Arithmétique et Cryptographie, Partie I
Arithmétique

A. Bonnecaze & R. Rolland

Institut de Mathématiques de Luminy (IML)

Oujda, Mai 2009

Arithmétique et cryptographie

- Domaine de la cryptographie à clé publique
- Division, multiplication, exponentiation, extraction de racines, factorisation, test de primalité,...
- Les algos traditionnels se révèlent souvent trop lents lorsqu'on travaille avec des grands nombres
- La complexité d'un algorithme est primordiale
- But de la présentation :
  - Introduction à l'arithmétique pour pouvoir comprendre les primitives crypto
  - Présentation des principaux algorithmes utilisant des grands nombres

Plan

La transformation de Hörner binaire

Soit $S = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 \in \{0,1\}^8$, l'écriture binaire de S. Ainsi :

$$S = a_7 2^7 + a_6 2^6 + a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0.$$

L'algorithme de Hörner peut être vu comme l'entrée d'un tel nombre dans un registre à décalage.

(1) on décale les bits déjà entrés d'une position vers la gauche,

(2) on insère le bit suivant à la position la plus à droite, laissée libre par le décalage.

Notons S_i le nombre déjà entré, qui s'écrit en binaire $a_7 a_6 \cdots a_{7-i}$. Alors le nombre S_{i+1} obtenu à partir de S_i par transformation de Hörner s'écrit :

$$S_{i+1} = 2S_i + a_{7-i-1}.$$

La multiplication par 2 correspond au décalage à gauche des bits déjà entrés, et l'addition de a_{7-i-1} correspond à l'insertion du bit suivant à la place laissée libre.

La transformation de Hörner binaire

Si on décrit la suite des opérations on calcule successivement :

$$S_0 = a_7, S_1 = 2S_0 + a_6, \cdots, S_7 = 2S_6 + a_0,$$

ou encore :

$$S = S_7 = 2(2(2(2(2(2a_7 + a_6) + a_5) + a_4) + a_3) + a_2) + a_1) + a_0.$$

La transformation de Hörner binaire

L'algorithme

entrées : un tableau A de N bits
sortie : le nombre $S = \sum_{i=0}^{N-1} A[i]2^i$
$S \leftarrow 0$;
  $i \leftarrow N - 1$;
  Tantque $i \geq 0$ Faire
    $S \leftarrow 2 * S + A[i]$;
    $i \leftarrow i - 1$;
  retourner S ;

Complexité : N boucles (une mutiplication par 2 et une addition)
L'algorithme coute $0(N)$ opérations élémentaires

Généralisation

Basé sur l'égalité

$$\sum_{k=0}^{n} a_k X^{n-k} = (\cdots(((a_0 X + a_1)X + a_2)X + a_3)X + \cdots + a_n).$$

$$\begin{aligned} S_0(X) &= a_0, \\ S_1(X) &= XS_0(X) + a_1, \\ \cdots &= \cdots \\ S_n(X) &= XS_{n-1}(X) + a_n. \end{aligned}$$

Applications

L'algorithme de Hörner intervient dans de nombreuses situations :

1. évaluation d'un polynôme en un point,
2. traduction binaire - décimal,
3. division euclidienne,
4. calcul d'une puissance,
5. etc.

Exemple : Evaluation d'un polynôme en un point

Soit le polynôme

$$P(X) = 4X^9 + 3X^8 + 2X^7 + X^6 + 2X^5 + 3X^4 + 4X^3 + 2X^2 + X + 1$$

On souhaite déterminer $P(3)$

- Combien d'opérations faut-il si on utilise l'algorithme naïf ?
- Combien d'opérations faut-il si on utilise l'algorithme de Hörner ?

Division entière

Soit a et b entiers. On dit que a divise b, ou $a|b$ si $\exists d$ tq $b = ad$.

Théorème

On suppose que b est non nul. Alors, il existe un couple unique (q, r) de nombres entiers tels que :

$$\begin{cases} a = bq + r \\ 0 \leq r < |b| \end{cases}$$

La valeur $r \equiv a \mod b$ est appelée le reste de la division.

Théorème

Si $m|a$ et $m|b$, alors $m|(\alpha a + \beta b)$ pour tout entier α, β

Preuve : $a = rm$; $b = sm$. Ainsi, $\alpha a + \beta b = \alpha rm + \beta sm = m(\alpha r + \beta s)$

Algorithmes

L'algorithme le plus simple consiste à soustraire autant de fois b de a qu'il est possible, jusqu'à obtenir un reste $< b$.

```
division:=proc(a,b)
local r, q, u ;

r := a ;
q := 0 ;
while (r >= b)
  do
    r := r − b ;
    q := q + 1 ;
 od ;
u := [q, r] ;
return(u) ;
end ;
```

Algorithmes

Dans la pratique a est souvent très grand devant b

Par exemple
a de 1024 bits (donc de l'ordre de 2^{1024})
b de 128 bits.
Quel est le nombre de soustractions à faire ?

L'algorithme de Division binaire est plus efficace. Il est basé sur l'écriture binaire des nombres.

Division binaire

```
local r,q,aux,n,u ;
  r := a ; q := 0 ; n := 0 ; aux := b ;
  while (aux <= a)
     aux := 2 * aux ;
     n := n + 1 ;
  while (n > 0)
     aux := aux/2 ;
     n := n - 1 ;
     if (r < aux)
       then
         q := 2 * q ;
       else
         q := 2 * q + 1 ;
         r := r - aux ;
  u := [q, r] ;
  return(u) ;
end ;
```

Comparaison des deux algorithmes

- Le premier algorithme (Euclide pour la division) est inutilisable car le nombre de tours de boucles contenant des opérations simples est $O(a)$
- Le deuxième algorithme aboutit même pour de grands nombres, son nombre de tours de boucles contenant des opérations simples est $O(\ln(a))$.

Modulo

Si $n|(a-b)$ alors

$$(a \mod n) = (b \mod n)$$

On écrit

$$a \equiv b \mod n$$

et on dit que *a* est congru à *b* modulo *n*
Les entiers peuvent être divisés en *n* classes d'équivalences en fonction de leur résidu modulo *n*

$$[a]_n = \{a + kn,\ k \in \mathbb{Z}\}$$

$$\mathbb{Z}_n = \{[a]_n\ : 0 \leq a \leq n-1\}$$

ou simplement

$$\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$$

PGCD

Soit a, b des entiers dont au moins un est non nul

- $$pgcd(a,b) = max(d : d|a \text{ et } d|b)$$
- $$ppcm(a,b) = min(d > 0 : a|d \text{ et } b|d)$$
- a et b sont premiers entre eux si $pgcd(a,b) = 1$

Algorithme d'Euclide

Calcul du PGCD

$R0 := |a|$;
$R1 := |b|$; $(b \neq 0)$
Tantque $R1 > 0$ Faire
$R := Reste\_Division(R0, R1)$;
$R0 := R1$;
$R1 := R$;

En sortie $R1 = 0$, et $R0 = pgcd(a, b)$.
Les conditions :

$$\begin{cases} \text{L'ensemble des diviseurs communs de } R0 \text{ et } R1 \text{ est} \\ \quad \text{l'ensemble des diviseurs communs de } a \text{ et } b. \\ \\ R_1 \geq 0 \end{cases}$$

constituent un invariant de boucle.

Algorithme d'Euclide

Calcul du PGCD

$R0 := |a|$;
$R1 := |b|$; $(b \neq 0)$
Tantque $R1 > 0$ Faire
$R := Reste\_Division(R0, R1)$;
$R0 := R1$;
$R1 := R$;

En sortie $R1 = 0$, et $R0 = pgcd(a, b)$.
L'algorithme se termine car $R1$ décroît strictement à chaque tour de boucle. A la fin $R1 = 0$, donc l'ensemble des diviseurs de $R0$ et de $R1$ est l'ensemble des diviseurs de $R0$, et par conséquent $R0 = \text{pgcd}(a, b)$.

Exemple : $pgcd(53, 39)$

$$
\begin{array}{lccc}
pgcd & (R_0 & , & R_1) \\
pgcd & (53 & , & 39) \\
pgcd & (39 & , & 14) \\
pgcd & (14 & , & 11) \\
pgcd & (11 & , & 3) \\
pgcd & (3 & , & 2) \\
pgcd & (2 & , & 1) \\
pgcd & (1 & , & 0)
\end{array}
$$

$$pgcd(53, 39) = 1$$

Algorithme d'Euclide Etendu

$$
\begin{array}{rcl}
53 = 1.39 + 14 & \Rightarrow & 14 = 53 - 39 \\
39 = 2.14 + 11 & \Rightarrow & 11 = 39 - 2.14 = -2.53 + 3.39 \\
14 = 1.11 + 3 & \Rightarrow & 3 = 14 - 1.11 = 3.53 - 4.39 \\
11 = 3.3 + 2 & \Rightarrow & 2 = 11 - 3.3 = -11.53 + 15.39 \\
3 = 1.2 + 1 & \Rightarrow & 1 = 3 - 1.2 = 14.53 - 19.39 \\
2 = 2.1 + 0 & &
\end{array}
$$

On obtient $14.53 - 19.39 = 1$

Remarque :

on peut utiliser cet algorithme pour obtenir l'inverse de 39 modulo 53 :

$$39^{-1} \equiv -19 \mod 53 \ (\equiv 34 \mod 53)$$

Propriétés

Théorème

Soit a, b deux entiers non tous nuls et d le plus petit entier positif de $S = \{ax + by : x, y \in \mathbb{Z}\}$. Alors $pgcd(a, b) = d$

preuve : $|a| \in S$ donc S contient un entier positif.
Par définition, il existe x, y tq $d = ax + by$. $d \leq |a|$, donc il existe q, r tq

$$a = qd + r, \quad 0 \leq r < d$$

donc,

$$r = a - qd = a - q(ax + by) = a(1 - qx) + b(-qy) \in S.$$

$r < d$ implique que $r = 0$, et ainsi $d|a$. De même, on a $d|b$.
donc $d \leq gcd(a, b)$.
D'autre part $pgcd(a, b)|a$ et $pgcd(a, b)|b$, et ainsi $pgcd(a, b)$ divise toute combinaison linéaire de a, b, donc tout élément de S , donc d, et ainsi $pgcd(a, b) \leq d$. Donc $d = pgcd(a, b)$.

Propriétés (suite)

Corollaire : Bézout

a et b sont premirs entre eux $\iff \exists x,\ y \in \mathbb{Z}$ tel que $xa + yb = 1$.

Preuve :

($\Leftarrow$) Soit $d = pgcd(a, b)$, et $xa + xb = 1$. $d|a$ et $d|b$ et ainsi, $d|1$, donc $d = 1$.

($\Rightarrow$) $pgcd(a, b) = 1$. d'après le théo précédent, 1 est le plus petit entier positif dans $S = \{ax + by : x, y \in \mathbb{Z}\}$, cad. $\exists x, y$ tel que $ax + by = 1$.

Propriétés (suite)

Théorème fondamental de l'arithmétique

Si $c|ab$ et $pgcd(b,c) = 1$ alors $c|a$.

Preuve : On sait que $c|ab$. On a, $c|ac$.
Donc,
$c|pgcd(ab,ac) = a.pgcd(b,c) = a.1 = a$.

Entiers et nombres premiers

Définition

Un entier $p \geq 2$ est appelé un nombre premier s'il est divisible seulement par 1 et lui-même.

Théorème d'unique factorisation

Tout entier positif peut être représenté comme un produit de nombres premiers d'une manière unique (à permutation des nombres premiers près)

Preuve du théorème

Tout entier peut être représenté comme un produit de premiers car si un élément n'est pas premier, il peut être factorisé en nombres premiers plus petits.
Unicité : Supposons qu'un entier puisse être représenté de deux manières.

$$p_1 p_2 p_3 \dots p_s = q_1 q_2 q_3 \dots q_r$$

où tous les facteurs sont premiers, et $p_i \neq q_j$
Alors

$$p_1 | q_1 q_2 q_3 \dots q_r$$

Mais $pgcd(p_1, q_1) = 1$ et donc

$$p_1 | q_2 q_3 \dots q_r$$

en continuant, on obtient $p_1 | q_r$. Contradiction.

Groupes

Un groupe $(G, \oplus)$ est un ensemble G muni d'une opération binaire $\oplus$ tq

- Cloture : $a \oplus b \in G$ pour tout $a, b \in G$
- Identité : il existe un élément $e \in G$ tq $e \oplus a = a \oplus e = a$ pour tout $a \in G$
- Associativité : $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ pour tout $a, b, c \in G$
- Inverses : pour tout élément $a \in G$ il existe un unique élément $b \in G$ vérifiant $a \oplus b = b \oplus a = e$

Un groupe commutatif est appelé abélien

Définition

L'ordre du groupe, $|G|$, est le nombre d'éléments dans G.

Lemme : $(\mathbb{Z}_n, +_n)$ est un groupe fini abélien additif modulo n.

Propriétés

Soit $a^k = \underbrace{a \oplus a \oplus \ldots \oplus a}_{k}$ (notation multiplicative)

- $a^0 = e$
- e est unique
- tout élément $a \in G$ admet un et un seul inverse noté a^{-1}
- $a^{-k} = \oplus_{i=1}^{k} a^{-1}$
- $a^m \oplus a^n = a^{m+n}$
- $(a^m)^n = a^{mn}$

Ordre

L'ordre d'un élément a d'un groupe G est le plus petit $t > 0$ tel que $a^t = e$.
Notation $Ord(a, G)$ ou $Ord(a)$

Exemples

- L'ordre de 2 dans $(\mathbb{Z}_3, .)$ est 2
- Quel est l'ordre de 2 dans $(\mathbb{Z}_3, +)$?
- Quel est l'ordre de 2 dans $(\mathbb{Z}_7, .)$?

Sous-groupes

Si $(G, \oplus)$ est un groupe, $G' \subseteq G$, et $(G', \oplus)$ est aussi un groupe, alors $(G', \oplus)$ est appelé un sous-groupe de G.

Théorème

Si $(G, \oplus)$ est un groupe fini et G' un sous ensemble de G tel que $a \oplus b \in G'$ pour tout $a, b \in G'$, alors $(G', \oplus)$ est un sous-groupe de $(G, \oplus)$

Exemples

- $(\{0, 2, 4, 6\}, +_8)$ est un sous-groupe de $(Z_8, +_8)$
- $(\mathbb{Z}_3, +_3)$ est-il un sous-groupe de $(\mathbb{Z}_6, +_6)$?
- $(\{0, 2, 4\}, .6)$ est-il un sous-groupe de $(\mathbb{Z}_6, .6)$?

Lagrange

Si $(G, \oplus)$ est un groupe fini et $(G', \oplus)$ est un sous-groupe de $(G, \oplus)$, alors $|G'|$ divise l'ordre de $|G|$

Générateurs

Soit $g \in G$, on note $\langle g \rangle = \{g^k, 1 \leq k \leq Ord(g)\}$

Théorème

$\langle g \rangle$ admet $Ord(g)$ éléments distincts

Preuve (par contradiction) suppose qu'il existe $1 \leq i < j \leq Ord(g)$, tel que $g^i = g^j$. Ainsi, $e = g^{j-i}$ or $Ord(g) > j - i > 0$.

Lemme

$\langle g \rangle$ est un sous-groupe de G

$\langle g \rangle$ est appelé un groupe cyclique, il est engendré par g.
g est appelé un générateur de$\langle g \rangle$.

Exemples

- $\{0, 2, 4, 6\} \subset \mathbb{Z}_8$ peut être généré par 2 ou 6
- 3 est-il un générateur de $\mathbb{Z}_4$?

Propriétés des sous-groupes

- L'ordre d'un élément divise l'ordre du groupe
- Tout groupe d'ordre premier est cyclique
- Soit G un groupe fini et $a \in G$, alors $a^{|G|} = e$

Théorème

Soit $a \in G$ tel que $a^s = e$, alors $Ord(a)|s$

Preuve On a $s = q.Ord(a) + r$, où $0 \leq r < Ord(a)$.
Ainsi,

$$e = a^s = a^{q.Ord(a)+r} = (a^{Ord(a)})^q \oplus a^r = a^r.$$

Puisque $Ord(a)$ est minimal, on a $r = 0$.

Corps

Définition

Un corps $(F, \oplus, \otimes)$ est un ensemble muni de deux opérations binaires $\oplus$ et $\otimes$, et deux éléments 0 et 1, ayant les propriétés suivantes :

- $(F, \oplus)$ est un groupe abélien (0 est l'identité)
- $(F \backslash \{0\}, \otimes)$ est un groupe abélien (1 est l'identité)
- Distributivité : $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

Exemples

- $(\mathbb{Q}, +, .)$, $(\mathbb{Z}_p, +, .)$, p premier, sont des corps
- Les structures suivantes sont-elles des corps : $(\mathbb{Z}_6, +, .)$, $(\{1, 2, 3, 4\}, +_5, .5)$?

Inverses

Définition : Soit a un entier. Il est inversible si il existe b tel que

$$ab \equiv 1 \mod n$$

Dans ce cas b est l'inverse de a modulo n. On écrit $b \equiv a^{-1} \mod n$.

Théorème

Si $pgcd(a, n) = 1$, il existe b tel que $ab \equiv 1 \mod n$

Preuve D'après Bézout, il existe $x, y \in \mathbb{Z}$ tel que $xa + yn = 1$.
Donc, $xa \equiv 1 \mod n$.

Conclusion : a admet un inverse modulo n SSi $pgcd(a, n) = 1$. L'inverse peut être calculé avec l'algo d'Euclide étendu

Le groupe multiplicatif $\mathbb{Z}_n^*$

Définition

L'ensemble des entiers inversibles modulo n est noté $\mathbb{Z}_n^*$

$$\mathbb{Z}_n^* = \{i \in \mathbb{Z}_n \quad : \quad pgcd(i, n) = 1\}$$

Exemples

- Si p premier, $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$
- Quel est l'ordre de $\mathbb{Z}_p^*$?
- Quels sont les éléments de $\mathbb{Z}_{15}^*$?
- Peut-on comparer $|\mathbb{Z}_{15}^*|$ avec $|\mathbb{Z}_3^*|$ et $|\mathbb{Z}_5^*|$?
- Existe-t-il un entier n tq $\mathbb{Z}_n^* = \mathbb{Z}_n$?

Fonction d'Euler

Définition

La fonction d'Euler $\phi(n)$ représente le nombre d'éléments dans $\mathbb{Z}_n^*$

$$\phi(n) = |\mathbb{Z}_n^*| = |\{i \in \mathbb{Z}_n \ : \ pgcd(i, n) = 1\}|$$

$\phi(n)$ est le nombre d'éléments de $\{0, \ldots, n-1\}$ qui sont premiers avec n
$\phi(1) = 1$ car $\mathbb{Z}_1^* = \{0\}$

Exemples

- Que vaut $\phi(23)$?
- Que vaut $\phi(64)$?

Fonction d'Euler

Théorème

Soit $n = p_1^{e_1} p_2^{e_2} \cdots p_l^{e_l}$ l'unique factorisation de n. Alors

$$\phi(n) = \prod(p_i^{e_i - 1}(p_i - 1)) = n \prod(1 - \frac{1}{p_i})$$

Si la factorisation de n n'est pas connue, on ne sait pas déterminer $\phi(n)$

- $\phi(p) = p - 1$
- $\phi(p^e) = (p-1)p^{e-1} = p^e - p^{e-1}$
- $\phi(pq) = (p-1)(q-1)$
- Si $pgcd(a, b) = 1$ alors $\phi(ab) = \phi(a)\phi(b)$

Fonction d'Euler

- Soit $n = pq$ le produit de deux premiers
- Connaissant n et $\phi(n)$, peut-on retrouver p et q ?

Théorème d'Euler

Pour tout a et n, si $pgcd(a, n) = 1$ alors

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

Preuve On sait que $a \in \mathbb{Z}_n^*$. Donc on sait que $a^{|\mathbb{Z}_n^*|} = a^{\phi(n)} = 1$

Petit théorème de Fermat

Soit p un nombre premier, pour tout entier a, on a

$$a^p \equiv a \mod p$$

Exercices

- Calculer $8^{11} \mod 11$, $7^{11} \mod 11$, $5^8 \mod 7$, $14^{16} \mod 17$, $67^{258} \mod 68$, $3^{45} \mod 5$, $7^{31} \mod 11$, $4^{25} \mod 35$, $7^8 \mod 30$.
- Montrer que Z_5 peut s'écrire sous la forme $Z_5 = \{0\} \cup \{g^0, g^1, g^2, g^4\}$.
- Chiffrement RSA (avec des petits nombres) : Bob désire chiffrer un message pour Alice.
  - Alice choisit p et q premiers et calcule $n = pq$. (elle choisit 3.11)
  - Alice choisit alors un entier $d < n = 33$ (elle choisit $d = 7$). d est la clé privée d'Alice.
  - Alice calcule sa clé publique e telle que $ed = 1 \mod (\phi(n))$. Quelle est la valeur de e ?
  - Bob chiffre en calculant $C = m^e \mod n$ et Alice déchiffre en calculant $D = C^d \mod n$. Pourquoi D est-il égal à m ?
  - Bob chiffre le message BONJOUR. On utilise le codage suivant A=01, B=02, C=03, ..., J=10,...
  - Bob chiffre chaque lettre du message avec la clé publique. Quel est le résultat ?

Générateurs de $\mathbb{Z}_n^*$

- Si $\mathbb{Z}_n^*$ admet un générateur, alors $\mathbb{Z}_n^*$ est cyclique
- $\mathbb{Z}_n^*$ admet un générateur SSI $n = 2, 4, p^k$, où $2p^k$ (p premier)
- Si α est un générateur de $\mathbb{Z}_n^*$, alors

$$\mathbb{Z}_n^* = \{\alpha^i \mod n : 0 \leq i \leq \phi(n) - 1\}$$

- Si α est un générateur de $\mathbb{Z}_n^*$, alors $b = \alpha^i \mod n$ est aussi générateur SSI $PGCD(i, \phi(n)) = 1$
- Si $\mathbb{Z}_n^*$ est cyclique, alors le nombre de générateurs de $\mathbb{Z}_n^*$ est $\phi(\phi(n))$

Calcul de l'inverse modulaire

Soit un entier a. Supposons que $\phi(n)$ soit connu. Alors

$$a^{\phi(n)} \equiv a.a^{\phi(n)-1} \equiv 1 \mod n$$

Donc

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

Exemple Soit $n = 15$ $a = 11$, $\phi(15) = 8$,

$$a^{-1} \mod n \equiv 11^7 \mod 15 \equiv 11$$

En général cette méthode n'est pas plus rapide que l'algorithme d'Euclide étendu.

Restes Chinois

Comment travailler avec plusieurs modules ?

Théorème

Si m et n sont premiers entre eux alors la condition :

$$\begin{cases} a \equiv b & (m) \\ a \equiv b & (n) \end{cases}$$

est équivalente à :

$$a \equiv b \quad (mn).$$

Preuve Si $a \equiv b \quad (mn)$ les deux relations $a \equiv b \quad (m)$ et $a \equiv b \quad (n)$ ont bien lieu.
Réciproquement si ces deux relations ont lieu alors il existe k_1 et k_2 tels que :

$$a - b = k_1 m = k_2 n.$$

On voit alors que m divise $k_2 n$ et comme il est premier avec n il divise k_2. Si bien que :

$$a - b = k_3 mn.$$

Restes Chinois : un exemple numérique

Résoudre

$$\begin{cases} x \equiv 2 & \mod 3 \\ x \equiv 3 & \mod 5 \end{cases}$$

	z	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$z \mod 3$		0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
$z \mod 5$		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4

Le couple $(2, 3)$ correspond à $z = 8$

Comment résoudre le problème sans calculer le tableau ?

- voir la suite du cours
- (pour les paresseux) On peut utiliser le logiciel magma : $CRT([2, 3], [3, 5])$;

Restes Chinois

Soient m et n premiers entre eux. On cherche toutes les solutions entières de :

$$\begin{cases} x \equiv a & (m) \\ x \equiv b & (n) \end{cases}$$

On considère u et v tels que $um + vn = 1$.

Théorème des restes chinois

On obtient une solution en prenant :

$$x = bum + avn.$$

Toutes les solutions sont alors de la forme :

$$x + kmn.$$

Preuve du théorème

Par un calcul direct on vérifie que $x = bum + avn$ est bien une solution. On vérifie alors que pour tout entier k, $x + kmn$ est aussi une solution.
Si maintenant x et y sont deux solutions, par différence on obtient :

$$\begin{cases} x \equiv y & (m) \\ x \equiv y & (n) \end{cases},$$

ce qui nous permet de conclure :

$$y = x + kmn$$

grâce au théorème précédent
Remarque Il y a donc une solution unique y vérifiant $0 \leq y < mn$; ce qui peut s'exprimer encore en disant qu'il y a une unique solution dans $\mathbb{Z}/mn\mathbb{Z}$.

Généralisation

Théorème

Si les entiers $n_1, n_2, \ldots, n_k$ sont deux à deux premiers entre eux, alors le système :

$$\begin{cases} x \equiv a_1 \mod n_1 \\ x \equiv a_2 \mod n_2 \\ \vdots \\ x \equiv a_k \mod n_k \end{cases}$$

a une solution unique modulo $n = n_1 n_2 \ldots n_k$.

$$x \equiv \sum_{i=1}^{k} a_i N_i M_i \mod n,$$

avec $N_i = n/n_i$ et $M_i = N_i^{-1} \mod n_i$

Application : Attaque sur RSA

- Stephane doit envoyer le même message M à Alice, Claire et Corinne
- les clés publiques sont respectivement $(n_1 = 26, e = 7)$, $(n_2 = 35, e = 7)$, $(n_3 = 33, e = 7)$, où n_i sont les modules RSA.
- Stephane envoie les valeurs $C_1 = 24$, $C_2 = 23$, $C_3 = 29$.
- Comment Estelle va-t-elle procéder pour retrouver M après avoir intercepté les trois valeurs et les clés publiques ?

RQ

Soit $a \in \mathbb{Z}_n^*$; a est un résidu quadratique modulo n s'il existe $x \in \mathbb{Z}_n^*$ tel que $x^2 = a$ ($x^2 \equiv a \mod n$). Si un tel x n'existe pas, a est un non-résidu quadratique modulo n.
L'ensemble de tous les RQ modulo n est noté Q_n et celui des non-RQ, $\overline{Q}_n$.
Par définition $0 \notin \mathbb{Z}_n^*$, et donc $0 \notin Q_n$ et $0 \notin \overline{Q}_n$.

Proposition

Soit $p > 2$ un nombre premier et a un générateur de $\mathbb{Z}_p^*$. Alors $b \in \mathbb{F}_p$ est un RQ modulo p s'il existe un entier pair i tel que : $b \equiv a^i \mod p$. Il s'ensuit que :

$$|Q_p| = |\overline{Q}_p| = \frac{p-1}{2}.$$

La moitié des éléments sont des résidus et l'autre moitié des non-résidus.

RQ

Exercice : Un générateur de $\mathbb{Z}_p^*$ peut-il être un RQ modulo n ?

Critère d'Euler : x est un résidu quadratique modulo p, premier, ssi :

$$x^{\frac{p-1}{2}} \equiv 1 \pmod{p}.$$

Exemple. 6 est un générateur de $\mathbb{Z}_{13}^*$. Les puissances de 6 sont :

i	0	1	2	3	4	5	6	7	8	9	10	11
6^i	1	6	10	8	9	2	12	7	3	5	4	11

Ainsi $Q_{13} = \{1, 3, 4, 9, 10, 12\}$ et $\overline{Q}_{13} = \{2, 5, 6, 7, 8, 11\}$.

RQ

Proposition

Soit $n = p\,q$, p et q étant premiers distincts. Alors $a \in \mathbb{Z}_n$ est un RQ modulo n ssi la classe de a modulo p appartient à Q_p et celle de a modulo q à Q_q.

Dans la situation précédente on a :

$$\begin{aligned} |Q_n| &= |Q_p|.|Q_q| \\ &= \frac{(p-1)(q-1)}{4} \\ |\overline{Q}_n| &= 3\,\frac{(p-1)(q-1)}{4}\,. \end{aligned}$$

Exemple. $Q_{21} = \{1, 4, 16\}$, $\overline{Q}_{21} = \{2, 5, 8, 10, 11, 13, 17, 19, 20\}$.

Symboles de Legendre et Jacobi

Le symbole de Legendre permet de savoir si $a \in \mathbb{Z}$ est un RQ modulo p

Définition.

Soit $p > 2$ un premier et $a \in \mathbb{Z}$. Le symbole de Legendre $(\frac{a}{p})$ est ainsi défini :

$$\left(\frac{a}{p}\right) = \begin{cases} 0 \text{ si } p|a\,, \\ 1 \text{ si } a \in Q_p\,, \\ -1 \text{ si } a \in \overline{Q}_p\,. \end{cases}$$

a est un RQ modulo p si et seulement si le symbole de Legendre est égal à 1.

Propriétés du symbole de Legendre

- $(\frac{a}{p}) \equiv a^{(p-1)/2} \pmod{p}$. En particulier, $(\frac{1}{p}) = 1$ et $(\frac{-1}{p}) = (-1)^{(p-1)/2}$. Ainsi, $-1 \in Q_p$ si $p \equiv 1 \pmod 4$ et $-1 \in \overline{Q}_p$ si $p \equiv 3 \pmod 4$.
- $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$. Donc, si $a \in \mathbb{Z}_n$, $a \neq 0$, alors $(\frac{a^2}{p}) = 1$.
- $a \equiv b \pmod p \Rightarrow (\frac{a}{p}) = (\frac{b}{p})$.
- $(\frac{2}{p}) = (-1)^{(p^2-1)/8}$. Donc, $(\frac{2}{p}) = 1$ si $p \equiv \pm 1 \pmod 8$ et $(\frac{2}{p}) = -1$ si $p \equiv \pm 3 \pmod 8$.
- Loi de réciprocité quadratique : si $q \neq p$ est premier impair, alors :

$$(\frac{p}{q}) = (-1)^{\frac{(p-1)(q-1)}{4}} (\frac{q}{p}).$$

Exercice : Que vaut $(\frac{33}{47})$?

Symbole de Jacobi (généralisation à entiers impairs)

Soit $n \geq 3$ un entier impair dont la factorisation en puissances de premiers est :

$$n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k} .$$

Alors le symbole de Jacobi est ainsi défini :

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \dots \left(\frac{a}{p_k}\right)^{e_k}$$

Si n est premier on retrouve le symbole de Legendre.

Attention

Contrairement au symbole de Legendre, le symbole de Jacobi ne dit pas si un nombre est RQ modulo n (non premier).
Il est vrai que si a est un RQ, alors $(\frac{a}{n}) = 1$, mais la réciproque est fausse.

Racine carrée

Définition.

$x \in \mathbb{Z}_n^*$ est une racine carrée de $a \in Q_n$ si $x^2 \equiv a \pmod{n}$

Théorème

Nombre de racines carrées :

(a) *Si $p > 2$ est premier, $a \in Q_p$ a exactement deux racines modulo p.*

(b) *Si $n = pq$, p et q premiers, impairs, distincts, alors, si $a \in Q_n$, a possède exactement 4 racines carrées distinctes modulo n.*

Exemple. Racines carrées de 16 dans $\mathbb{Z}_{65}$?

$65 = 5.13$

Si x racine de 16 dans $\mathbb{Z}_{65}$, $x \mod 5$ est une racine de 16 modulo 5

$16 \mod 5 \equiv 1$

d'où $S = \{\pm 4, \pm 9\}$

Entiers de Blum

Définition. Un entier de Blum est le produit de deux premiers distincts congrus à 3 modulo 4

Exemple : 21 ou 77.

Proposition

Soient $n = p\,q$ un entier de Blum et $a \in Q_n$. Alors a possède exactement quatre racines carrées (modulo n), dont l'une est un carré (appartient à Q_n).

Ils sont par exemple utilisés dans le cryptosystème de Rabin pour simplifier le calcul de la racine carré.

Racine carré de $a \mod p$, avec $p \equiv 3 \mod 4$

On a

$$p \equiv 3 \mod 4$$

Alors

$$x_1 \equiv a^{\frac{p+1}{4}} \mod p$$

est une racine carrée

Preuve

$$x_1^2 \equiv a^{\frac{p+1}{2}} \equiv a^{\frac{p-1}{2}}.a$$

a étant un carré, on a

$$a^{\frac{p-1}{2}} \equiv 1 \mod p$$

d'après le petit théorème de Fermat. Donc

$$x_1^2 \equiv a \mod p$$

Racine carré de a modulo p avec $p \equiv 1 \mod 4$

on choisit m non carré, on pose $p-1 = 2^s t$, t impair et :

$$z = m^t,\ B = a^t,\ X = a^{\frac{t+1}{2}},\ Y = z,\ R = s-1\,.$$

Puisque m n'est pas un carré, on a :

$$z^{2^{s-1}} = m^{t2^{s-1}} = m^{\frac{p-1}{2}} \equiv -1 \mod p$$

Algorithme de Shank

Tant Que $R \geq 1$:
  Si $B^{2^{R-1}} \equiv 1 \mod p$,
    $Y = Y^2$,
  Sinon
    $B = BY^2$,
    $X = XY$,
    $Y = Y^2$,
  $R = R - 1$.

Exemple : racine carrée de 15 modulo 17.

boucle	B	X	Y	R
1	-1	11	9	2
2			13	1
3	1	7	-1	0

Autres algorithmes utiles

- Exponentiation modulaire
- Test de primalité
- Construction de nombres premiers
- Calcul du logarithme discret

Exponentiation modulaire : Algo square and multiply

Il s'agit de calculer dans un groupe a^n (ou na en notation additive).

Hörner
entrées : un tableau N de $K+1$ bits représentant n en binaire
sortie : le nombre $n = \sum_{i=0}^{K} N[i]2^i$

$n \leftarrow 0$;
$i \leftarrow K$;
Tantque $i \geq 0$ faire
  $n \leftarrow 2 * n$;
  si $N[i] == 1$
    alors $n \leftarrow n + 1$;
  $i \leftarrow i - 1$;
retourner n ;

Square and multiply
entrées : un tableau N de $K+1$ bits représentant n en binaire
sortie : le nombre $P = a^n$

$P \leftarrow 1$;
$i \leftarrow K$;
tq $i \geq 0$ faire
  $P \leftarrow P^2$;
  si $N[i] == 1$
    alors $P \leftarrow P * a$;
  $i \leftarrow i - 1$;
retourner P ;

Square and multiply : aspect récursif

fonction *puissance*(a, n)

si n vaut 0
  alors retourner 1
  sinon si n est pair
    alors retourner $(puissance(a, n/2))^2$
    sinon retourner $a * (puissance(a, (n-1)/2))^2$

Comparaison entre Hörner et square and multiply :
leurs nombres d'itérations sont identiques.
le nombre de boucles $=$ taille de n
A chaque boucle on fait une élévation au carré plus éventuellement une multiplication par a.
Si on note $t = \lfloor \ln(n) + 1 \rfloor$ la taille de n,
L'algorithme utilise $O(t)$ opérations simples.

Test de primalité

En cryptographie à clé publique, on a besoin de grands nombres premiers.
Déterminer si un nombre est premier est appelé le problème "Prime" noté $\mathcal{P}$.
On sait depuis 2002 que ce problème est polynomial
Mais en pratique,

- on utilise un test probabiliste de non-primalité
- et si besoin est, on lance un algorithme déterministe pour confirmer que le candidat premier trouvé par l'algorithme probabiliste est réellement premier

Test de non-primalité de Miller-Rabin

D'après le petit théorème de Fermat,

$$p \text{ premier} \Longrightarrow a^{p-1} \equiv 1 \mod p$$

Mais la réciproque est fausse : $561 = 3.11.17$ passe le test de Fermat (nombre de Carmichael)
Le test de Miller-Rabin améliore la méthode

- s'il répond qu'un nombre n'est pas premier alors il est sûr que ce nombre ne l'est pas
- Il peut aussi répondre qu'un nombre est probablement premier

Les témoins de Miller

Théorème

Soit n un entier impair > 1. Posons $n - 1 = 2^s t$ avec t impair. S'il existe a $(1 < a < n)$ tel que :

$$a^t \not\equiv 1 \mod n$$

et :

$$a^{2^i t} \not\equiv -1 \mod n \quad \forall i = 0, \cdots, s-1,$$

alors n n'est pas premier.

Preuve Supposons n premier. Pour $i = 0, \cdots, s$ posons $a_i = a^{2^i t} \mod n$. D'après le petit théorème de Fermat $a_s = 1$. Dans ces conditions ou bien tous les a_i valent 1 et dans ce cas a_0 vaut 1, ou bien il existe un i tel que $0 \leq i < s$, $a_i \neq 1$ et $a_{i+1} = 1$. Dans ce cas, puisque $a_{i+1} \equiv a_i^2 \mod n$ et que $\mathbb{Z}/n\mathbb{Z}$ est un corps on en déduit que $a_i \equiv -1 \mod n$.

Les témoins de Miller

- Si a vérifie les conditions du théorème
- il apporte une preuve de la non-primalité de n
- a s'appelle un témoin de Miller relatif à n.

Idée : utiliser le théorème pour détecter si un nombre est premier ou tout au moins s'il a de bonnes chances de l'être.

Le théorème de Rabin permet de majorer pour un entier n composé, le nombre d'éléments strictement compris entre 1 et n qui ne sont pas des témoins de Miller.

Théorème (Théorème de Rabin)

Soit n un entier impair composé > 9. Posons $n - 1 = 2^s t$ avec t impair. Les entiers $1 < a < n$ qui satisfont à la condition :

$$a^t \equiv 1 \mod n,$$

ou bien à l'une des conditions :

$$a^{2^i t} \equiv -1 \mod n \quad (0 \leq i \leq s-1),$$

sont en nombre au plus :

$$\frac{\Phi(n)}{4}.$$

Test de Miller-Rabin

On choisit a au hasard ($a < n$) et on calcule :

$$a^t \mod n.$$

Si on trouve 1 alors a n'est pas un témoin de Miller pour n, sinon on calcule les nombres :

$$a^{2^i t} \mod n,$$

et si pour un certain i on trouve -1 alors a n'est pas un témoin de Miller pour n.
Faisons ce test avec k valeurs aléatoires de a ; si aucune des valeurs a, tirées au hasard, n'est témoin de Miller, le nombre n est vraisemblablement premier.
Plus précisément, si n est composé, la probabilité d'être déclaré premier est $< \frac{1}{4^k}$. On peut prendre par exemple $k = 50$.

Construire un nombre premier de taille donnée

- On choisit aléatoirement un nombre impair
- On teste ce nombre
- S'il n'est pas premier, on lui ajoute 2

Construire p et q premiers tels que $p = kq + 1$

- On fixe un nombre premier q
- On choisit aléatoirement k
- On teste si $p = kq + 1$ est premier
- Si p n'est pas premier, on incrémente k de 2 et on reteste la primalité de $p = kq + 1$

L'algorithme aboutit en pratique.
Cette méthode est utilisée dans le système RSA

Les nombres de Sophie Germain

- au lieu de fixer q, on fixe k (par exemple 2)
- Si q et $2q+1$ sont premiers, q est un nombre de Sophie Germain.
- On construit un q premier et on teste si $2q+1$ est premier, sinon on passe au nombre premier q suivant.

DLP

Soit G un groupe cyclique d'ordre n (noté multiplicativement)
Soit α un générateur du groupe G. On a

$$G = \{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\}$$

Ainsi tout élément x de G s'écrit d'une unique façon sous la forme :

$$x = \alpha^k$$

avec $0 \leq k \leq n-1$. L'exposant k est appelé le logarithme discret de x.

Le problème du logarithme discret (DLP) est de trouver k connaissant x.

Pour certains groupes, le DLP est difficile (par exemple $\mathbb{Z}/p\mathbb{Z}^*$, $|p| > 1024 bits$)

Cas de $\mathbb{Z}/p\mathbb{Z}^*$

- Si on représente les éléments comme des entiers de $\{1, \cdots, p-1\}$, il existe des algorithmes sous-exponentiels pour résoudre DLP
- Pour un groupe générique, le problème est exponentiel
- L'algorithme "pas de géant, pas de bébé" est un exemple d'algo exponentiel qui permet de résoudre DLP dans tout groupe
- Conséquence : dans le premier cas, la taille du groupe doit être plus grande que dans le second
- 1024 bits contre 160 bits pour un groupe générique
- Sous-groupe d'ordre premier q de $\mathbb{Z}/p\mathbb{Z}^*$ se comporte comme un groupe générique (environ 1024 bits pour p, 160 bits pour q)
- Groupe des points d'une courbe elliptique : aucun algo sous-exponentiel, donc exposants et clés plus petits

DLP : Pas de bébé, pas de géant

$x = \alpha^k \mod p$

- On fixe $w < n \quad (n = Ord(\alpha))$
- On pose $k = k_0 + w\,k_1$, $0 \leq k_0 < w$, $0 < k_1 < n/w$
- On veut résoudre $\alpha^{k_0} = x\,\alpha^{-wk_1}$
- On calcule et stocke tous les α^{k_0} (pas de bébés)
- On calcule $x\,\alpha^{-wk_1}$ (pas de géant) que l'on compare à α^{k_0}
- Si les valeurs sont différentes, on incrémente k_1

Le coût de cet algorithme est $O(w)$ opérations pour la première phase, et $O(n/w)$ pour la seconde. En choisissant w de l'ordre de $(n)^{1/2}$, on obtient une complexité de $O((n)^{1/2})$.

Exemple d'algorithme cryptographique : DSA (FIPS PUB 186-2

Standard du NIST pour la signature digitale

Les paramètres :

- p module premier, $2^{L-1} < p < 2^L$ pour $512 \leq L \leq 1024$ et $64|L$
- q un diviseur premier de $p-1$, $2^{159} < q < 2^{160}$
- $g = h^{(p-1)/q} \mod p$, $1 < h < p-1$ tq $h^{(p-1)/q} \mod p > 1$
- $x \in_R [1, \ldots, q-1]$
- $y = g^x \mod p$
- $k \in_R [1, \ldots, q-1]$

p, q, g : public

x, y : clé privée, publique

k : généré à chaque signature, doit être gardé secret

Signature DSA

La signature de M est le couple

$$(r = (g^k \mod p) \mod q,\ \ s = (k^{-1}(SHA1(M) + xr)) \mod q)$$

Vérification Soit M', r', s' les versions reçues de M, r, s

- Vérifier que $0 < r' < q$ et $0 < s' < q$
- Calculer $w = (s')^{-1} \mod q$
- Calculer $u_1 = ((SHA1(M'))w) \mod q$
- Calculer $u_2 = ((r')w) \mod q$
- Calculer $v = (((g)^{u_1}(y)^{u_2}) \mod p) \mod q$
- Si $v = r'$ la signature est vérifiée

Preuve que $v = r'$

Lemma

Avec les mêmes notations et $g = h^{(p-1)/q} \mod p$, *on a* $g^q \mod p = 1$, *et si* $m \mod q = n \mod q$, *alors* $g^m \mod p = g^n \mod p$.

Preuve : $g^q \mod p = (h^{(p-1)/q} \mod p)^q \mod p$
$= h^{(p-1)} \mod p = 1$

Soit $m \mod q = n \mod q$, i.e., $m = n + kq$, $k \in \mathbb{Z}$

$$\begin{aligned} g^m \mod p &= g^{n+kq} \mod p \\ &= (g^n g^{kq}) \mod p \\ &= ((g^n \mod p)(g^q \mod p)^k) \mod p \\ &= g^n \mod p \end{aligned}$$

Preuve que $v = r'$ (suite)

Théorème

Si $M' = M, r' = r$, et $s' = s$ dans la vérification de la signature, alors $v = r'$.

Preuve :

$$\begin{aligned} v &= ((g^{u_1} y^{u_2}) \mod p) \mod q \\ &= ((g^{SHA1(M)w} y^{rw}) \mod p) \mod q \\ &= ((g^{SHA1(M)w} g^{xrw}) \mod p) \mod q \\ &= ((g^{(SHA1(M)+xr)w}) \mod p) \mod q \end{aligned}$$

de plus

$$s = (k^{-1}(SHA1(M) + xr)) \mod q)$$

donc

$$w = (k(SHA1(M) + xr)^{-1}) \mod q)$$

et

$$(SHA1(M) + xr)w \mod q = k \mod q$$

donc par le lemme

$$v = (gk \mod p) \mod q = r = r'$$

Génération des nombres premiers

DSA a besoin de deux nombres premiers p et q satisfaisant :

- $2^{159} < q < 2^{160}$
- $2^{L-1} < p < 2^L$ pour $L = 512 + 64j$ et $0 \leq j \leq 8$
- q divise $p - 1$

Détermination de q

- Choisir un nombre appelé SEED de 160 bits, g est le nombre de bits de SEED
- Calculer $U = SHA1[SEED]$ XOR $SHA1[(SEED + 1) \mod 2^g]$
- $q = U$ OR 2^{159} OR 1 (q doit être impair et de la bonne taille)
- Utiliser un test de primalité pour tester q
- Si q n'est pas premier, choisir un nouveau SEED

Détermination de p

1. On pose $counter = 0$ et $offset = 2$
2. Pour $k = 0, \ldots, n$ $V_k = SHA1[(SEED + offset + k) \mod 2^g]$
3. On pose

$$W = V_0 + V_1 * 2^{160} + \cdots + V_{n-1} * 2^{(n-1)*160} + (V_n \mod 2^b) * 2^{n*160}$$

   et $X = W + 2^{L-1}$
4. Soit $c = X \mod 2q$ et $p = X - (c - 1)$. On a alors $p = 1 \mod 2q$.
5. Si $p < 2^{L-1}$ aller à 7
6. Tester la primalité de p, si ok aller à 9
7. $counter = counter + 1$ et $offset = offset + n + 1$
8. Si $counter \geq 2^{12} = 4096$ aller à 1 sinon aller en 2.
9. Sauver les valeurs $SEED$ et $counter$ (pour certifier le processus de génération)

Conclusion

- Les nombres premiers jouent des rôles très importants en cryptographie à clés publiques
- Les algorithmes utilisant des grands nombres doivent être rapides (bonne complexité)
- Est-il possible de diminuer la taille des clés sans affecter la sécurité ?
- Oui, la cryptographie elliptique permet dans certains cas d'utiliser des tailles de clés plus petites