3 \times 1 + 2$$

$$2 = 5 - 1 \times 3 = (1 \times 120 - 5 \times 23) - 1 \times (-4 \times 120 + 21 \times 23) = 5 \times 120 - 26 \times 23$$

$$u_{i+1} = u_{i-1} - (r_{i-1} || r_i) u_i$$

Algorithme d'Euclide étendu: calcul des v_i

$$120 = 23 \times 5 + 5$$

$$23 = 5 \times 4 + 3$$

$$5 = 3 \times 1 + 2$$

$$3 = 2 \times 1 + 1$$

r	=	u	x	a	+	v	x	b
120	=	1	x	120	+	0	x	23
23	=	0	x	120	+	1	x	23
5	=	1	x	120	+	-5	x	23
3	=	-4	x	120	+	21	x	23
2	=	5	x	120	+	-26	x	23
1	=	-9	x	120	+	47	x	23

$$5 = 3 \times 1 + 2$$

$$2 = 5 - 1 \times 3 = (1 \times 120 - 5 \times 23) - 1 \times (-4 \times 120 + 21 \times 23) = 5 \times 120 - 26 \times 23$$

$$v_{i+1} = v_{i-1} - (r_{i-1} || r_i) u_i$$

Algorithme d'Euclide étendu

r							= u × a + v × b
120							= 1 × 120 + 0 × 23
23							= 0 × 120 + 1 × 23
5	= 120 - 5 × 23						= 1 × 120 + -5 × 23
3	= 23 - 4 × 5 = 1 × 23		- 4 × (1 × 120 - 5 × 23)				= -4 × 120 + 21 × 23
2	= 5 - 1 × 3 = (1 × 120 - 5 × 23) - 1 × (-4 × 120 + 21 × 23)		- 1 × (4 × 120 - 26 × 23)				= 5 × 120 + -26 × 23
1	= 3 - 1 × 2 = (-4 × 120 + 21 × 23) - 1 × (5 × 120 - 26 × 23)		- 1 × (9 × 120 + 47 × 23)				= -9 × 120 + 47 × 23

Algorithme d'Euclide étendu:

Entrée: entiers a, b

Sortie:

r entier,

u, v entiers relatifs tels que $r = PGCD(a, b)$ et $r = au + bv$

$r_0, u_0, v_0, r_1, u_1, v_1 \leftarrow a, 1, 0, b, 0, 1$

tant que $r_1 \neq 0$ faire

$q = r_0 // r_1$ (quotient entier)

$r_0, u_0, v_0, r_1, u_1, v_1 \leftarrow r_1, u_1, v_1, r_0 - q \times r_1, u_0 - q \times u_1, v_0 - q \times v_1$

fin tant que

Renvoyer (r_0, u_0, v_0)

Calcul de l'inverse modulaire avec Python

Calcul de $23^{-1} \pmod{120}$

```
>>> pow(23, -1, 120)
47
>>>
```