N ° d'ordre : 33 /2009 - M /MT

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIQUE
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
« HOUARI BOUMEDIEN »
FACULTE DE MATHEMATIQUES

Présenté pour l'obtention du diplôme de MAGISTER

EN : Mathématiques

Spécialité : Algèbre et Théorie Des Nombres

Par : AOUINA Mohamed

Sujet

Tests de primalité et Application en Cryptographie

Soutenu le 20 Juillet 2009, devant le jury composé de :

Mr. AIDER Méziane,	Professeur,	U.S.T.H.B	Président
M^{r}. HERNANE Mohaned Ouamar,	Maître de Conférences,	U.S.T.H.B	Directeur de Thèse
M^{r}. KHELLADI Abdelkader,	Professeur,	U.S.T.H.B	Examinateur
M^{r}. ZITOUNI Mohamed,	Professeur,	U.S.T.H.B	Examinateur
M^{r}. HACHAICHI Mohamed Salah,	Maître de Conférences,	U.S.T.H.B	Examinateur
M^{r}. CHERCHEM Ahmed,	Maître Assistant classe A,	U.S.T.H.B	Invité

Remerciements :

Ce travail n'aurait jamais vu le jour sans l'aide précieuse de Monsieur Mohand Ouamar HERNANE, mon directeur de Thèse qui m'a proposé le sujet de mon mémoire et guidé tout au long de ces deux années de thèse ; par ses orientations, encouragements et explications aussi claires que précises. J'exprime toute ma gratitude à Messieurs les Professeurs Mohamed ZITOUNI, Benali BENZAGHOU et Mohamed Saleh HACHAICHI qui m'ont initié à l'Algèbre et Théorie des Nombres grâce aux cours de la première année de Post-Graduation.

Je remercie vivement le Professeur Meziane AIDER, d'avoir accepté de présider mon jury de soutenance de cette thèse au vue de la surcharge de travail dûe à ses responsabilités de Vice Doyen Chargé de la Post-Graduation et de la Recherche au sein de la faculté de Mathématiques. Mes remerciements vont également à Messieurs les Professeurs Mohamed ZITOUNI, Abdelkader KHELADI et HACHAICHI Mohamed Salah pour avoir accepté d'examiner ce travail et pour l'honneur qu'ils me font en acceptant de participer au jury. Je remercie aussi Monsieur CHERCHEM Ahmed Maître Assistant A, à l'USTHB, d'avoir accepté de faire partie du Jury en tant que membre invité.

Table des matières :

Notations :

Dans ce mémoire, nous utiliserons les notations suivantes :

(1) Log, désigne le logarithme Népérien et Log_a le logarithme Népérien de base a.

(2) P(n), désigne le plus grand facteur premier de n.

(3) p_n le *nième* nombre premier.

(4) G.R.H : Abréviation pour désigner « Hypothèse de *Riemann généralisée* »

(5) $\pi(x)$, nombre de nombres premiers inférieurs ou égaux à x.

(6) φ, la fonction arithmétique (indicatrice) d'*Euler*.

(7) Λ, la fonction arithmétique de Von Mangoldt.

(8) ϑ et ψ, les fonctions de *Tchebychev*

(9) *GIMPS*, abréviation de « Great Internet *Mersenne* Prime Search ».

(10) $\mathbb{N}$: monoïde des entiers naturels

(11) $\mathbb{N}^*$: monoïde $\mathbb{N}$ sans le nombre zéro (0)

(12) $\mathbb{Z}$: anneau des entiers rationnels

(13) $\mathbb{Z}/n\mathbb{Z}$: anneau des entiers rationnels modulos n ; $a \equiv b \pmod{c}$ dans l'anneau $\mathbb{Z}$: a congru à b modulo c ; $a - b$ est un multiple de c.

(14) $(\mathbb{Z}/p\mathbb{Z})^*$: groupe cyclique modulo un nombre premier p

(15) Pgcd(a ,b) : plus grand commun diviseur de deux entiers naturels a et b

(16) Le symbole $a \mid b$ signifie l'entier naturel a divise l'entier naturel b .

Introduction :

Les nombres premiers jouent un rôle important en Théorie des Nombres et ses applications en cryptographie. Pour reconnaître qu'un entier est premier ou non, il existe plusieurs tests appelés Test de primalité. Pour savoir si un entier positif de 100 chiffres est premier ou non, en utilisant le test de Fermat (1650) il aurait fallu pour cela utiliser les plus grands et puissants ordinateurs de l'époque, ceci nécessiterait un siècle de calculs. Il a fallu quatre siècles pour découvrir un nouveau test de primalité ; en test de Solovay- Strassen (1976), ce test améliore celui de Fermat, il repose essentiellement sur le symbole de Jacobi. En 1978 Miller et Rabin ont démontré une version améliorée du test Solovay-Strassen.

Le premier chapitre de ce mémoire est consacré à quelques rappels sur les congruences, les fonctions arithmétiques d'Euler, Möbius et les fonctions de Chebychev. Dans le second chapitre, nous rappelons quelques notions et propriétés des nombres premiers : le théorème des nombres premiers, la distribution des nombres premiers et quelques nombres premiers particuliers tels que les nombres de Fermat et les nombres de Mersenne.

Au troisième chapitre, nous décrivons quelques tests de primalité ; nous nous sommes intéressés particulièrement aux tests de Fermat, de Solovay Strassen, de Miller Rabin, de Lucas-Lehemer, d'Adleman, de Pomerance et de Rumely.

Dans le quatrième chapitre, nous présentons le problème du logarithme discret dans les corps finis, ainsi que les algorithmes de Shanks, de Pollard, de Polhig-Hellman et d' Adleman que nous avons appliqué à quatre exemples choisis.

Le cinquième et dernier chapitre est consacré à l'application des tests de primalité et du logarithme discret en cryptographie. Nous en donnons deux applications :
La première concerne le test de Rivest-Shamir et Adelman (RSA), nous avons traité deux exemples et nous avons effectué les calculs à l'aide du logiciel Mathématica.
La deuxième application, concerne le Cryptosystème d'EL GAMAL, qui repose essentiellement sur le problème du logarithme discret.

A la fin de ce mémoire nous présentons en annexe deux applications la première concerne la génération de clés et la deuxième permet de chiffrer et déchiffrer un message par la méthode RSA.

Chapitre I
Rappels d'arithmétique

Dans ce chapitre, nous rappelons quelques notions d'arithmétique ainsi que certaines fonctions arithmétiques que nous utiliserons dans ce mémoire.

1. Division Euclidienne et Congruences :

Théorème 1 :

Soit $b \in \mathbb{N}^*$ un entier strictement positif et $a \in \mathbb{Z}$. Alors il existe un unique couple (q,r) d'entiers naturels tel que $a = bq + r, \quad 0 \le r < b.$

Preuve : *c f* [21]

Définition 1 :

Soit a et b deux entiers relatifs. Soit n un entier naturel fixé.

On dit que a et b sont congrus modulo n si et seulement si $a - b$ est un multiple de n.

On note alors :

$$a \equiv b \pmod{n}$$

Théorème 2 : (des Restes Chinois**)**

Soient m_1 , m_2 ,…, m_t des entiers positifs premiers entre eux deux à deux

Soient x_1 , x_2 ,…, $x_t \in \mathbb{Z}$, on pose $m = \prod_{j=1}^{t} m_j$.

Alors le système de t congruences $(S)\begin{cases} x \equiv x_1 \pmod{m_1} \\ \quad \ldots\ldots\ldots\ldots \\ x \equiv x_t \pmod{m_t} \end{cases}$

admet au moins une solution dans $\mathbb{Z}$, deux solutions de ce système diffèrent d'un multiple de m.

Preuve : avec le :

Lemme 1 :

Soient $a \in \mathbb{Z}$ et m_1 , m_2 ,.., $m_t \in \mathbb{N}^*$ tels que $\text{pgcd}(m_i, m_j) = 1$, pour tout i , j , $i \neq j$.

Si pour tout i, $1 \le i \le t$, m_i divise a, alors $\prod_{i=1}^{t} m_i$ divise a.

Preuve du théorème 2 :

Posons $m = \prod_{j=1}^{t} m_j$ et $s_i = \frac{m}{m_i} = \prod_{\substack{j=1 \\ i \neq j}}^{t} m_j$, $1 \le i \le t$;

l'hypothèse pgcd (m_i , m_j) =1, pour tout $i \neq j$ implique pgcd (m_i , s_i) = 1

par le théorème de Bezout, il existe un couple $(u_i , v_i) \in \mathbb{Z}^* \times \mathbb{Z}^*$ tel que $u_i.m_i + v_i.s_i = 1$

on a $v_i.s_i \equiv 1 (\text{mod } m_i)$ et $v_i.s_i \equiv 0 (\text{mod } m_j)$, $i \neq j$,

d'où $\begin{cases} x_i.v_i.s_i \equiv x_i (\text{mod } m_i) \\ x_i.v_i.s_i \equiv 0 (\text{mod } m_j) \end{cases}$ pour tout $i \neq j$

Soit $x = \sum_{i=1}^{t} x_i.v_i.s_i = x_1.v_1.s_1 + x_2 v_2 s_2 + ... + x_t.v_t.s_t$, alors $x \equiv x_i.v_i.s_i \ (\text{mod } m_i)$

et par conséquent, $x \equiv x_i (\text{mod } m_i)$, pour tout i, $1 \le i \le t$

Supposons que x' est une autre solution de ce système, alors $x = x_i (\text{mod } m_i)$ et $x' \equiv x_i (\text{mod } m_i)$

pour tout i, $1 \le i \le t$, d'où $x' \equiv x (\text{mod } m_i)$.

Comme pgcd (m_i, m_j) = 1, si $i \neq j$ alors $m = \prod_{i=1}^{t} m_i$ divise $x' - x$

donc $x' = x + k.m$, avec $k \in \mathbb{Z}$.

□

2. La fonction arithmétique σ somme de diviseurs :

Définition 2 :

On appelle fonction arithmétique somme des diviseurs d'un entier n, la fonction notée σ définie pour tout $n \ge 1$, par $\sigma(n) = \sum_{d|n} d$.

Calcul de $\sigma(n)$:

Si $n = p^{\alpha}$, où p est premier et $\alpha \ge 1$, alors $\sigma(n) = \frac{p^{\alpha+1} - 1}{p - 1}$.

Si $n = \prod_{i=1}^{k} p_i^{\alpha_i}$, où p_i sont des nombres premiers distincts et $\alpha_i \geq 1$, alors

$$\sigma(n) = \prod_{i=1}^{k} \left(\frac{p_i^{\alpha_i} - 1}{p_i - 1} \right)$$

Définition 3:

Un entier $n \geq 1$ est un nombre parfait si et seulement si $\sigma(n) = 2n.$

Exemple 1:

Les entiers n = 6, 28,496,8128 sont des nombres parfaits. 46 nombres parfaits sont connus à ce jour.

3.Le groupe (ℤ / n ℤ)* et la fonction φ d'Euler :

Définition 4:

L'ensemble des éléments inversibles de l'anneau **(ℤ / n ℤ, +, .)** muni de la multiplication est un groupe. On le note (ℤ / n ℤ)*.

Théorème 3 : Soit $n \geq 2$ et $\bar{a} \in$ (ℤ / n ℤ)*, alors

$\bar{a}$ est inversible si et seulement si a et n sont premiers entre eux.

Preuve :

Le calcul de l'inverse de $\bar{a}$ se fait à l'aide de l'algorithme de division Euclidienne.

Proposition 1:

Pour que l'anneau ℤ / p ℤ soit un corps il faut et il suffit que p soit un nombre premier.

Preuve :

On note F_p le corps ℤ / p ℤ lorsque p est premier.

Définition 5: Soit n un entier non nul et a premier avec n.

On appelle ordre de a modulo n l'ordre de l'élément $\bar{a}$ du groupe (ℤ / n ℤ)*.

C'est le plus petit entier $k > 0$, tel que $a^k \equiv 1 (\text{mod } n)$.

Définition 6: On appelle fonction φ d'Euler ou indicatrice d'Euler, l'application

$\varphi : \mathbb{N}^*$ T $\mathbb{N}^*$, définie par

$\varphi(1) = 1$, $\varphi(n)$ = card { $k \in \mathbb{N} : 1 \leq k \leq n$ et pgcd(k, n)=1}

On a, $$\varphi(n) = \sum_{\substack{k=1 \\ (k,n)=1}}^{n} 1 = \left|(\mathbb{Z}/n\mathbb{Z})^*\right|$$

La fonction φ d'Euler satisfait la :

Proposition 2 :

1. $\varphi(p^{\alpha}) = p^{\alpha} - p^{\alpha-1}$ pour p premier et $\alpha \geq 1$;
2. $\varphi(p) = p-1$ pour tout nombre premier p
3. $\varphi(m.n) = \varphi(m)\varphi(n)\ (d/\varphi(d))$, où m et $n \in \mathbb{Z}^*$ et $d = p\gcd(m,n)$;
4. $\varphi(m.n) = \varphi(m).\varphi(n)$ si m et n sont premiers entre eux ;
5. $\varphi(n)$ est pair pour $n \geq 3$,
6. si n possède r nombres impairs de facteurs premiers distincts, alors : $2^r \mid \varphi(n)$.

Preuve : *c.f* [3]

Exemple 2 : Valeurs $\varphi(n)$ pour n composé

n	100	101	9763298658742318956742 1	10034567894567890345677 3
$\varphi(n)$	40	100	9701112043081539855091 2	6684233590591889318928 0

Proposition 3 :

Soit $n \geq 1$, tel que $n = \prod_{i=1}^{k} p_i^{\alpha_i}$, où p_i premiers distincts et $\alpha_i \geq 1$, alors

$$\varphi(n) = n\prod_{i=1}^{k}\left(1 - \frac{1}{p_i}\right).$$

Preuve : *c.f* [3]

Théorème 4: *c.f* [6]

Soit $n \geq 1$, on a : $\sum_{d|n} \varphi(d) = n$.

Preuve : *c.f* [3]

Théorème 5:

Soit n un entier positif et $a \in \mathbb{Z}$ un entier premier avec n, alors,

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

Preuve :

S'obtient grâce à l'application du théorème de *Lagrange* au groupe $G = (\mathbb{Z}/n\mathbb{Z})^*$ des éléments inversibles de l'anneau $\mathbb{Z}/n\mathbb{Z}$.

Si a premier avec n alors, $\bar{a} \in (\mathbb{Z}/n\mathbb{Z})^*$, comme $|(\mathbb{Z}/n\mathbb{Z})^*| = \varphi(n)$ donc $a^{\varphi(n)} \equiv 1 (\mathrm{mod}\, n)$

□

4. Le groupe multiplicatif du corps $\mathbb{Z}/p\mathbb{Z}$:

La définition de la fonction φ d'Euler implique que si p est un nombre premier alors $\varphi(p) = p - 1$; nous obtenons alors le

Théorème 6 : (petit théorème de Fermat**)**

Si p est un nombre premier et $a \in \mathbb{Z}$ tel que pgcd (a , p) = 1 alors,

$$a^{p-1} \equiv 1 \bmod p .$$

Preuve : C'est une conséquence du théorème d'Euler.

□

Remarque : Tout élément de $\mathbb{Z}_p$ vérifie la congruence $a^p \equiv a \bmod p$.

Le petit théorème de Fermat, donne une condition nécessaire de primalité : Si on veut montrer qu'un entier n n'est pas premier, il suffit de trouver un entier a premier à n tel que $a^n \equiv 1 (\bmod\ n)$.

Théorème 7 :

Soit p un nombre premier, alors le groupe multiplicatif $(\mathbb{Z}/p\mathbb{Z})^*$ est cyclique.

Preuve : *c.f* [14]

Les logarithmes discrets :

Définition 7:

Soit p un nombre premier, g un générateur du groupe $(\mathbb{Z}/p\mathbb{Z})^*$ et $\beta \in (\mathbb{Z}/p\mathbb{Z})^*$.

Le logarithme discret de β en base g noté $\log_g \beta$ est l'unique entier α tel que

$$0 \le \alpha \le p - 2 \quad \text{et} \quad \beta = g^{\alpha}$$

Si n est un entier tel que $\beta = g^n$, le logarithme discret de β en base g est le reste de la division euclidienne de n par $p - 1$.

Théorème 8:

Soit p un nombre premier et g un générateur du groupe multiplicatif $(\mathbb{Z}/p\mathbb{Z})^*$.

Pour tous a et b dans $(\mathbb{Z}/p\mathbb{Z})^*$, on a :

$$\log_g a.b \equiv (\log_g a + \log_g b) \bmod (p-1)$$

□

Exemple 3: p =13, et g = 7. Le calcul des puissances de g , montre que g =7 est un générateur du groupe multiplicatif $(\mathbb{Z}/13\mathbb{Z})^*$, (table 1)

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

Table 1

de la tabale1, on déduit la table des valeurs des logarithmes en base g (table 2)

β	1	2	3	4	5	6	7	8	9	10	11	12
$\log_7 \beta$	0	11	8	10	3	7	1	9	4	2	5	6

Table2

5. La fonction arithmétique de *Möbius* : *c f* [3]

Définition 8 : La fonction arithmétique de Möbius notée μ **,** est définie par

$$\mu(n) = \begin{cases} 1 \quad si \quad n = 1 \\ (-1)^k \quad si \quad n = p_1.p_2...p_k,\ p_i \quad premiers\ distincts \\ 0 \quad si \quad n = mp^2, \quad p \quad premier \end{cases}$$

Exemple. 4 :

n	1	2	3	4	5	6	7	8	9	10
μ (n)	1	-1	-1	0	-1	1	-1	0	0	1

Pour calculer les valeurs de la fonction de *Möbius*, nous utilisons le logiciel Maple

6. Résidus et non résidus quadratiques :

Définition 9: Soit p un nombre premier impair et x un entier tel que $1 \leq x \leq p$ -1.

On dit que x est un résidu quadratique modulo p , si la congruence $y^2 \equiv x \pmod p$ admet une solution dans $\mathbb{Z}/p\mathbb{Z}$.

On dit aussi que x est un carré modulo p .

Exemple 5 : pour $p = 11$, les résidus quadratiques modulo 11 sont 1, 3, 4, 5 et 9

Critère d'Euler :

Théorème 9: *cf* [3]

Soit p un nombre premier impair et soit $a \in \mathbb{Z}$ tel que pgcd (a , p) = 1, alors

a est un résidu quadratique modulo p si et seulement si $a^{\frac{p-1}{2}} \equiv 1 (\text{mod } p)$.

Preuve : Soit p un nombre premier, a un entier premier avec p et soit g un générateur du groupe cyclique $(\mathbb{Z} / p \mathbb{Z})^*$.

Si $\bar{a}$ est un carré dans $\mathbb{Z} / p \mathbb{Z}$, alors $\bar{a} = g^{2t}$, on a $\bar{a}^{(p-1)/2} = \bar{a}^{2t(p-1)/2} = g^{(p-1)t} = 1$

Si $\bar{a}$ n'est pas un carré dans $\mathbb{Z} / p \mathbb{Z}$, posons $\bar{a} = g^{2t+1}$, alors $\bar{a}^{(p-1)/2} = g^{(p-1)t} g^{(p-1)/2} = -1$, puisque $g^{(p-1)/2}$ est une racine de $x^2 - 1 = 0$ différente de 1, c'est donc -1. □

7. Le symbole de Legendre :

Définition 10 : Soient a un entier et p un nombre premier impair.

Le symbole de Legendre $\left(\frac{a}{p}\right)$ est défini par,

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & si \quad p \quad divise \ a \\ +1 & si \quad a \quad est \quad un \quad carré \mod p \\ -1 & si \quad a \quad n'est \ pas \quad un \quad carré \mod p \end{cases}$$

Le symbole de Legendre satisfait la:

Proposition 4:

Soient p un nombre premier impair, a et $b \in \mathbb{Z}$ tels que pgcd(a , p) = pgcd(b , p) = 1. Alors,

(1) $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} (\text{mod } p)$ (Critère d'Euler)

(2) $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{a.b}{p}\right)$ (Multiplicativité du symbole de Legendre)

(3) $\left(\frac{a^2}{p}\right) = 1$ et $\left(\frac{a^2.b}{p}\right) = \left(\frac{b}{p}\right)$

(4) $\left(\frac{a + \lambda.p}{p}\right) = \left(\frac{a}{p}\right)$ (Périodicité du symbole de Legendre)

(5) $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}}$ (Caractère quadratique de -1)

(6) $\left(\frac{2}{p}\right)=\begin{cases}+1 & si \quad p\equiv \pm 1(\mathrm{mod}\,8)\\ -1 & si \quad p\equiv \pm 3(\mathrm{mod}\,8)\end{cases}$

Preuve : *c f* [6]

Loi de réciprocité quadratique:

Proposition 5: *cf* [3]

Soient p et q deux nombres premiers impairs distincts, alors

$$\left(\frac{p}{q}\right)=(-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}\left(\frac{q}{p}\right)$$

Preuve : (*c f* [3]**)**

Le symbole de Jacobi :

Lorsque a n'est pas premier, le calcul du symbole de Legendre $\left(\frac{a}{p}\right)$ exige la factorisation de l'entier a .

Définition 11 :

Soit m un entier et n un entier positif impair dont la décomposition en facteurs premiers est $n=p_1^{\alpha_1}.p_2^{\alpha_2}...p_k^{\alpha_k}$. Le symbole de *Jacobi* $\left(\frac{m}{n}\right)$ est défini par,

$$\left(\frac{m}{n}\right)=\left(\frac{m}{p_1}\right)^{\alpha_1}\left(\frac{m}{p_2}\right)^{\alpha_2}....\left(\frac{m}{p_k}\right)^{\alpha_k}$$

Le symbole de Jacobi satisfait la ;

Proposition 6 : Soit n un entier positif impair et $a\in\mathbb{Z}$ alors,

(1) $\left(\frac{a}{n}\right)=1$ ne caractérise pas les carrés inversible de $\mathbb{Z}/n\mathbb{Z}$

(2) le symbole de *Jacobi* ne vérifie pas le critère d'Euler : en général, $\left(\frac{a}{n}\right)\not\equiv a^{\frac{n-1}{2}}(\mathrm{mod}\,n)$

(3) $\left(\frac{a+\lambda.n}{n}\right)=\left(\frac{a}{n}\right)$ (Périodicité), $\lambda\in\mathbb{Z}$

(4) $\left(\frac{a.b}{n}\right)=\left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$ (Multiplicativité)

(5) $\left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}}$

(6) $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}} = \begin{cases} +1 & si \quad n \equiv \pm 1 \bmod 8 \\ -1 & si \quad n \equiv \pm 3 \bmod 8 \end{cases}$

Preuve : *c f* [3]

Loi de réciprocité quadratique : pour des entiers m et n premiers entre eux

Proposition 7:

Pour tous entiers m et n impairs, on a $\left(\frac{m}{n}\right) = (-1)^{\frac{m-1}{2}\cdot\frac{n-1}{2}}\left(\frac{n}{m}\right)$

Preuve : *c f* [3]

La fonction arithmétique de *Von Mangoldt*

Définition 12 : La fonction arithmétique de *Von Mangoldt* notée Λ est définie par,

$$\Lambda(n) = \begin{cases} \log p & si \quad n = p^m, \quad p \ premier \quad et \quad m \geq 1 \\ 0 & \sin on \end{cases}$$

Les fonctions de *Tchebychev* : *c f* [3]

Il y en a deux fonctions de *Tchebychev* $\psi(x)$ *et* ϑ (x).

(1) **La fonction de *Tchebychev*** $\psi(x)$

Définition 13: La fonction ψ de *Chebychev* est définie pour $x > 0$, par

$$\psi(x) = \sum_{n \leq x} \Lambda(n)$$

(2) La fonction de *Tchebychev* $\vartheta(x)$

Définition 14 : Elle est définie pour $x > 0$, par

$$\vartheta(x) = \sum_{p \leq x} \log p \text{, où } p \text{ est premier.}$$

Relation entre $\vartheta(x)$ **et** $\psi(x)$**:** Les fonctions $\vartheta(x)$ et $\psi(x)$ sont liées par la relation,

$$\psi(x) = \sum_{m=1}^{+\infty} \vartheta(x^{1/m})$$

La fonction d'évaluation du nombre de nombres premiers dans un intervalle réel $[0, x]$:

Définition 15: On note par $\pi(x)$ le nombre des nombres premiers $p \leq x$.

C'est-à-dire $\pi(x)$ = card $\{p \leq \ x : p \ \ est \ \ premier \ \ \} = \sum_{p \leq x} 1$

Exemple 6 : $\pi(10\ ) = 4$, $\pi(10^2) = 25$, $\pi(10^3) = 168$, $\pi(10^4) = 1229$, $\pi(10^5) = 9592$, $\pi(10^6) = 78498$. Avec le logiciel Maple calcule les valeurs de $\pi(x)$.

En 2008, Tomas OLIVEIRA a calculé $\pi(10^{23})$, la valeur est 1925320391606818006727 .

Chapitre II

Nombres premiers

Introduction :

Dans les anciennes civilisations, les premiers nombres utilisés sont les nombres entiers naturels : 1,2 ,3…avec l'avènement de l'arithmétique et de l'Algèbre, les nombres naturels ont eu des développements : nombres naturels, nombres algébriques.

Théorème 10 : Il existe une infinité de nombres premiers.

Preuve : *cf* [12]

Supposons qu'il existe un nombre fini de nombres premiers $p_1 = 2$, $p_2 = 3, ..., p_k$.

Posons $P = p_1 p_2 ... p_k + 1$

Si P est un nombre premier alors :

Il existe j tel que $1 \leq j \leq k$, $P = p_j$

Or $P = p_1 p_2 ... p_k + 1 \geq p_j$ pour tout $1 \leq j \leq k$

donc P est composé.

Soit p un nombre premier divisant P, il existe $j \in \{1,2...,k\}$ tel que $p = p_j$

Comme p divise $p_1 p_2 ... p_k$ alors p divise $P - p_1 p_2 ... p_k = 1$

Ce qui est absurde, donc l'ensemble des nombres premiers est infini

□

Exemple 7: Voici les premiers nombres premiers $p < 600$ sont obtenus à l'aide du logiciel de *Maple* .

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157,163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277,281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419,421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563,569, 571, 577, 587, 593, 599.

Exemple 8:

Le plus grand nombre premier connu est $2^{43\,112\,609}$-1, il comporte 12 978 189 chiffres, il a été découvert en 2008, par une équipe de chercheurs du GIMPS.

Théorème 11 : (Théorème fondamental d'arithmétique)

Tout entier strictement positif $n > 1$, s'écrit comme produit de puissance de nombres premiers d'une façon unique à l'ordre près des facteurs,

$$n = p_1^{\alpha_1}.p_2^{\alpha_2}...p_k^{\alpha_k} \text{ où } p_i \text{ premier et } \alpha_i \geq 1.$$

Preuve : *cf* [13]

□

Théorème 12 :

Pour tout $n \geq 1$, on a $\frac{n}{6\log n} \leq \pi(n) \leq \frac{6.n}{\log n}$,

Preuve : (*cf* [3])

Théorème 13: *c f* [3]

Soit $n \geq 1$, et p_n le n-ième nombre premier .

Alors $\frac{n.\log n}{6} < p_n < 12\left(n.\log n + n.\log\frac{12}{e}\right)$

Preuve : *c f* [6]

Théorème 14 : (Théorème de Wilson)

Soit n un entier positif.

Alors n est premier si et seulement si $(n\text{-}1)! + 1 \equiv 0 \bmod n$

Preuve : (*c f* [3]). C'est une conséquence du petit théorème de Fermat.

□

Le théorème des nombres premiers :

Gauss (1792) et Legendre (1798) avaient conjecturé que le nombre des nombres premiers inferieurs ou égaux à x est de l'ordre de $\frac{x}{\log x}$. Cette conjecture a été prouvée en (1896) indépendamment par *Hadamard* et *de la Vallée Poussin*, elle est connue sous le nom du théorème des nombres premiers ;

Théorème 15: $\pi(x) \sim \frac{x}{\log x}$, lorsque $x \to +\infty$.

Preuve : (*c.f* [3])

elle est due à *Hadamard* et *de la Vallée Poussin*

La seconde est due à *A.Selberg* et *P.Erdös* 1949.

□

Théorème 16 : soit la fonction $\pi(x)$ et les fonctions de Tchebychev :

(1) $\pi(x) \sim \frac{x}{\log x}, \quad x \to +\infty$

(2) $\vartheta(x) \sim x, \quad x \to +\infty$

(3) $\psi(x) \sim x, \quad x \to +\infty$

Preuve : *c.f* [3]

Nombres particuliers : nous considérons deux types ;

1.Nombres de Mersenne :

Définition 16:

Un nombre de Mersenne est un nombre de la forme $M_n = 2^n - 1$, où $n \in \mathbb{N}^*$.

Certains nombres de Mersenne sont premiers, d'autres composés.

Exemple 9 :

$M_2 = 3$, $M_3 = 7$, $M_5 = 31$, et $M_7 = 127$ sont premiers, mais $M_{11} = 2047 = 23.89$ est composé.

Théorème 17:

Soient a et n deux entiers positifs.

Si $a^n - 1$ est premier alors $a = 2$ et n est premier.

Preuve : *c f* **[**13**]**

Nombres de Fermat :

Définition 17:

Un nombre de *Fermat* est un nombre de la forme $F_n = 2^{2^n} + 1$.

Exemple 10:

Les nombres de *Fermat* premiers connus sont : $F_0 = 3$, $F_1 = 5$ $F_2 = 17$ $F_3 = 257$ et $F_4 = 65537$.

$F_5 = 4294967297$ n'est pas premier.

Chapitre III

Tests de primalité

Introduction :

Pour trouver des grands nombres premiers, nous choisissons en pratique des nombres aléatoires et nous testons leur primalité jusqu'à l'obtention d'un nombre premier. En utilisant le théorème des nombres premiers, nous montrons que la probabilité de découvrir un entier p premier compris entre 1 et N est d'environ $\frac{1}{\log N}$. C'est test probabiliste.

III.1. Le crible d'Eratosthène :

La plus simple méthode qui nous permet de tester la primalité d'un nombre est celle des divisions euclidiennes; mais dès que le nombre à tester devient grand, cette méthode ne convient pas en pratique.

Proposition 8: Le crible d'Eratosthène est de complexité $O(\sqrt{n}\ (\log n)^2)$.

Démonstration :

Il suffit de vérifier qu'il n'existe pas de diviseurs entiers de n parmi les entiers autres que 1 inférieurs à la racine carrée de n. Si n est composé, n a un diviseur d différent de 1 et n tel que $d < \sqrt{n}$.

Il faut donc faire $\sqrt{n}$ divisions dont le temps de calcul est en $O((\log n)^2)$,

ce qui donne une complexité en $O(\sqrt{n}\,(\log n)^2)$.

□

Tests de primalité classiques fondés sur les congruences :

III.2. Test de Fermat :

Le test de Fermat est basé sur le petit théorème de Fermat que nous avons déjà introduit au Chapitre 1.

Théorème 18 :

Pour que p soit premier, il est nécessaire que, pour tout entier a,

a premier avec p implique $a^{p-1} \equiv 1 (\text{mod } p)$.

Preuve : *c f* [21]

Exemple 11:

Soit $n = 234567893$, on choisit $a = 2$. A l'aide de l'algorithme de calcul des puissances, on vérifie que, $2^{n-1} \equiv 88148951 \pmod{n}$, donc l'entier n n'est pas premier.

Le logiciel Maple, permet de calculer les puissances modulo n.

Exemple 12: $n = 341$, $a = 2$.

On a, $2^{340} \equiv 1 (\text{mod } 341)$, cependant $n = 341 = 11 \times 31$ n'est pas premier.

Cet exemple montre qu'il existe des nombres qui satisfont la condition nécessaire de primalité de Fermat et qui sont composés, nous les caractérisons par la,

Définition 18:

Soient a et p deux entiers positifs. On dit que p est *pseudo - premier* en base a si et seulement si $a^{p-1} \equiv 1 (\text{mod } p)$.

Remarque :

En vertu du théorème de Fermat, tout nombre premier est *pseudo – premier* en base a tel que $(a, p) = 1$.

Définition 19 :

L'entier a est un *faux témoin* de primalité au sens de Fermat, si n est non premier et *pseudo -premier* en base a.

Exemple 13 :

$a = 2$ est un faux témoin de primalité pour l'entier $n = 341$ puisque $2^{340} \equiv 1 (\text{mod } 341)$

Nombres de *Carmichael* :

Le test de *Fermat* permet de montrer qu'un entier n est composé, mais ne donne pas de certitude sur sa primalité même s'il vérifie la congruence $a^{n-1} \equiv 1 (\text{mod } n)$ pour de nombreuses bases a tel que $\text{pgcd}(a, n) = 1$. Il existe des entiers vérifiant cette propriété et qui sont composés.

Exemple 14 : $n = 561$, soit a tel que $(a, n) = 1$, on a $a^{560} \equiv 1 (\text{mod } 561)$,

mais n est composé puisque $561 = 3 \times 11 \times 17$.

De tels nombres sont appelés nombres de Carmichael :

Définition 20 :

On appelle nombre de Carmichael un entier n composé tel que $a^{n-1} \equiv 1 (\text{mod } n)$ pour tout entier a , $1 < a < n$ et $\text{pgcd}(a,n) = 1$.

C'est un entier non premier et qui est pseudo – premier en base a pour tous les entiers premiers avec n.

Théorème 19 : Il existe une infinité de nombres de *Carmichael*.

Preuve : *c f* [13] , [21]

Les critères d'Euler et de Miller – Rabin :

L'existence d'un nombre infini de nombres de Carmichael, ne permet pas de trouver un test de primalité basé uniquement sur le théorème de Fermat, il existe d'autres critères.

Critère d'Euler :

Théorème 20 :

Pour que p soit un nombre premier, il est nécessaire que pour tout a tel que $\text{pgcd}(a, p) = 1$

Le symbole de Legendre $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} (\text{mod } p)$

Preuve : *c f* [3]

Définition 21 :

Soit n un entier impair. On dit que n est pseudo premier eulérien en base a si $\text{pgcd}\ (a,n) = 1$, et si $a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) (\text{mod}\, n)$

Tests de Lucas :

Théorème 21:

Soit un entier $n > 1$. S'il existe un entier $a > 1$, tel que

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

(2) $a^m \not\equiv 1 (\text{mod}\, n)$ pour $m = 1, 2, ..., n-2$

Alors n est premier.

Preuve : (*cf* [5]**)**

L'inconvénient de ce test réside dans la nécessité de faire $n-2$ multiplications par l'entier a et la vérification que 1, n'est pas un résidu modulo n d'une certaine puissance $m \leq n-2$ de a.

Théorème 22 :

Soit un entier $n > 1$. S'il existe un entier $a > 1$, tel que

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

(2) $a^m \not\equiv 1 (\mathrm{mod}\, n)$ pour tout diviseur m de $n-1$.

Alors n est premier.

Preuve : (*cf* [5])

Il ne peut s'appliquer qu'aux entiers possédant une certaine forme particulière : les entiers de la forme : $n = 2^n + 1$ ou $n = 3 \times 2^n + 1$.

Théorème 23 :

Soit un entier $n > 1$. Si pour tout facteur premier p de n, il existe un entier $a > 1$, tel que

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

(2) $a^{\frac{n-1}{p}} \not\equiv 1 (\mathrm{mod}\, n)$.

Alors n est premier.

Preuve : *c f* [13]

Théorème 24: Test de Pepin

Soit un entier $n \geq 2$ et $F_n = 2^{2^n} + 1$, soit un entier $k \geq 2$.

Alors les conditions les conditions suivantes sont équivalentes :

(1) F_n est premier et $\left(\frac{k}{F_n}\right) = -1$

(2) $k^{\frac{F_n - 1}{2}} \equiv -1 (\mathrm{mod}\, F_n)$

Preuve : *c f* [13]

Montrons que (1) implique (2) :

Le critère d'Euler pour le symbole de Legendre implique, $k^{(F_n-1)/2} \equiv \left(\frac{k}{F_n}\right)$

L'hypothèse $\left(\frac{k}{F_n}\right) \equiv -1 (\mathrm{mod}\, F_n)$ implique alors $k^{(F_n-1)/2} \equiv -1 (\mathrm{mod}\, F_n)$

Montrons que (2) implique (1) :

Soit a un entier tel que $1 \leq a \leq F_n$ et vérifiant la congruence $a \equiv k (\mathrm{mod}\, F_n)$..

Comme $a^{(F_n-1)/2} \equiv -1 (\mathrm{mod}\, F_n)$ implique $a^{F_n - 1} \equiv 1 (\mathrm{mod}\, F_n)$, par le Théorème 23, on déduit que F_n est premier, donc on a nécessairement $k = 3, 5, 10$, puisque $F_n \equiv 2 (\mathrm{mod}\, 3)$, $F_n \equiv 5 ((\mathrm{mod}\, 5), F_n \equiv 1 (\mathrm{mod}\, 8)$, en appliquant la loi de réciprocité de Jacobi aux symboles

$\left(\frac{k}{F_n}\right), k = 3,5,10$, on trouve, $\left(\frac{k}{F_n}\right) = \left(\frac{k}{F_n}\right) = \left(\frac{k}{F_n}\right) = -1$,

ce qui achève la preuve du théorème.

□

Algorithme du Test de *Fermat*

Entrées: Un nombre n > 1.

Sorties: Indique si n est composé, n'indique pas s'il est premier.

Choisir un entier a au hasard entre 2 et n-1.

si pgcd(a, n) ≠ 1 **alors**

STOP, n est composé.

finsi

si $a^{n-1} \not\equiv 1 (\text{mod } n)$ **alors**

STOP, n est composé.

Fin

Remarques :

(1) Le plus petit nombre de *Carmichael* connu "pseudo premier" est 561 = 3 × 11 ×17

(2) un test probabiliste est rapide pour la vérification de la primalité, si les grandes puissances $a^m \pmod{n}$ peuvent être calculées assez rapidement.

Exemple 15 : Nous utilisons le test de *Fermat* pour tester la primalité de $p = 323$

Nous calculons $2^{p-1} = 2^{322}$ modulo 323. Les résultats des calculs sont collectés dans le tableau ci-dessous :

s	n	n^s	$2^{2^s} \bmod 323$
0	323	0	2
1	161	1	4
2	80	0	16
3	40	0	256
4	20	0	290
5	10	0	120
6	5	1	188
7	2	0	137
8	1	1	35

D'où $2^{322} \equiv 4 \times 188 \times 35 \equiv 157 \pmod{323}$,
donc 323 n'est pas premier.

III.3. Test de *Solovay- Strassen* :

Le test de Solovay-Strassen utilise le critère d'Euler et les propriétés du symbole de Jacobi. C'est un test probabiliste.

Description de l'algorithme :

Soit n un entier strictement plus grand que 1, on choisit au hasard r entiers a_j tel que $a_j < n$ et $\text{pgcd}(a_j, n) = 1$ pour $j = 1,2,...,r$.

Pour chaque valeur de j, $1 \le j \le r$, on calcule les deux congruences :

$$a_j^{\frac{n-1}{2}} \pmod{n} \quad \text{et} \quad \left(\frac{a_j}{n}\right) \pmod{n} \ ;$$

Deux cas possibles

1) Il existe l, $1 \le l \le r$ tel que $a_l^{\frac{n-1}{2}} \equiv \left(\frac{a_l}{n}\right) \pmod{n}$, alors n n'est pas premier

2) Pour tout j, $1 \le j \le r$, $a_l^{\frac{n-1}{2}} \equiv \left(\frac{a_l}{n}\right) \pmod{n}$, alors n est probablement premier.

Exemple 16: Soit $n = 8911$, choisissons $a = 2$.

On a $2^{\frac{n-1}{2}} \equiv 2^{4455} \equiv 6364 (\mathrm{mod}\, n)$, et $\left(\frac{2}{n}\right) \equiv 1 (\mathrm{mod}\, n)$, puisque $n \equiv 7 (\mathrm{mod}\, 8)$, donc

$2^{\frac{n-1}{2}} \equiv \left(\frac{2}{n}\right) (\mathrm{mod}\, n)$, par conséquent le test de Solovay-Strassen implique que n n'est pas premier. On trouve la factorisation $n = 8911 = 7 \times 19 \times 67$ **.**

Définition 22:

On dit que a est un faux témoin de primalité de n au sens d'*Euler*, si n est non premier et pseudo premier eulérien en base a.

Théorème 25 : Si n est non premier, le nombre d'entiers a, $1 \leq a \leq n$-1 tels que $a^{\frac{p-1}{2}} \equiv \left(\frac{a}{n}\right) (\mathrm{mod}\, n)$ est plus petit que $\frac{1}{2}\varphi(n)$.

Preuve : *c.f* [21]

Les faux témoins de primalité de n sont les éléments de l'ensemble

G = { a, $1 \leq a \leq n$-1 , $a^{\frac{p-1}{2}} \equiv \left(\frac{a}{n}\right) (\mathrm{mod}\, n)$ }

Par la multiplicativité du symbole de *Jacobi,* G est un sous ensemble de ($\mathbb{Z}/n\mathbb{Z}$)*

Donc il suffit de démontrer que G n'est pas ($\mathbb{Z}/n\mathbb{Z}$)* tout entier

Pour prouver que card G $\leq \frac{1}{2}\varphi(n)$. On distingue deux cas :

(1) Si n possède un facteur carré d'un nombre premier p, alors

$n = p^s.q$ tel que $s \geq 2$ et pgcd(p , q) = 1

Posons $a = 1 + p^s.q$, alors a satisfait les deux congruences suivantes :

$a \equiv 1 (\mathrm{mod}\, p)$ et $a \equiv 1 (\mathrm{mod}\, q)$, et la définition du symbole de *Jacobi,* implique

$$\left(\frac{a}{n}\right) = \left(\frac{1}{p}\right)^s \left(\frac{1}{q}\right) = 1.1 = 1$$

En utilisant la formule du binôme, et l'inégalité $2.(s-1) \geq s$, on a

la congruence $$a^{\frac{n-1}{2}} \equiv 1 + \frac{n-1}{2}.p^{s-1}.q (\mathrm{mod}\, p^s)$$

La plus grande puissance de p qui divise $a^{\frac{n-1}{2}} - 1$, est donc p^{s-1},

il en résulte que $a^{\frac{n-1}{2}} - 1$ n'est pas multiple de n et $a \notin$ G

(2) Si n est sans facteur carré, il est le produit de k nombres premiers distincts

$n = q_1 q_2 ... q_k$, $k \geq 2$

Nous choisissons u qui n'est pas un carré modulo q_1, et un entier a tels que

$$a \equiv u(\mathrm{mod}\, q_1),\ \ a \equiv 1(\mathrm{mod}\, q_i),\ 2 \le i \le k$$

Par définition du symbole de *Jacobi,* $\left(\frac{a}{n}\right) = \left(\frac{a}{q_1}\right)\left(\frac{a}{q_2}\right)\left(\frac{a}{q_3}\right)..... = (-1).1.1... = -1$

La congruence, de $a \equiv 1(\mathrm{mod}\, q_2)$ implique $a^{\frac{n-1}{2}} \equiv 1(\mathrm{mod}\, q_2)$

q_2 divise n exclut $a^{\frac{n-1}{2}} \equiv -1(\mathrm{mod}\, n)$.

□

Exemple 17 : Nous utilisons le Symbole de Jacobi pour tester la primalité de $n = 253$.

Soit $a = 12$, nous calculons les symboles :

$$\left(\frac{12}{253}\right) = \left(\frac{2^2.3}{253}\right) = \left(\frac{2}{253}\right)^2\left(\frac{3}{253}\right)$$

$$= 1^2\left(\frac{3}{253}\right) \text{ tel que } 253 \equiv 1(\ \mathrm{mod}\ 4)$$

$$= \left(\frac{3}{253}\right);\quad 253 = 83.3 + 1$$

Nous calculons $a^{\frac{n-1}{2}} \equiv 12^{126} (\mathrm{mod}\ 253) \equiv 133\ (\mathrm{mod}\ 253)$

donc 253 n'est pas premier.

.4. Le test de *Miller Rabin* :

Soit n un entier tel que $n - 1 = 2^t m$ où m est un entier impair et $t \in$ □.

Nous voulons tester la primalité de l'entier n.

Description de l'algorithme :

Etape 1 : On choisit au hasard un entier a tel que $2 \le a \le n - 2$.

Etape 2 : On calcule $x_0 \equiv a^m (\mathrm{mod}\, n)$ **.**

(1) Si $x_0 \equiv \pm 1(\mathrm{mod}\, n)$, alors n est probablement premier.

(2) Si $x_0 \equiv \pm 1(\mathrm{mod}\, n)$ et $t = 1$, alors n est composé.

Sinon, on pose $j = 1$ et on passe à l'étape 3.

Etape 3 : On calcule $x_j \equiv a^{2j} (\mathrm{mod}\, n)$.

(1) Si $x_j \equiv 1(\mathrm{mod}\, n)$, alors n est composé.

(2) Si $x_j \equiv -1(\mathrm{mod}\, n)$, alors n est probablement premier

Sinon, on pose $j = j + 1$, puis on passe à l'étape 4.

Etape 4:

(1) Si $j = t-1$, passer à l'étape 5.

(2) Sinon, retourner à l'étape 3.

Etape 5 : Calculer $x_{t-1} \equiv a^{m2^{t-1}} \pmod{n}$.

(1) Si $x_{t-1} \equiv -1 \pmod{n}$, alors n est composé.

(2) Si $x_{t-1} \equiv -1 \pmod{n}$, alors n est probablement premier.

Exemple 18 :

Soit $n = 1729$, alors, $n-1 = 1728 = 2^6 \times 27$, il en résulte $t = 6$ et $m = 27$.

Choisissons l'entier $a = 2$, on a $x_0 \equiv 2^{27} \equiv 645 \pmod{n}$, on pose $j = 1$ et on calcule x_1, on trouve $x_1 \equiv 2^{27} \equiv 1065 \pmod{n}$, on pose $j = 2$, puis on calcule x_2, on trouve :

$$x_2 \equiv 2^{2\times 27} \equiv 1 \pmod{n},$$

le test de Miller-rabin permet alors de conclure que l'entier $n = 1729$ est composé.

Sa décomposition est $n = 1729 = 7 \times 13 \times 19$.

Définition 23: Soit n un entier impair, $n = 1 + 2^s q$ où q est impair.

On dit que n est pseudo premier fort en base a si $a^q \equiv 1 \pmod{n}$, ou s'il existe un entier k, $0 \le k \le s-1$, tel que $a^{q.2^k} \equiv -1 \pmod{n}$.

Définition 24 :

On dit que a est un faux témoin de primalité de l'entier n, au sens de Miller Rabin, si n est non premier, mais pseudo-premier fort en base a.

Théorème 26 :

N étant un entier positif fixé, Soit n un entier choisi avec la probabilité uniforme, $1 \le n \le N$, on applique r fois le test de Miller Rabin à l'entier n. Si tous les résultats sont positifs, la probabilité que n n'est pas premier, est inférieure à $\frac{N}{\pi(n)} \times \frac{1}{4^r} \approx \frac{\log N}{4^r}$

□

Théorème 27 : Pour que le nombre impair p, p -1 = $2^s.q$ et q impair, soit premier, il est nécessaire que, pour tout a premier avec p, on ait :

(1) Soit $a^q \equiv 1 \pmod{p}$

(2) Soit il existe k, $1 \le k \le s$ -1 tel que $(a^q)^{2^k} \equiv -1 \pmod{p}$

Démonstration : Nous obtenons q et s par p -1 = $2^s.q$, q impair

Considérons l'identité polynomiale $x^{2^s} - 1 = (x-1)(x+1)(x^2+1)...(x^{2^{s-1}}+1)$

Soit a premier avec p. Posons $x = a^q$.

Alors, par le théorème de *Fermat* dans le corps $\mathbb{Z}/p\mathbb{Z}$

$$0 = a^{p-1} - 1 = a^{q.2^s} - 1$$
$$= x^{2^s} - 1$$
$$= (x-1)(x+1)(x^2+1)...(x^{2^{s-1}}+1)$$

donc $(x-1)(x+1)(x^2+1)...(x^{2^{s-1}}+1) = 0$

L'un des éléments de ce produit est nul dans le corps $\mathbb{Z}/p\mathbb{Z}$, donc le théorème est démontré

□

Proposition 9:

L'algorithme de *Miller-Rabin* est polynomial, de complexité $O((\log(n)^3)$.

Preuve : *cf* [13]

Théorème 28 :

Sous l'hypothèse de *Riemann* généralisée, pour $n \in \mathbb{N}$ est composé, il existe un témoin de *Miller* $\leq 2.\log(n)^2$ □

Théorème 29 : de *Bach*

Sous l'hypothèse de *Riemann* généralisée, on a :

(1) pour tout premier p, il existe $x \in \mathbb{N}$ ***non résidu quadratique*** modulo p tel que $x \leq 2.\log(p)^2$;

(2) pour tous premiers p, p', $p \neq p'$, il existe $x \in \mathbb{N}$ qui est ***non-résidu quadratique*** modulo p, ***résidu quadratique*** modulo p', et qui satisfait $x \leq 2.\log(p.p')^2$.

Preuve : *cf* [5]

4. Recherche de nombres premiers particuliers:

1. Nombres de *Fermat* :

Définition 25 : On appelle nombre de Fermat F_n le nombre, $F_n = 2^{2^n} + 1$, $n \geq 0$

Proposition 10:

Soient a et c deux entiers non nuls. Si a est impair ou si c a un facteur impair, alors

l'entier a^c+1 est composé

Démonstration : supposons $a = 2j+1$ pour j entier, alors $a^c+1 = (2j+1)^c + 1$

$$= \sum_{k=0}^{c} C_c^k (2j)^k + 1 = \sum_{k=1}^{c} C_c^k (2j)^k + 2 = \sum_{k=1}^{c} C_c^k 2^k j^k + 2$$

donc 2 divise a^c+1 d'où a^c+1 est composé

(2) Supposons que $c = k.(2m+1)$, pour m et k entiers strictement positifs, alors

$$a^c+1 = a^{k(2m+1)}+1 = (a^k)^{2m+1}+1$$

$$= (a^k+1)(a^{2km} - a^{k(2m-1)} + ... + 1) \text{ et donc } a^c+1 \text{ est composé.} \square$$

Lemme 2 : Si a^c+1 est premier, alors a est pair et c est une puissance de deux.

Preuve : On applique la proposition 7 $\square$

Le lemme 2 est à l'origine par *Fermat* des nombres de la $2^{2^n}+1$ qui porte aujourd'hui son nom , ses nombres sont premiers pour n=0 ,1,2,3 et 4 et a conjecturé que F_n est premier pour tout n, cette conjecture s'est avérée fausse.

Théorème 30 : 641 divise $F_5 = 4294967297$.

$\square$

Le test *Pépin* est un test particulier utilisé pour tester la primalité des nombres de Fermat,

Critère de *Pepin* : *cf* [13]

Soit k un entier positif, le k-ième nombre de *Fermat* $F_k = 2^{2^k} + 1$ est premier si et seulement si $3^{\frac{F_k-1}{2}} \equiv -1 (\mathrm{mod}\, F_k)$

$\square$

2.Nombres de Mersenne :

Un nombre de Mersenne est un nombre de la forme $M_p = 2^p - 1$ où p est un nombre premier,

Théorème 31 : Soit p un nombre premier alors :

1. Si $p \equiv 3 \bmod 4$, alors $2p + 1$ est premier, si et seulement si $2p + 1$ divise M_p.
2. Si $2p + 1$ divise M_p, alors $2p + 1$ est premier.

Démonstration :

Preuve de (1)

Supposons que 2p+1 est premier, alors $2p+1 \equiv -1 (\mathrm{mod}8)$, et

en utilisant le critère d'Euler, on obtient, $2^p \equiv 1 \pmod{2p+1}$

Preuve de (2) :

$2^p \equiv 1 \pmod{2p+1}$ et que $2p+1 \equiv \prod_{i=1}^{k} p_i^{\alpha_i}$ où p_i est premier.

On a $2^{\varphi(2p+1)} \equiv 1 \pmod{2p+1}$

Donc p divise $\varphi(2p+1)$ or $\varphi(2p+1) = \prod_{i=1}^{k} p_i^{\alpha_{i-1}}(p_i - 1)$

Ce qui implique qu'il existe un entier i tel que p divise $(p_i - 1)$

or , il existe un entier n tel que : $2p + 1 = n.p_i$, p_i et n sont impairs

Donc si $n = 1$, la démonstration est finie

Sinon $n \rangle 2$ et $p \rangle p_i - 1$, et on a une contradiction.

□

Théorème 32 : Soit M_n des nombres de Mersenne

1. si $M_n = 2^n - 1$ est premier alors n est premier
2. si $M_p = 2^p - 1$ est nombre premier de Mersenne, alors $n = 2^{p-1}(2^p - 1)$ est parfait.
3. réciproquement, si $n \geq 2$ est un nombre pair et parfait alors $n = 2^{p-1}(2^p - 1)$ et $M_p = 2^p - 1$ est un nombre premier de Mersenne.

Preuve : *cf* [5]

Proposition 11: Si $p = 2^n - 1$ est premier alors n est premier.

Preuve : Supposons que n n'est pas premier alors : il existe au moins $u \geq v \geq 2$ tel que $n = u.v$ d'où $p = (2^u)^v - 1$. La proposition précédente, implique alors $u = 1$.

□

Exemple 19 :

Factorisation du nombre 2^{1092}-1, qui contient 330 chiffres décimaux. Avec le logiciel *Maple*, les calculs sont effectués dans un temps 635,81 secondes avec un espace mémoire de 52,86 Mo :

$3^2 . 5 . 7^2 . 13^2 . 29 . 43 . 53 . 79 . 113 . 127 . 157 . 313 . 337 . 547 . 911 . 10932.$
1249.1429 .1613 . 2731.3121.4733.5419 . 8191 .14449 . 21841121369 . 224771. 503413 .
1210483.1948129.22366891.108749551.112901153.23140471537.25829691707.10531075
0819.467811806281.4093204977277417.8861085190774909.556338525912325157.27570
071795154656694685449 7.869775958019498449 93.292653113147157205779127526827.
319475398781398849939742864389565956 9

III.5. Test de primalité de *Lucas* : *c f* [12]

Lucas a inventé et proposé ce test de primalité pour démontrer que le nombre de *Mersenne* M_{127} est premier. Ce nombre est le plus grand nombre identifié avant l'utilisation des ordinateurs.

Théorème 33 :

Soit a et n deux entiers positifs tels que

$$a^{n-1} \equiv 1 \pmod{n}$$

et, pour tout facteur premier p de n-1

$$a^{\frac{n-1}{p}} \not\equiv 1 \pmod{n}.$$

Alors n est premier et a est un générateur du groupe multiplicatif $(\mathbb{Z}/n\mathbb{Z})^*$.

Preuve : Soit $\bar{a}$ la classe de a modulo n

L'hypothèse $(\bar{a})^{n-1}$ =1, montre que $\bar{a}$ est un élément de $(\mathbb{Z}/n\mathbb{Z})^*$ tel que $\bar{a}$ est l'inverse de $(\bar{a})^{n-2}$. Il suffit de démontrer que l'ordre de $\bar{a}$ est $n-1$

Si ord($\bar{a}$) = $n-1$ est vrai , tous les $(n-1)$ éléments , $\bar{a}$, $(\bar{a})^2, \ldots, (\bar{a})^{n-1}$ sont non nuls, deux à deux distincts, et inversibles. Ce sont tous des éléments non nuls de $\mathbb{Z}/n\mathbb{Z}$,

Alors $\mathbb{Z}/n\mathbb{Z}$ est un corps et n est premier, $\bar{a}$ est un générateur du groupe multiplicatif $(\mathbb{Z}/n\mathbb{Z})^*$.

Soit ω l'ordre de $\bar{a}$ dans $(\mathbb{Z}/n\mathbb{Z})^*$. Montrons que $\omega = n-1$,

D'après l'hypothèse, on a $(\bar{a})^{n-1}$ =1, ω est diviseur de $n-1$.

Si ω est un diviseur propre de $n-1$, alors il existe un autre diviseur q de $n-1$,

tel que $\omega \mid \frac{n-1}{q}$

Ce qui implique $(\bar{a})^{\frac{n-1}{q}}$ =1 , contradiction avec l'hypothèse, cela implique raisonnement par l'absurde

□

Exemple 20:

Soit n = 947 et a = 2. On a $946 = 2 \cdot 11 \cdot 43$ et $2^{946} \equiv 1 \pmod{n}$.

Or $2^{946/2} = 2^{473} \equiv 946 \pmod{947}$,

$2^{946/11} = 2^{86} \equiv 215 \pmod{947}$,

$2^{946/43} = 2^{22} \equiv 41 \pmod{947}$,

Alors 947 est premier.

III 4. Test de primalité de *Lucas-Lehmer*

Théorème 34 : (Test de *Lucas-Lehmer*) *c f* [5]

Soit p un nombre premier impair. Soit la suite d'entiers $(S_k)_k$ définie par la relation de récurrence :

$$S_1 = 4 \text{ et } \forall k \geq 1 \quad S_{k+1} = S_k^2 - 2$$

Alors, le nombre de *Mersenne* $M_p = 2^p - 1$ est premier si et seulement si

$S_{p-1} \equiv 0 (\text{mod}\, M_p)$

Preuve : *cf* [5]

Considérons le polynôme $f(X) = X^2 - 2^{(p+1)/2} X - 1$, soit q un nombre premier divisant n et $\alpha, \beta \in F_{q^2}$ les racines de $f(X) = 0$ sur $\in F_q$, on a $\alpha + \beta = 2^{\frac{p+1}{2}}$ *et* $\alpha\beta = -1$

Posons $V(k) = \alpha^k + \beta^k$ et soit $\overline{S_k}$ la classe de $(\text{mod}\, q)$.

On a

$$\overline{S_k} = V(2^k) \qquad (1)$$

Un raisonnement par récurrence permet de montrer que la relation (1) est vérifiée pour tout entier $k \geq 1$,

$\overline{S_1} = V(2) = \alpha^2 + \beta^2 = (\alpha + \beta)^2 - 2\alpha\beta = (2^{(p-1)/2})^2 + 2 = 2^{p+1} + 2 = 4$,

Supposons $\overline{S_k} = V(2^k) = \alpha^{2^k} + \beta^{2^k}$, alors,

$\overline{S_{k+1}} = (\alpha^{2^k} + \beta^{2^k})^2 - 2 = \alpha^{2^{k+1}} + \beta^{2^{k+1}} + 2\alpha^{2^k}.\beta^{2^k} - 2 = \alpha^{2^{k+1}} + \beta^{2^{k+1}} = V(2^{k+1})$.

Supposons que $n = 2^p - 1$ est premier et montrons que $S_{p-1} \equiv 0 (\text{mod}\, n)$.

Le polynôme f est irréductible sur F_n car son discriminant n'est pas un carré, en effet on a

$$\Delta = (2^{(p+1)/2})^2 - 4(-1) \equiv 6 (\text{mod}\, n).$$

$\left(\frac{2}{n}\right) = +1$, puisque $(2^{(p+1)/2})^2 \equiv 2 (\text{mod}\, n)$ implique $\left(\frac{\Delta}{n}\right) = \left(\frac{2}{n}\right)\left(\frac{3}{n}\right) = \left(\frac{3}{n}\right)$

En utilisant la loi de réciprocité quadratique on trouve $\left(\frac{3}{n}\right) = -\left(\frac{n}{3}\right) = -\left(\frac{1}{3}\right) = -1$, d'où

$\left(\frac{\Delta}{n}\right) = -1$ ce qui prouve que le polynôme $f(x)$ est irréductible.

Comme, $\alpha^{n+1} = \beta^{n+1} = \alpha\beta = -1$, alors $\alpha^{n+1} + \beta^{n+1} = \overline{S_p} = -2$, $S_p = S_{p-1}^2 - 2$, implique

$S_{p-1} \equiv 0 (\text{mod}\, n)$.

Pour démontrer la réciproque, on raisonne par l'absurde, on supposera qu'on a $S_{p-1} \equiv 0 (\mathrm{mod}\, n)$ et n n'est pas premier.

Le test de Lucas-Lehmer est très pratique pour tester les nombres de Mersenne.

Description de l'algorithme :

Soit $M_n = 2^n - 1$, où $n \geq 3$, le nombre de *Mersenne* à tester.

Etape 1 : on pose $S_1 = 4$.

On calcule $S_j \equiv S_{j-1}^2 (\mathrm{mod}\, M_n)$ pour $j = 2,3,...,n-1.$

Etape 2 :

Si $S_{n-1} \equiv 0 (\mathrm{mod}\, M_n),$ alors M_n est premier.

Sinon M_n est composé.

Exemple 24 :

Pour $n = 13$, on obtient $M_{13} = 8191$.

Les calculs modulo $M_{13} = 8191$ de S_j donnent : $S_1 = 4$, $S_2 = 14$, $S_3 = 194$, $S_4 = 4870$, $S_5 = 3953$, $S_6 \equiv 5970$, $S_7 \equiv 1857$, $S_8 \equiv 36$, $S_9 \equiv 1294$, $S_{10} \equiv 3470$, $S_{11} \equiv 128$, $S_{12} \equiv 0$, donc $M_{13} = 8191$ est premier.

Critère de *Pocklington* : *c f* [13]

Théorème 35 :

Soit un entier positif $n = ab + 1$ où $a, b \in \mathbb{N}$, $b > 1$. On suppose que pour tout diviseur premier q de b il existe un entier m tel que

$$m^{n-1} \equiv 1 (\mathrm{mod}\, n) \text{ et } (m^{(n-1)/q} - 1, n) = 1,$$

alors pour tout diviseur premier p de n, on a $p \equiv 1 (\mathrm{mod}\, b)$.

De plus si $b > \sqrt{n} - 1$ alors n est premier.

Preuve: (*cf* [13])

III.5.Test de primalité *Alderman, Pomerance et Rumely*:

Il est dû à *Adleman*, *Pomerance* et *Rumely*. il permet de tester la primalité d'un entier positif quelconque N, même si on ne connaît pas les facteurs premiers de N -1 ou N +1. La complexité est polynomiale en $O\left((\log N)^{c \log\log\log N} \right)$; ce test utilise des outils de la théorie algébrique des nombres : la loi de réciprocité générale du symbole de restes de puissances et les racines de l'unité. Il existe deux versions de ce test, une déterministe et l'autre probabiliste.

Ce test utilise la théorie des corps cyclotomiques $\mathbb{Q}(\zeta_n)$,des caractères multiplicatifs de groupes abéliens finis et des sommes de Gauss

Soient p, q deux nombres premiers tels que p^k divise $q-1$ et pgcd($p.q$, N) =1

et χ un caractère multiplicatif d'ordre p^k et de conducteur q et g un générateur de $\mathbb{Z}_q^*$

alors $\chi(g) = \zeta_{p^k}$ où ζ_{p^k} = racine primitive p^k-ième de l'unité, selon la théorie des nombres , une racine primitive n-ième de l'unité est un nombre complexe $\zeta_n = \exp(\frac{2\pi i}{n})$

ce nombre engendre le $n^{\text{ième}}$ corps cyclotomique $\mathbb{Q}(\zeta_n)$

si n et m sont deux entiers premiers entre eux , alors les corps associés satisfont :

$\mathbb{Q}(\zeta_n) \cap \mathbb{Q}(\zeta_n) = \mathbb{Q}$ et $\mathbb{Q}(\zeta_n, \zeta_m) = \mathbb{Q}(\zeta_{n.m})$ (*c.f* [20]).

Le nombre $\zeta_n = \exp(\frac{2\pi i}{n})$ est racine de son polynôme minimal qui est un polynôme unitaire $f_n(x)$ et irréductible. Ce polynôme est de degré $\varphi(x)$. il peut être calculé avec la

Proposition 12:

Soit $f_n(x)$ le $n^{\text{-ième}}$ polynôme et $\mu(d)$ la fonction de Möbius

Alors, (1) $x^n - 1 = \prod_{d|n} f_d(x)$ et $1 \le d < n$

(2) $f_n(x) = \prod_{d|n} (x^{\frac{n}{d}} - 1)^{\mu(d)}$ tel que r premiers à n .

Preuve :

c f [Ed WELSS – Algebraic Number Theory Mac Graw Hill Company , New York 1970]

□

Exemple 21:

Calcul de 65^{ieme} polynôme racine primitive de l'unité est égale à $\zeta_{65} = \exp(\frac{2\pi i}{65})$ le corps cyclotomique $\mathbb{Q}(\xi)$ est une extension abélienne du corps $\mathbb{Q}$ de degré $\varphi(65) = 48$

Le 65^{ieme} corps cyclotomique est de degré 48

Le 65^{ieme} polynôme cyclotomique est de degré 48 :

$f_{65}(x) = x^{48} + a_1 x^{47} + ... + a_{48} \in \mathbb{Z}[x]$

$= \frac{x^{65} - 1}{f_1(x) f_{13}(x) f_5(x)}$

Avec $f_1(x) = x - 1$ et $f_{13}(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$

et $f_5(x) = x^4 + x^3 + x^2 + x + 1$.

Soit χ un caractère multiplicatif d'ordre p^k et de conducteur q, $\chi : \mathbb{Z}_q^* \to \mathbb{C}$ définie par sa valeur en g, g étant un générateur de $\mathbb{Z}_q^*$ par $\chi(g) = \zeta_{p^k}$

où ζ_{p^k} désigne une racine primitive p^k-ième de l'unité .

La somme de *Gauss* associée à χ est $\tau(\chi) = \sum_{x=1}^{q-1} \chi(x)\zeta_q^x$.

Proposition 13:

Soit les sommes de Gauss $\tau(\chi)$, si N est premier, et si pgcd(N, p.q) = 1, alors

$\frac{\tau(\chi)^N}{\tau(\chi^N)} = \chi(N)^{-N}$ (1) est un élément de $\mathbb{Z}/N\mathbb{Z}[\zeta_{p^k}, \zeta_q]$.

□

l'algorithme consiste à tester les identités (1)

Pour tous les couples (p,q) tels que p^k divise $q-1$ choisis de telle façon que

s = $\prod_{q \in Q} q > \sqrt{N}$. Si elles sont toutes vérifiées, alors tout diviseur r de N appartient au

groupe multiplicatif $< N \bmod s >$, d'ordre t tel que $t = \text{ppcm}_{q \in Q}(q-1)$

On vérifie qu'aucun élément de cet ensemble n'est un diviseur premier non trivial de N. Le coût de l'algorithme est déterminé par cette dernière phase, avec un coût t, qui domine le reste de l'algorithme est constitué de nombreuses opérations sur des polynômes de degré le plus petit possible, avec le plus de diviseurs possibles, et tel que s(t) = $\prod_{q-1 \mid t} q$.

Soit plus grand que $\sqrt{N}$.

Théorème 36 : Il existe $c_1, c_2 > 0$ tels qu'il existe t convenable avec

$$(\log N)^{c_1 \log\log\log N} \leq t \leq (\log N)^{c_2 \log\log\log N}$$

Preuve : *c f* [13]

□

H.W. Lenstra a amélioré cet algorithme *c.f* [19]. Il donne une version beaucoup plus pratique de l'algorithme en remplaçant les sommes de *Gauss* par celle des sommes de *Jacobi*.

Cet algorithme implémenté par *H. Cohen* et *A.K Lenstra* sont les premiers à avoir prouvé efficacement la primalité des nombres de taille variant entre 100 à 200 chiffres décimaux.

Chapitre IV

Logarithme discret

IV. Logarithme Discret : *cf* [19]

Définition 28 :

Soit G un groupe cyclique d'ordre n et g un générateur de G, alors pour tout élément β de G, le logarithme discret de β en base g noté $\log_g \beta$ est l'unique entier x tel que $0 \leq x \leq n-1$ satisfait $\beta = g^x$

Exemple 22 :

Soit p = 97 alors (ℤ/97ℤ)* est un groupe cyclique d'ordre n = 96, (ℤ/97ℤ)* = {1,2,…,96}

(ℤ/97ℤ)* possède $\varphi(97) = 96$ générateurs dont g = 5

donc : $5^{32} \equiv 35 \pmod{97}$

Définition 29 : (Le problème de logarithme discret DLP) :

Soit p un nombre premier, g un générateur du groupe $(\mathbb{Z}/p\mathbb{Z})^*$ et β un élément de $(\mathbb{Z}/p\mathbb{Z})^*$,alors , il existe un entier positif x tel que: $0 \leq x \leq p-1$ vérifiant la congruence:

$g^x \equiv \beta \pmod{p}$

Définition 30: (Le problème généralisé de logarithme discret):

Soit un groupe G cyclique fini d'ordre n, g un générateur de G et β un élément de G.

On appelle logarithme discret généralisé β en base g, l'entier x tel que: $0 \leq x \leq n-1$ vérifiant la congruence: $g^x = \beta$.

IV.1. Algorithme des pas de bébé et des pas de géant de *SHANKS*

Soit $m = \lceil t \rceil$, où $t = \sqrt{p-1}$, le plus petit entier supérieur ou égal à t.

Ecrivons le nombre x cherché sous la forme $x = qm + r$, avec $0 \le r < m$.

Comme $x < p-1$ et $m \ge \sqrt{p-1}$, alors $q \le \sqrt{p-1}$.

Déterminer x tel que $g^x = \beta$, revient donc à chercher q, r satisfaisant

$$0 \le r < m, \quad 0 < q \le \sqrt{p-1}, \quad g^{qm+r} \equiv \beta \pmod{p},$$

Ce qui implique $0 \le r < m, \quad 0 < q \le \sqrt{p-1}, \quad (g^m)^q \equiv \beta g^{-r} \pmod{p}.$

Nous déterminons d'abord l'ensemble $B = \{ \beta . g^{-r} \bmod p \, ; 0 \le r < m \}$, puis nous calculons $u = g^m \pmod{p}$, et pour $k = 0,1,2,...,m-1$

On calcule u^k et on vérifie à chaque fois si la valeur obtenue appartient ou non à l'ensemble B Si oui on détermine la valeur de l'entier r telle que $u^k \equiv \beta g^{-r} \bmod p$, on s'arrête lorsqu'on obtient q vérifiant la congruence $u^q \equiv g^{qm} \equiv \beta g^{-r} \pmod{p}$.

Ces données constituent la structure de donnée appelée dictionnaire

Algorithme *de Shanks:* (pas de bébé et pas de géant)

```
Fonction logdiscret(p,g,a:entier) ;
Var D: dictionnaire d'entiers à informations entières,
m,u,q,v,r :entier ;
début
   D:= dictionnaire vide ;
   m:= [√(p-1)];
   q:= 0 ;
   pour r de 0 à m-1 faire insere(D,ag^-rmodp,r) ;
   u:=g^mmodp;v:=1;
   tant que (r:=cherche(v,D))=ECHEC faire
         q:=q+1; v:= v.u(modp)
   fin
   retourner q.m+r;
fin
```

Exemple 23:

Nous prenons l'exemple du groupe $(\mathbb{Z}/113\mathbb{Z})^*$ donc : il y a $\varphi(113) = 112$ générateurs dont 3 engendre le groupe $(\mathbb{Z}/113\mathbb{Z})^*$. Considérons $\beta = 57 \in (\mathbb{Z}/113\mathbb{Z})^*$

Maintenant, nous cherchons à résoudre la congruence: $3^x \equiv 57 \pmod{113}$

(1) On prend $m = [\sqrt{112}] \approx 11$

(2) Nous calculons toutes valeurs de couple (j, $3^x \pmod{113}$) pour $0 \le j < 11$:

j	0	1	2	3	4	5	6	7	8	9	10
3^x (mod 113)	1	3	9	27	81	17	51	40	7	21	63

Nous ordonnons le tableau à l'aide des valeurs : 3^x (mod113)

j	0	1	8	2	5	9	3	7	6	10	4
3^x (mod 113)	1	3	7	9	17	21	27	40	51	63	81

(3) $g^{-1} \equiv 3^{-1} \pmod{113} = 38$

Nous avons $3^{p-1} \equiv 1 \pmod{113}$ implique $3^p \equiv 3 \pmod{113}$

Donc $3^{-1} \equiv 38 \pmod{113}$

Nous élevons la congruence à la puissance m :

$(3^{-1})^m = 38^m \equiv 58 \pmod{113}$

(4) nous calculons $\beta\, g^{-im} = 57.3^{-im} \equiv 57.58^i \bmod 113$ tel que i= 0,1,2,…., < 11-1 = 10

la congruence est calculée dans le tableau ci-dessous tout dépend la valeur de i :

i	0	1	2	3	4	5	6	7	8	9
$57.58^i \bmod 113$	57	29	100	37	112	55	26	39	2	3

Finalement, nous obtenons que $\beta\, g^{-9m} = 57.3^{-9m} \equiv 3 \bmod 113$ mais $m = 11$

et par conséquent $3^{100} \equiv 57 \bmod 113$ donc $\log_3 57 = 100$

Le logiciel *Maple* permet de calculer le logarithme discret

Exemple 24:

Nous prenons l'exemple le groupe (□/131□)*, donc : il y a $\varphi(131)= 130$ générateurs dont 2 est un générateur. Nous cherchons à résoudre la congruence $2^x \equiv 15 \bmod 131$

Nous choisissons t =12 et nous calculons les valeurs suivantes :

x	0	1	2	3	4	5	6	7	8	9	10	11
g^x	1	2	4	8	16	32	64	128	125	119	**107**	83
$a.g^{-xt}$	15	94	100	59	99	44	5	75	77	**107**	33	102

Nous en déduisons: $2^{10} \equiv 15 \times 2^{-12\times 9}$

Soit $2^{118} \equiv 15(\text{mod } p)$

IV.2 .Algorithme $p-1$ de Pollard pour la factorisation des grands nombres

La méthode $p-1$ de Pollard permet de factoriser des entiers impairs N très grands à la seule condition que N possède au moins un facteur premier p tel que les facteurs premiers de $p-1$ soient très petits.

Description de l'algorithme :

Soit p un facteur premier de N. On suppose que pour tout facteur premier q de $p-1$ on ait $q \le B$. On a alors $(p-1)$ divise $B!$

On prend un entier g tel que $2 \le g \le N-1$.

Description de l'algorithme :

N, B, g sont donnés

Etape 1 :

On pose : $g = a$

Etape 2 :

Pour $j = 2,3..., B$, on calcule $a = a^j \pmod N$

Etape 3 :

On calcule $d = p\gcd(a-1, N)$

Etape 3 : test

Si $1 < d < N$, alors d est un facteur de N

Sinon aucun facteur n'est trouvé.

Explication :

On a $(p-1)$ divise $B!$

A la fin de l'étape 2 de l'algorithme, on a $a \equiv g^{B!} \pmod{N}$ et puisque p divise N

donc $a \equiv g^{B!} \pmod{p}$.

Comme $g^{p-1} \equiv 1 \pmod{p}$ et que $(p-1)$ divise $B!$, alors $a \equiv 1 \pmod{p}$.

Donc p divise $(a-1)$ et p divise N **(Etape 3)**, par conséquent p divise $d = p\gcd(a-1, N)$.

On a trouvé un diviseur de N, on refait le même procédé pour factoriser d et n/d

Exemple 25:

$N = 15770708441$. On prend $B = 180$. $p = 135979$ Après calculs on trouve $a = 11620221425$, $d = 135979$.

La factorisation de N est $15770708441 = 135979 \times 115979$

Remarque :

La méthode a réussi car $135979 = 2 \times 3 \times 131 \times 173$ n'a que des petits facteurs premiers.

Le choix de $B = 180$ est justifié puisque B doit être plus grand que 173.

On vérifie qu'on a bien $p - 1 = 135978$ divise $B!$ pour $B \geq 173$.

Définition 31 :

Soit B un entier positif. Un entier n est dit B- lisse (en anglais B-smooth) si tous ses diviseurs premiers sont inférieurs ou égaux à B.

Cette définition induit celle des classes d'entiers B-lisses.

Fonctionnement de l'algorithme :

Soit B une borne de lissage. Soit Q le plus petit multiple commun de toutes les puissances des premiers plus petites ou égales à B qui sont plus petits ou égaux à n

Si $q^m \leq n$. Alors m In $q \leq$ In n et donc $m \leq \left[\frac{Inn}{Inq}\right]$ nous obtenons alors

$Q = \prod_{q \leq B} q^{\left[\frac{Inn}{Inq}\right]}$ où le produit est sur tous les premiers q distincts inférieurs ou égaux à B.

Si p est un facteur premier de n tel que $p-1$ est B-lisse

Alors $p-1$ divise Q, et par conséquence pour tout a tel que pgcd(a , p) = 1

Le petit théorème de *Fermat* implique que $a^{Q} = ((a^{p-1})^k \equiv 1 \pmod{p})$

Si pgcd(a^{Q}-1, n) = d implique p divise d donc $d = n$ l'algorithme échoué dans ce cas

Si a^{Q}-1 = z . d donc, si q divise z alors $d = n$ et l'algorithme a encore échoué

Algorithme : Algorithme Pollard $p-1$

Entrées:un entier composé n qui n'est pas une puissance de premier.

Sorties: un facteur non trivial d de n.

1. choisir une borne de lissage B.
2. choisir un entier aléatoire a tel que $2 \leq a \leq n-1$,et calculer d = pgcd(a,n).si $d \geq 2$ alors renvoyer(d).
3. Pour chaque premier $q \leq B$, faire:

- Calculer $l = \left[\frac{Inn}{Inq}\right]$
- Calculer a $\leftarrow a^{q^l} \pmod n$(utiliser l'alogorithme d'élévation à la puissance)

4. calculer d = pgcd(a-1,n).
5. si d=1 ou d = n, recommencer avec un autre a.
   sinon retourner (d).

Exemple 26 :

Algorithme de pollard p-1 pour trouver un facteur non trivial de n = 19048567

(1) Choisir la borne de lissage B = 19

(2) Choisir l'entier a=3 et calculer le pgcd(3,n) = 1

(3) Le tableau suivant liste les valeurs intermédiaire ses variables q, l et a après chaque itération de l'étape. 3 dans l'algorithme

q	l	a
2	24	2293244
3	15	1355889
5	10	16937223
7	8	15214586
11	6	9685355
13	6	13271154
17	5	11406961
19	5	554506

(4) Calculer d = pgcd(554506-1, n) = 5821

(5) deux facteurs non triviaux de n sont p = 5281 et q = $\frac{n}{p}$ = 3607.

On remarque que p -1 = 5280 = 2^5.3.5.11 et q-1 = 3606 = 2.3.601

implique p -1 est 19-lisse, alors que q-1 ne l'est pas .

Soit n un entier ayant un facteur premier p tel que $p-1$ est B-lisse le temps d'exécution de l'algorithme de Pollard $p-1$, pour trouver le facteur p est en $O(\frac{B \ln n}{\ln B})$ modulo des multiplications.

Ⅳ.3. Algorithme (Rho) ρ de Pollard :

Cet algorithme de calculs de l'algorithme discret est un algorithme aléatoire avec une même complexité que celui de *Shanks*.

Nous supposons dans cet algorithme que G un groupe cyclique d'ordre n premier puis

Nous partitionnons le groupe G en trois sous-ensembles disjoints G_1 , G_2 et G_3

tels que $G_1 \cup G_2 \cup G_3 = G$

Nous définissons une séquence d'éléments d'un groupe $x_0, x_1, x_2, \ldots$ pour $x_0 = 1$

$$x_{i+1} = \begin{cases} \beta x_i & x_i \in G_1 \\ x_i^{\,2} & x_i \in G_2 \\ g x_i & x_i \in G_3 \end{cases}$$

Pour $i \geq 0$ cette séquence d'éléments du groupe définit deux séquences d'entiers

$a_0 , a_1 , a_2 \ldots$ et $b_0 , b_1 , b_2 \ldots$ satisfont $x_i = g^{a_i} \beta^{b_i}$ pour $i \geq 0$

$a_0 = 0,\ b_0 = 0$ pour $i \geq 0$

$$a_{i+1} = \begin{cases} a_i & x_i \in G_1 \\ 2a_i \bmod n & x_i \in G_2 \\ a_i + 1 & x_i \in G_3 \end{cases} \quad \text{et} \quad b_{i+1} = \begin{cases} b_i + 1 & x_i \in G_1 \\ 2b_i \bmod n & x_i \in G_2 \\ b_i & x_i \in G_3 \end{cases}$$

L'algorithme de *Floyd* cycle finding est utilisé pour trouver deux éléments du groupe x_1 et x_2

tels que $x_i = x_{2i}$ où $g^{a_i} \beta^{b_i} = g^{a_{2i}} \beta^{b_{2i}}$ et donc: $\beta^{b_i - b_{2i}} = g^{a_{2i} - a_i}$

Puis nous prenons les logarithmes de la base g

donc $(b_i - b_{2i}) \log_g \beta \equiv a_{2i} - a_i \pmod n$

Nous cherchons à résoudre cette congruence, l'inconnu étant $\log_g \beta$

s'il y a échec, nous changeons les entiers aléatoire a_0 ,b_0 dans l'intervalle [1, n-1] et

Nous commençons par $x_0 = g^{a_0} \beta^{b_0}$ avec des petits paramètres.

Algorithme: Algorithme Rho de Pollard

Entrées : Un nombre n.
Sorties : Un facteur de n.
Choisir une valeur initiale x_0 entre 1 et n - 1, typiquement la valeur 2.
Choisir une valeur c,définir une fonction polynomiale $f(x)=(x^2+c) \bmod n$, typiquement,c = 1.
Tant que on a pas trouvé de facteur **faire**
Calculer $x_i = f(x_{i-1})$.
Calculer $x_{2i} = f(f(x_0))$.
pgcd(|xi - x2i|, n) est un facteur de n.
si le facteur est plus grand que 1 et plus petit que n **alors**
on a trouvé un facteur !
fin si
si le facteur est 0 **alors**
L'algorithme a échoué. On peut essayer de recommencer avec une nouvelle valeur de c. Si on retrouve plusieurs fois 0, n est probablement premier.
Sinon

```
 Augmenter i de 1.
Fin si
fin tant que
  Retourner le facteur.
                          -----
            Algorithme: Algorithme  Rho de Pollard
```

Exemple 27 : nous résolvons cet exemple par l'algorithme ρ de *Pollard*

Soit G un sous groupe (ℤ/383ℤ)*, g =2 un générateur de G d'ordre n = 191

Supposons que: β = 228

La partition des éléments de G en trois sous ensembles disjoints G_1, G_2, G_3

tels que $G_1 \cup G_2 \cup G_3 = G$ et de même cardinalité i.e : $|G_1| = |G_2| = |G_3|$

est définie par :

$$\begin{cases} x \equiv 1 \bmod 3 & x \in G_1 \\ x \equiv 0 \bmod 3 & x \in G_2 \\ x \equiv 2 \bmod 3 & x \in G_3 \end{cases}$$

le tableau au dessous donne toutes les valeurs $x_i, a_i, b_i, x_{2i}, a_{2i}, b_{2i}$ à la fin de chaque itération de l'étape de l'algorithme .

Nous constatons que $x_{14} = x_{28} = 144$ donc $r \equiv b_{14} - b_{28} \pmod{191} = 125$

Donc $r^{-1} \equiv 125 \pmod{191}$

Nous remplaçons r^{-1} par sa valeur dans la congruence suivante :

$r^{-1}.(a_{14} - a_{28}) \equiv 110 \bmod 191$ ce qui implique que $\log_2 228 = 110$

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x_i	228	279	92	184	205	14	28	256	152	304	372	121	12	144
a_i	0	0	0	1	1	1	2	2	2	3	3	6	6	12
b_i	1	2	4	4	5	6	6	7	8	8	9	18	19	38
x_{2i}	279	184	14	256	304	121	144	235	72	14	256	304	121	144
a_{2i}	0	1	1	2	3	6	12	48	48	96	97	98	5	10
b_{2i}	2	4	6	7	8	18	38	152	154	118	119	120	51	104

Exemple 28:

Nous allons calculer par l'algorithme de Pollard le logarithme discret de 3 en base 127 dans ℤ/1021ℤ. nous posons:

$G_1 = \{1 \ldots 340\}$, $G_2 = \{341 \ldots 680\}$, $G_3 = \{681 \ldots 1021\}$

Prenons $n_0 = 115$, on a $\beta_0 = 542$. Les valeurs stockées sont alors $(\beta_2{}^k, n_2{}^k, m_2{}^k)$

Pour $0 \le k \le 5$ (c'est-à-dire $2^5 = 32$):

k	β_k	n_k	m_k
2	418	116	1
4	555	233	2
8	124	467	6
16	550	855	26
16	346	734	217
58	346	314	281

Soit donc à résoudre : $3^{64} \equiv 127^{420} \bmod 1021$

Soit m le logarithme discret recherché $3 \equiv 127^m \pmod{1021}$, comme 127 est une base,

Nous obtenons : $64.m \equiv 420 \bmod 1020$

Or, pgcd (1020, 64) = 4 et 4 divise 420. Par conséquent, cette équation admet une seule solution modulo 1020/4 =255 et se ramène à: $16.m \equiv 105 \pmod{255}$

En utilisant l'algorithme d'Euclide étendu, on obtient : $1 = 16.16 - 255$ (1)

Par conséquent, $16 \cdot 16 \equiv 1 \pmod{255}$, puis en multipliant (1) par 16, on obtient :

$m \equiv 150 \pmod{255}$

Les solutions logarithmes discrets possibles sont donc les $m = 150 + 255k \pmod{1020}$

Tel que : $0 \le k < 4$. nous calculons alors : $127^{150} \equiv 3 \pmod{1021}$

Le logarithme discret de x = 3 en base 127 dans $\mathbb{Z}/1021\mathbb{Z}$ est donc m = 150.

Exemple 29: la factorisation du nombre 6944629145383337877043

A l'aide de :

l'algorithme ρ : 372ème itération 845951 x 8209256972783693 temps : 00:20
2725ème itération 7601089 x1080010637 temps : 01:21

l'algorithme *p-1 :* à la 60ème itération 7601089 x 913636078380787 temps: 00:05
à la 1000ème itération aucun diviseur temps : 01:13

l'algorithme *Fermat* : néme itération (n assez grand) aucun diviseur obtenu
temps : 35:05

Exemple 30:

L'entier p=809 est premier et on pourrait vérifier que l'élément α =89 est d'ordre n= 101 dans $(\mathbb{Z}/809\mathbb{Z})^*$ l'élément β = 618 est dans le sous groupe $< \alpha >$

Nous calculons $\log_\alpha \beta$

Nous définissons les ensembles G_1 G_2 G_3 comme suit:

$$G_1 = \{ x \in \mathbb{Z}/809\mathbb{Z} \text{ tel que } x \equiv 1 \bmod 3\}$$

$$G_2 = \{ x \in \mathbb{Z}/809\mathbb{Z} \text{ tel que } x \equiv 0 \bmod 3\}$$

$G_3 = \{ x \in \mathbb{Z}/809\mathbb{Z} \text{ tel que } x \equiv 2 \bmod 3\}$

Pour i = 1,2,….. nous obtenons les triplets (x_{2i}, a_{2i}, b_{2i}) et (x_i, a_i, b_i) suivants :

i	(x_i, a_i, b_i)	(x_{2i}, a_{2i}, b_{2i})
1	618 , 0 , 1	76 , 0 , 2
2	76 , 0 , 2	113 , 0 , 4
3	46 ,0 , 3	488 ,1 ,5
4	113 , 0 , 4	605 , 4 ,10
5	349 , 1, 4	422 , 5 , 11
6	488 ,1 ,5	683 ,7 ,11
7	555 , 2 , 5	451 ,8 ,12
8	605 , 4 ,10	344, 9 ,13
9	451, 5 ,10	112, 11,13
10	422, 5, 11	422, 11, 15

La première collision dans la table ci-dessus est $x_{10} = x_{20} = 422$, on a alors:

$C = (11-5)(11-5)^{-1} \bmod 101 = (6.25) \bmod 101 = 49$

Par conséquence $\log_{89}(618) = 49$ dans le groupe multiplicatif $(\mathbb{Z}/809\mathbb{Z})^*$

Conclusion :

L'algorithme de Pollard est un algorithme très efficace dans la factorisation des nombres qui ont moins de 20 chiffres décimaux et d'une complexité environ $o(\sqrt[4]{n})$.

Ⅳ.3. Algorithme de *Pohlig-Hellman* :

Soit p un nombre premier et soit g un générateur du groupe multiplicatif cyclique $\left(\frac{Z}{pZ}\right)^*$.

On veut résoudre l'équation à l'aide de la méthode de *Pohlig-Hellman* $g^a \equiv \beta \pmod p$.

Soi $p-1=\prod_{i=1}^{k} p_i^{\alpha_i}$ la décomposition de l'entier $p-1$ en produit de facteurs premiers distincts. La valeur de a est déterminée d'une manière unique modulo $p-1$.

Le théorème des restes Chinois implique que si on peut calculer $a \pmod{p_i^{\alpha_i}}$ pour tout i, $1 \le i \le k$, alors on peut calculer $a \bmod (p-1)$.

Soit q un nombre premier tel que $p-1 \equiv 0 \pmod{q^{\alpha}}$ et $p-1 \equiv 0 \pmod{p^{\alpha+1}}$.

Commençons par montrer comment calculer $x = a \bmod q^{\alpha}$ où $0 \le x \le q^{\alpha}-1$.

On écrit x en base q, $x=\sum_{i=0}^{\alpha-1} a_i q^i$ où $1 \le \alpha_i \le q-1$ pour $1 \le i \le \alpha-1$.

$x = a \bmod q^{\alpha}$ ést équivalente à $a = x + sq^{\alpha}$ pour un certain s donné.

On a : $\beta^{(p-1)/q} \equiv g^{(p-1)a_0/q} \pmod p$, en effet, comme , $g^{(p-1)/q} \equiv g^{(p-1)(x+q^{\alpha}s)/q} \pmod p$,

il suffit donc de montrer que $g^{(p-1)(x+q^{\alpha}s)/q} \equiv g^{(p-1)a_0/q} \pmod p$ ce qui est à montrer :

$$\frac{(p-1)(x+sq^{\alpha})}{q} q \equiv \frac{(p-1)a_0}{q} \pmod{p-1}.$$

On a $$\frac{(p-1)(x+sq^{\alpha})}{q} q - \frac{(p-1)a_0}{q} = (p-1)\left(\sum_{i=1}^{\alpha-1} a_i q^{i-1} + q^{\alpha} s\right) \equiv 0 \pmod{p-1}$$

Description de l'algorithme :

Etape 1 : calcul de a_0

On calcule $\beta^{(p-1)/q} \pmod p$.

Si $\beta^{(p-1)/q} \equiv 1 \pmod p$ alors $a_0 = 0$

Sinon on calcule successivement $\lambda = a^{(p-1)/q} \bmod p$, $\lambda^2 \bmod p$,...

On s'arrête dès qu'on obtient $\lambda^j \equiv \beta^{(p-1)/q} \pmod p$ pour un certain entier j, on prend alors $a_0 = j$

Si $c = 1$, on s'arrête. Si $c > 1$, on passe à l'étape suivante :

Etape 2 : calcule de a_1

1) On définit, $\beta_1 = \beta g^{-a_0}$ et on pose $x_1 = \log_g \beta_1 \bmod q^{\alpha}$, alors $x_1 = \sum_{i=1}^{\alpha-1} a_i q^i$.

2) On a, $\beta_1^{(p-1)/q^2} \equiv g^{(p-1)a_1/q} (\text{mod } p)$.

On calcule $\beta_1^{(p-1)/q^2} (\text{mod } p)$ et l'on cherche $\lambda^j \equiv \beta_1^{(p-1)/q^2} (\text{mod } p)$, on prend alors $a_1 = j$

Si $c = 2$, on a fini, sinon on passe à l'étape 3.

Etape 3 :

Continuer $c - 2$ fois pour obtenir les autres coefficients $a_2, a_3, ..., a_{c-1}$.

Exemple 31:

Supposons $p = 29$, on a

$$n = p - 1 = 28 = 2^2 .7$$

Supposons $\alpha = 2$ et $\beta = 18$; on cherche donc à déterminer $a = \log_2 18$. On calcule tout d'abord $a (\text{mod } 4)$ puis $a (\text{mod } 7)$

Commençons avec $q = 2$ et $c = 2$. On a

$$\gamma_0 = 1$$

et

$$\gamma_1 \equiv \alpha^{\frac{28}{2}} (\text{mod } 29)$$
$$\equiv 2^{14} (\text{mod } 29)$$
$$= 28$$

Ensuite,

$$\delta_1 \equiv \beta^{\frac{28}{2}} (\text{mod } 29)$$
$$\equiv 18^{14} (\text{mod } 29)$$
$$= 28$$

On a donc $a_0 = 1$. On calcule ensuite

$$\beta_1 \equiv \beta_0 . \alpha^{-1} (\text{mod } 29)$$
$$= 9$$

et $$\beta_1^{\frac{28}{4}} \equiv 9^7 (\text{mod } 29)$$
$$= 28$$

Comme $$\gamma_1 \equiv 28 (\text{mod } 29)$$

On a $a_1 = 1$. On a donc $a \equiv 3 (\text{mod } 4)$

On prend ensuite $q = 7$ et $c = 1$. On a $\beta^{\frac{28}{7}} \equiv 18^4 \pmod{29}$

$= 25$

et $\gamma_1 \equiv \alpha^{\frac{28}{7}} \pmod{29}$

$\equiv 2^4 \pmod{29}$

$= 16$

On peut alors calculer

$\gamma_2 = 24$

$\gamma_3 = 7$

$\gamma_4 = 25$

Donc $a_0 = 4$ et $a \equiv 4 \pmod{7}$

On obtient donc

$a \equiv 3 \pmod{4}$

$a \equiv 4 \pmod{7}$

$a \equiv 11 \pmod{28}$ avec le théorème des restes chinois. On a donc montré

$11 = \log_2 18$ dans $\mathbb{Z}/29\mathbb{Z}$.

Proposition 15 :

Notons DL(G) la complexité du calcul du logarithme discret dans le groupe G.

On a : $DL(G) \in o(\sum_j k_j DL(G_j' + \delta + (\log n)^3)$

où δ désigne O(log n) opérations dans G (multiplications, inversions).

□

Exemple 32:

Soit p = 29 et α =2 . p est un nombre premier et α un élément primitif modulo p .

On a $n = p - 1 = 28 = 2^2.7$

Nous prenons β = 18 et nous voulons alors déterminer $\alpha = \log_2 18$

Nous calculons a (mod 4) puis a (mod7)

Nous commençons avec q =2 et c = 2 et nous appliquons l'algorithme

Nous concluons $a_0 = 1$ et $a_1 = 1$ on a donc $a \equiv 3 \pmod 4$

Nous appliquons ensuite l'algorithme pour q = 7 et c = 1

nous trouvons $a_0 = 4$ nous avons donc $a \equiv 4 \pmod 7$

nous résolvons le système $a \equiv 3 \pmod 4$

$a \equiv 4 \pmod 7$

nous utilisons le théorème de restes chinois, alors nous trouvons $a \equiv 11 \pmod{28}$

Nous avons donc calculé $\log_2 18 \equiv 11 \pmod{29}$

Exemple 33 :

Calcul du logarithme discret par *la méthode de Pohlig-Hellman :*

Soit le nombre premier p = 8101, et $g = 6$ un générateur du groupe multiplicatif $(\mathbb{Z}/8101\mathbb{Z})^*$. Nous déterminons α tel que $g^{\alpha} \equiv 7531 \bmod (8101)$

La décomposition de $p-1$ en produit de facteurs premiers s'écrit $p-1 = 2^2.3^4.5^2$,

Nous résolvons les congruences suivantes : $x_2 \equiv \alpha \pmod{2^2}$ (1)

$x_3 \equiv \alpha \pmod{3^4}$ (2)

$x_5 \equiv \alpha \pmod{5^2}$ (3)

Solution de la congruence (1) :

Comme x_2 est une solution modulo 4, alors $x_2 = c_0 + 2c_1$ où $c_0, c_1 \in \{0,1\}$.

Déterminons les coefficients c_0, c_1 :

On a $7531^{(p-1)/2} = 7531^{4050} = -1 (\bmod p)$, d'où $g^{c_0(p-1)/2} = -1$ ce qui implique $c_0 = 1$.

En divisant 7531 par g^{c_0}, on obtient $7531g^{-1} = (7531)(6751) = 8006 \pmod{p}$.

Mais $8006^{(p-1)/4} = 7531^{2025} = 1$, d'où $g^{c_1(p-1)/2} = +1$, ce qui implique $c_1 = 0$

Ainsi, on a

$$x_2 = 1$$

Solution de la congruence (2) :

Comme x_3 est une solution modulo 81, alors $x_3 = c_0 + 3c_1 + 9c_2 + 27c_3$ où $c_0, c_1, c_2, c_3 \in \{0,1,2\}$.

Déterminons les coefficients c_0, c_1, c_2, c_3 :

Le calcul des puissances $g^{(p-1)/3}$ et $g^{2(p-1)/3}$, donne

$$g^{(p-1)/3} = 5883 \quad \text{et} \quad g^{2(p-1)/3} = 2217.$$

On trouve, $7531^{(p-1)/3} = 2217$, donc $c_0 = 2$.

En divisant 7531 par g^{c_0}, on obtient $7531g^{-2} = 6735 \pmod{p}$.

Et $6735^{(p-1)/9} = 1$, donc $c_1 = 0$.

En divisant 6735 par g^{3c_1}, on obtient $6735g^{0} = 6735 \bmod p$.

$6735^{(p-1)/27} = 2217$, donc $c_2 = 2$.

De la même façon, en divisant 6735 par g^{9c_2}, on obtient, $6735g^{-18} = 6992 \pmod{p}$,

$6992^{(p-1)/81} = 5883$, donc $c_3 = 1$

Ainsi, on trouve $x_3 = 47$

Solution de la congruence (3) :

Soit x_5 est une solution modulo 81, alors $x_3 = c_0 + 5c_1$ où $c_0, c_1 \in \{0,1,2,4\}$.

A l'aide de la même méthode et étapes précédentes, on trouve, $x_5 = 14$.

Solution de $g^{\alpha} \equiv 7531$:

Résolvons le système de congruences,

$$\begin{cases} \alpha \equiv 1 \pmod 4 \\ \alpha \equiv 47 \pmod{81} \\ \alpha \equiv 14 \pmod{25} \end{cases}$$

Grâce au théorème des restes Chinois, on trouve,

$$m_1 = \frac{8100}{4} = 2025, \quad \text{et} \quad y_1 = m_1^{-1} \bmod 4, \text{ implique } y_1 = 1$$

$$m_2 = \frac{8100}{81} = 100, \quad \text{et} \quad y_2 = m_2^{-1} \bmod 100, \text{ implique } y_2 = 64$$

$$m_3 = \frac{8100}{25} = 324, \quad \text{et} \quad y_3 = m_3^{-1} \bmod 25, \text{ implique } y_3 = 24$$

Donc $\alpha = (1)(2025)(1) + (47)(100)(64) + (14)(324)(24) = 6689 \bmod 8100$

IV.4. Algorithme d'*Adleman*:

L'algorithme de *Adleman* (index) est une méthode de résolution du problème logarithme discret dans les corps finis $\mathbb{F}_q$ tel que q est premier, ou dans $\mathbb{F}_{2^r}$ si p est premier et $p \approx 2^r$

Il est plus facile de résoudre ce problème dans $\mathbb{F}_{2^r}$ que dans $\mathbb{F}_p$ à l'aide de la méthode des indices. Cependant, il est plus pratique de programmer les cryptosystèmes dans $\mathbb{F}_{2^r}$.

Cet algorithme utilise la notion de friabilité dans $\mathbb{F}_{2^n}$; un élément est friable si sa décomposition ne fait intervenir que des « petits » facteurs, la friabilité est liée à la taille des éléments du groupe considéré. Dans le cas des corps $\mathbb{F}_{2^n}$, la friabilité d'un élément est liée à la factorisation du polynôme de plus petit degré parmi les représentants de l'élément en question. Si tous les facteurs de celui-ci sont de degré inférieur à une certaine borne m, on dit que cet élément est $m-$friable.

Exemple 34 :

On se place dans le corps $\mathbb{F}_{2^{127}} = \mathbb{F}_2[X]/(f(X))$ où $f(X) = X^{127} + X + 1$. Soit l'élément X^{400}.

On a, $X^{400} = X^{22} + X^{21} + X^{20} + X^{19} \bmod (X^{127} + X + 1)$

La factorisation de $X^{22} + X^{21} + X^{20} + X^{19}$ dans $\mathbb{F}_{2^{127}}$ s'écrit,

$$X^{22} + X^{21} + X^{20} + X^{19} = X^{19}(X+1)^3$$

Donc X^{400} est $1-$friable.

Description de l'algorithme : L'exécution de l'algorithme se fait en trois étapes.

On fixe d'abord le polynôme $f(X)$ qui définit le corps $K = F_{2^n}$, sur lequel nous ferons les calculs.

Etape 1 :

- On choisit une base B de facteurs : « petits » éléments $P \in K$:

$$B = \left\{ P_i \in K, P_i \text{ irréductible de } \deg \text{ré} \le m, \quad m \approx \sqrt{n} \right\}$$

- On choisit r au hasard $0 \le r \le 2^n - 1$ et on prend un élément primitif X.
- On calcule $X^r \bmod f$
- Si X^r est $m-$friable, on garde la décomposition $X^r \equiv \prod_i P^{\alpha_i} \bmod f$.(*)

Etape 2 :

- On itère l'opération précédente afin d'obtenir un nombre suffisant de relations (*).
- On prend le logarithme de base X, de chacune des relations.

- On écrit les équations obtenues : $m = \sum_i \alpha_i \log P_i \bmod (2^n - 1)$

Etape 3 :

- On prend un élément arbitraire Q de $K = F_{2^n}$.

- On calcule $X^r Q$ pour $0 \le r \le 2^n - 1$, on s'arrête dès qu'un élément friable est obtenu.

- Les valeurs $\log P_i$ étant connues, on calcule $\log Q$.

Exemple 35:

On considère le corps $\mathbb{Z}_2[x] / (x^3 + x + 1) = \{ a_0 + a_1 x + a_2 x^2 \ / a_i \in \mathbb{Z}_2 \}$ que nous noterons $\mathbb{Z}_2{}^3$. Nous effectuons les calculs dans $\mathbb{Z}_2{}^3$, on a $x^3 + x + 1 = 0$ donc $x^3 = -x - 1 = x + 1$, et les 7 éléments de $\mathbb{Z}^*_2{}^3$ engendré par x sont :

$x^1 = x,\ x^2 = x^2,\ x^3 = x + 1,\ x^4 = x^2 + x,\ x^5 = x^2 + x + 1,\ x^6 = x^2 + 1$ et $x^7 = 1$. Les exposants sont donc modulo 7. On a par exemple, $x^{12} = x^5 x^7 = x^5$.

Les logarithmes en base x, on obtient,

$\log_x (x^2 + x + 1) = 5$, puisque $x^5 = x^2 + x + 1$, $\log_x (x^2 + x) = 4$, $\log_x (x^2 + 1) = 6$, et $\log_x (x + 1) = 3$.

Soit $f(x)$ un polynôme de degré d irréductible modulo 2.

On a $\mathbb{Z}_q = \mathbb{Z}_2[x] / (f(x))$ où $q = 2^d$ et g est générateur de $\mathbb{Z}^*_q$.

La relation $g^n = y$ implique $\log_g y = n$ ou bien $\log y = n$

On a $\log(uv) \equiv \log u + \log v \ (\bmod q - 1)$

Etant donné g et y, le problème du logarithme discret dans $\mathbb{Z}_q$ consiste à résoudre l'équation $g^n = y$, autrement dit à trouver n modulo $q - 1$. ($n = \log_g y$).

En pratique , l'algorithme procède comme suit :

On choisit m tel que $1 < m < d$.

Etape 1 :

Trouver le logarithme de tous les polynômes de degré $\le m$:

- On calcule g^t et on écrit $g^t = h_1^{\alpha_1} h_2^{\alpha_2} ... h_r^{\alpha_r}$ avec $\deg r(h_i) \le m$

- On prend le logarithme pour obtenir un nombre suffisant d'équations linéaires en $\log h_i$:

$$t = \alpha_1 \log h_1 + \alpha_2 \log h_2 + ... + \alpha_r \log h_r .$$

- On pose : $a_i = \log h_i$ (les a_i $1 \le i \le r$ étant connus)

Etape 2 :

Calculer

- yg^t pour plusieurs valeurs de t jusqu'à obtenir $yg^t = h_1^{\beta_1}.h_2^{\beta_2}...h_r^{\beta_r}$.

- $\log y + t\log g = \log y + t = \beta_1 \log h_1 + \beta_2 \log h_2 + ... + \beta_r \log h_r = \beta_1 a_1 + \beta_2 a_2 + ... + \beta_r a_r$

L'inconnue est $\log y$.

Choix de m. Nous choisissons m entre 1 et d.

Exemple 36:

On considère le polynôme $f(x) = x^{11} + x^4 + x^2 + x + 1$. f est irréductible modulo 2.

On se place donc dans le corps $F_q = \dfrac{F_2[x]}{(f(x))}$ où $q = 2^{11}$. $g = x$ est un générateur de F_q^*.

On se propose de résoudre l'équation $g^n = x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + 1 = y$, c'est-à-dire trouver $\log y$. On choisit $m = 4$.

On a $\log g = \log x = 1$.

Posons : $a = \log(x+1)$, $c = \log(x^2 + x + 1)$, $d = \log(x^3 + x + 1)$, $e = \log(x^3 + x^2 + 1)$

$h = \log(x^4 + x + 1)$, $j = \log(x^4 + x^3 + 1)$, $k = \log(x^4 + x^3 + x^2 + x + 1)$.

Calculons les puissances de g, on a

$g^{11} = (x+1)(x^3 + x^2 + 1)$; $g^{41} = (x^3 + x^2 + 1)(x^3 + x + 1)^2$

$g^{56} = (x^2 + x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$; $g^{59} = (x+1)(x^4 + x^3 + x^2 + x + 1)^2$,

$g^{71} = (x^3 + x^2 + 1)(x^2 + x + 1)^2$; $g^{83} = (x^3 + x + 1)(x+1)^2$,

$g^{106} = (x+1)(x^4 + x^3 + 1)(x^4 + x^3 + x^2 + x + 1)$; $g^{126} = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x+1)^2$

D'où l'on obtient le système d'équations linéaires suivant :

$$(S)\begin{cases} a + e = 11 (\text{mod } 2047) & (1) \\ e + 2d = 41 (\text{mod } 2047) & (2) \\ c + d + e = 56 (\text{mod } 2047) & (3) \\ a + 2k = 59 (\text{mod } 2047) & (4) \\ e + 2c = 71 (\text{mod } 2047) & (5) \\ d + 2a = 83 (\text{mod } 2047) & (6) \\ a + j + k = 106 (\text{mod } 2047) & (7) \\ h + k + 2a = 126 (\text{mod } 2047) & (8) \end{cases}$$

Les équations (1), (2), (3) et (6), donnent

$a = 846$, $c = 453$, $d = 438$, $e = 1212$.

En remplaçant la valeur de $a = 846$ dans l'équation (4), on obtient $k = 630$

A partir des équations (7) on tire $j = 677$ et (8), donne $h = 1898$

La deuxième étape consiste à calculer $y(g^t)$ pour différentes valeurs de l'entier t.

On trouve :

$yg^{19} = (x^4 + x^3 + x^2 + x + 1)^2 = k^2$, ce qui implique $\log(y) + 19\log(g) = 2k$

d'où $\log(y) = 2k - 19 \equiv 1241 (\mathrm{mod}\, 2047)$ donc $x^{1241} = y$ par suite $n = 1241$.

On a donc $x^{1241} = x^9 + x^8 + x^6 + x^5 + x^3 + x^2 + 1 (\mathrm{mod}\, 2047)$

Exemple 37:

Supposons p =10007, α =5 qui est un élément primitif. Prenons $B = \{ 2, 3, 5, 7 \}$

Comme base de facteurs. Bien sûr, on a $\log_5 5 = 1$. Il y a donc trois logarithmes de la base à déterminer.

Des exemples de tirages d'exposants sont 4063,5136 et 9865.

Pour $x = 4063$, on calcule

$$5^{4063} (\mathrm{mod} 10007) \equiv 42 = 2.3.7$$

Cela donne la congruence

$$\log_5 2 + \log_5 3 + \log_5 7 \equiv 4063 \ (\mathrm{mod}\, 10006)$$

De la même manière, on a

$$5^{5136} (\mathrm{mod} 10007) \equiv 54 = 2.3^3$$

et $$5^{9865} (\mathrm{mod} 10007) \equiv 189 = 3^3.7$$

qui donnent $$\log_5 2 + 3\log_5 3 \equiv 5136 \ (\mathrm{mod} 10006)$$

et

$$3\log_5 3 + \log_5 7 \equiv 9865 \ (\mathrm{mod} 10006)$$

On a donc trois équitations linéaires aux inconnues $\log_5 2$, $\log_5 3$ et $\log_5 7$ modulos 10006 qui conduisent à l'unique solution $\log_5 2 = 6578$, $\log_5 3 = 6190$ et $\log_5 7 = 1301$.

Supposons maintenant que l'on cherche $\log_5 9451$. Si l'on tire l'exposant aléatoire s =7736, On a

$$9451.5^{7736} (\mathrm{mod} 10007) \equiv 8400.$$

Comme

$8400 = 2^4 3^1 .5^2 .7^1$ se factorise sur B, on a

$$\log_5 9451 \equiv 4\log_5 2 + \log_5 3 + 2\log_5 5 + \log_5 7 - s \ (\mathrm{mod}\, 10006)$$

$$\equiv 4\times 6578+6190+2+1301-7736 \pmod{10006}$$

$$\equiv 6057 \pmod{10006}$$

On vérifie que l'on a bien

$$5^{6057} \equiv 9451 \pmod{10007}.$$

Des analyses heuristiques de nombreuses versions de cet algorithme existent.

Sous des hypothèses raisonnables, le temps de calcul asymptotique du pré calcul est $O(e^{(1+o(1))\sqrt{\log p \log\log p}})$, et le temps de calcul de la recherche du logarithme discret est $O(e^{(1/2+o(1))\sqrt{\log p \log\log p}})$.

Chapitre V

Application en cryptographie

V .1 Introduction :

1.1 Introduction à la cryptographie :

La cryptographie : c'est la science du secret; crypter un message consiste à lui appliquer une série de transformations, afin de le rendre incompréhensible à toute personne qui n'en est pas le destinataire. La cryptographie est une discipline ancienne, traditionnellement étudiée par quelques élites travaillant pour des chefs d'états ou des gouvernements. A la fin des années 1970, les publications de chercheurs ont levé le voile sur cette discipline. La compétition économique et le besoin de protéger des données des regards indiscrets, ont fait de la cryptographie une science publique. Aujourd'hui, la cryptographie a envahi toute la vie civile; Elle permet d'assurer la sécurité des cartes à puces et du commerce électronique, de produire des signatures numériques.

V .1.2 Les principes de la cryptographie à clé publique:

Dans les années 1970, la cryptographie n'est plus seulement la caractéristique des militaires, elle se transmet aux domaines professionnels comme les banques, pour que la sécurité de leurs conventions soit essentielle. Elles sont donc devenues de grandes utilisatrices de messages codés. A cette époque il existait différents chiffrements disponibles, comme le *DES*.

Pour éviter plusieurs problèmes *Whitfield Diffie* et *Martin Hellman* proposent en 1976 une nouvelle méthode; il s'agit du chiffrement à clé publique.

Prenons par exemple, une société qui veut recevoir un message de haute importance d'une de ces filiales. La filiale décide donc de faire appel à un messager, mais comment être sûr que ce coursier n'ouvrira pas la lettre pour lire le message. Pour que le message ne soit pas lu, la société peut tout d'abord envoyer à sa filiale un cadenas sans sa clé en position ouverte, la filiale glissera alors le message dans une boite qu'elle fermera à l'aide du cadenas, puis elle transmettra la boite au coursier, et celui-ci ne pourra en n'aucun cas lire le message car c'est vous qui avez la clé du cadenas.

V .2 . *Le système R.S.A :*

Le cryptage RSA (du nom des inventeurs *Ronald Rivest*, *Adi Shamir* et *Leonard Adleman*)

est intéressant car la clé de cryptage est publique et il n'a donc pas de risques liés à l'envoi de la clé et au procédé de codage des données. *Mohamed* comme tout le monde peut crypter et envoyer un message. Par contre, seul le destinataire *Ali*, qui connaît la clé privée correspondante pourra reconstituer le message initial. *Ali*, le destinataire rend publique un triplet (RSA, N, e) où N est le produit de deux grands nombres premiers p et q qu'il est le seul à connaître,

où e est un entier premier avec le produit $(p-1)(q-1)$ compris entre 2 et $(p-1)(q-1)$.

V .2.1. Propriétés justifiant la méthode *RSA*:

Proposition 16:

Soient p et q deux nombres premiers et e un entier tel que, $1<e<(p-1)(q-1)$ et premier avec $(p-1)(q-1)$.

Alors il existe un unique entier d tel que $1<d<(p-1)(q-1)$ vérifiant :

$$ed \equiv 1 \pmod{(p-1)(q-1)}.$$

Preuve :

Supposons que $(p-1)(q-1)$ *et* e premiers entre eux, d'après le théorème de Bezout il existe deux entiers relatifs u et v vérifiant l'identité de Bezout ;

$$u(p-1)(q-1)+ve=1 \qquad (1)$$

Soient (r,s) un couple d'entiers relatifs vérifiant l'identité (1), alors

$$r(p-1)(q-1)+se=1 \qquad (2)$$

Les égalités (1) et (2) impliquent,

$$(r-u)(p-1)(q-1)=-(s-v)e \qquad (3)$$

D'après le théorème de *Gauss*, l'identité (3) implique,

$$(r-u) \text{ divise } e \text{ et } (p-1)(q-1) \text{ divise } v-s \qquad (4)$$

Les relations (4) impliquent, il existe un entier k tel que

$$r=u+ke \quad \text{et} \quad s=v-k(p-1)(q-1).$$

Soit donc k tel que u soit le plus grand des entiers négatifs vérifiant (3), alors v est le plus petit des entiers positifs.

Sous ces conditions, on a $ve=1-u(p-1)(q-1)$, par conséquent la valeur de d cherchée est $d=v$ d'où l'existence de d.

Unicité de l'entier d :

S'il existe un autre d' alors $e(d-d') \equiv 0 \pmod{(p-1)(q-1)}$. Comme $p \gcd(e,\ (p-1)(q-1)) = 1$, alors $d - d' \equiv 0 \pmod{(p-1)(q-1)}$, or $1 < d < (p-1)(q-1)$ et $1 < d' < (p-1)(q-1)$, on en déduit que $d' = d$ d'où l'unicité de d.

Proposition 17 :

Soient p et q deux nombres premiers distincts et e un entier tel que, $1 < e < (p-1)(q-1)$ et premier avec $(p-1)(q-1)$.

Si $b \equiv a^e \pmod{pq}$, alors $b^d \equiv a \pmod{pq}$.

Preuve :

Supposons $b \equiv a^e \pmod{pq}$ alors

$$b^d \equiv a^{de} \pmod{pq}. \qquad (1)$$

Par la proposition 13, on a $ed \equiv 1 \pmod{(p-1)(q-1)}$, il existe alors un entier k tel que

$$ed = 1 + k(p-1)(q-1). \qquad (2)$$

Les relations (1) et (2) impliquent,

$$a^{ed} = (a^p)^{k(q-1)} a^{1-k(q-1)} \qquad (3)$$

Le petit théorème de Fermat et la relation (3) impliquent,

$$a^{ed} \equiv a^{k(q-1)} a^{1-k(q-1)} \equiv a \pmod{p} \qquad (4)$$

Nous montrons de la même façon que l'on a,

$$a^{ed} \equiv a \pmod{q} \qquad (5)$$

De (4) et (5) on déduit qu'il existe deux entiers k et k' tels que,

$$a^{ed} \equiv a + k\,p \text{ et } a^{ed} \equiv a + k'q \quad (6)$$

Enfin (6) entraîne, $kp = k'q$, comme $(p,q) = 1$, alors l'entier $kp = k'q$ $k'q$ est un multiple de pq, on a donc

$$a^{ed} \equiv a \pmod{pq}$$

Ce qu'il fallait démontrer.

Algorithme : Génération d'un couple de clés RSA

Sorties : Un couple de clés R.S.A.
Générer deux très grands nombres premiers p et q.
Calculer N = p×q et d = (p - 1)×(q - 1).
Choisir un entier aléatoire e entre 1 et d, tel que pgcd(e,d) = 1.
Utiliser l'algorithme d'Euclide pour calculer l'entier e compris entre 1 et d tel que : e×d ≡1(mod N.

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

Chiffrement et déchiffrement

Pour chiffrer un message M, il faut d'abord le diviser en blocs de chiffres plus petits que N (avec des données binaires, choisir la plus grande puissance de 2 plus petite que N).

Si p et q ont tous les deux 100 chiffres, alors N aura un peu moins de 200 chiffres décimaux et chaque bloc de chiffres devra avoir un peu moins de 200 chiffres.

Pour obtenir un message chiffré C à partir d'un message clair M et d'une clé publique {e, N}

On calcule, $C \equiv M^e \pmod N$

Algorithme 2 : Chiffrement *RSA*

Entrées: Un message clair M, une clé publique {e,N}.
Sorties: Un message chiffré C.

$C \equiv M^e \pmod N$

Déchiffrement :

Pour restituer un message clair M à partir d'un message chiffré C et d'une clé privée {d, N}

Nous calculons, $M \equiv C^d \pmod N$

Algorithme 3 : Déchiffrement *RSA*

Entrées: Un message chiffré C, une clé publique {d,N}.
Sorties: Le message clair M.

$M \equiv C^d \pmod N$

Exemple 38:

Un émetteur quelconque "Mohamed" veut envoyer un message chiffré à "Ali".

1. Création des clés :

Pour se fabriquer un couple de clés, Ali commence par choisir deux nombres premiers aléatoires. Comme il s'agit ici d'un exemple, elle choisit deux petits nombres premiers.

$$p = 47$$

$$q = 71$$

Elle peut alors calculer N et d.

$$N = p \times q$$
$$= 3337$$
$$d = (p - 1)(q - 1) \quad \text{tel que} \quad d = \varphi(N)$$
$$= 46 \times 70 = 3220$$

Ali choisit aléatoirement l'exposant public e premier avec d

$$e = 79$$

Reste à calculer d, l'inverse de "e mod d".

Pour cela, *Ali* utilise l'algorithme étendu d'Euclide.

$e.d \equiv 1 \pmod{d}$ implique $d \equiv e^{-1} \pmod{d}$

$$e^{-1} \pmod{d} \equiv 79^{-1} \pmod{3220}$$

$$d \equiv 1019 \pmod{3220}$$

Ali choisit $d = 1019$

Alice peut maintenant publier sa clé publique constituée de (N , e) .

Elle efface ou détruit p et q et conserve précieusement d, qui est la clé secrète permettant de déchiffrer les messages qui lui seront envoyés.

2. Chiffrement :

Mohamed utilise la clé publique d'Ali pour chiffrer un message à son attention. Dans notre exemple, le message de Mohamed est 6882326879666683. Si le message est un texte, Mohamed le transforme en nombre à l'aide d'une convention publique de codage, comme le code ASCII par exemple. *Mohamed* commence par découper son message en blocs de chiffres plus petits que N.

$M_1 = 688$

$M_2 = 232$

$M_3 = 687$

$M_4 = 966$

$M_5 = 668$

$M_6 = 003$

Le premier bloc est chiffré avec la clé publique de *Ali* :

$$C_1 \equiv M^e_1 \pmod{N}$$

$$\equiv 688^{79} \bmod 3337$$

$$= 1570$$

Mohamed répète cette opération sur les autres blocs et obtient le message chiffré :

$$C = 1570\ 2756\ 2091\ 2276\ 2423\ 158$$

Il peut maintenant envoyer ce message à *Ali* par un canal non nécessairement sécurisé, comme la messagerie électronique par exemple.

3. Déchiffrement :

Pour déchiffrer le message, Ali utilise sa clé privée. Elle reconstitue le message clair contenu dans le premier bloc : $M_1 \equiv C^d_1 \pmod{N}$

$$\equiv 1570^{1019} \pmod{3337}$$

$$= 688$$

En répétant l'opération sur les autres blocs, elle retrouve le message clair :

688 232 687 966 668 003

La sécurité de *RSA* :

La sécurité de RSA repose sur la difficulté de la factorisation des grands nombres. Reconstituer un message clair à partir du message chiffré est par conjecture équivalent à factoriser le produit des deux nombres premiers. Cela n'est pas prouvé et on peut imaginer qu'un jour quelqu'un découvre une manière plus simple de le faire. Cependant, le fait que les cryptanalystes les plus brillants recherchent une telle méthode depuis 1976 et ne l'aient pas encore trouvée, permet d'accorder une grande confiance à l'algorithme R.S.A.

A titre d'information, Le 22 août 1999, un groupe de chercheurs est parvenu à factoriser un nombre de 512 bits (155 chiffres) en faisant travailler près de 285 ordinateurs, dont un supercalculateur, pendant prés de cinq mois. Voici ce nombre :

0941738641570527421809707322040357612003732945449205990913842131476349984288934784717997257891267332497625752899781833797076537244027146743531593354333897

= (026395928297411057720541965739916759007165678080380668033419335217907113077 79)

×(106603488380168454820927220360012878679207958575989291522270608237193062808643)

Exemple 39:

Ali, le récepteur rend publique un triplet (RSA, N, e) où $N = p \,.\, q$, p et q premiers,

où $d = \varphi(pq) = (p-1)(q-1)$ et e tel que $1 < e < (p-1)(q-1)$

donc $pgcd(e, (p-1)(q-1)) = 1$

Nous considérons le message : « ***ALGEBREETTHEORIEDESNOMBRES*** »

Chiffrement: Nous remplaçons les lettres par leurs positions dans l'ordre alphabétique dans la langue française "***01120705021805052020080515180905040519141513021805 19***"

Nous choisissons aléatoirement p et q tel que $p = 3121$, $q = 2731$, et nous calculons

$N = p \,.\, q = 8523451$

Nous choisissons e tel que $pgcd(e, \varphi(N)) = 1$, $e \in$ □ /N □, $e = 337$ convient.

Posons $M = B_1 .... B_k$ tel que $B_i < N$, nous obtenons la série

« ***0112070 5021805 0520200 8051518 0905040 5191415 1302180 0000519*** »

Nous calculons alors $C_i = M_i^e \bmod N$

$$C_1 = \mathbf{112070}^{337} \equiv 2982667 \ (\text{mod}8523451)$$

$$C_2 = \mathbf{5021805}^{337} \equiv 5245219 \ (\text{mod}8523451)$$

$C_3 = \mathbf{520200}^{337} \equiv 1040955$ **(**mod8523451)

$C_4 = \mathbf{8051518}^{337} \equiv 3273323$ **(**mod8523451)

$C_5 = \mathbf{905040}^{337} \equiv 4005429$ **(**mod8523451)

$C_6 = \mathbf{5191415}^{337} \equiv 910760$ **(**mod8523451)

$C_7 = \mathbf{1302180}^{337} \equiv 2649966$ **(**mod8523451)

$C_8 = 0000\mathbf{519}^{337} \equiv 720163$ **(**mod 8523451)

Mohamed envoie le résultat

« 2982667 5245219 1040955 3273323 4005429 910760 2649966 720163 » à *Ali*

Déchiffrement :

Ali calcule le nombre d vérifiant $1 < d < (p-1)(q-1)$ et $337 \times d \equiv 1$ (modulo 8517600), donc $d = 227473$.

Ali peut alors déchiffrer la série reçue tel que

$C^d \equiv M$ modulo N, il obtient le système de congruences suivantes

2982667^{227473}**(**mod8523451) ≡ ***112070***

5245219^{227473}**(**mod8523451) ≡ ***5021805***

1040955^{227473}**(**mod8523451) ≡ ***520200***

3273323^{227473}**(**mod8523451) ≡ ***8051518***

4005429^{227473}**(**mod8523451) ≡ ***0905040***

910760^{227473}**(**mod8523451) ≡ ***5191415***

2649966^{227473}**(**mod8523451) ≡ ***1302180***

720163^{227473}**(**mod8523451) ≡ ***519***

Donc M = ***01120705021805052020080515180905040519141513021805 19***

ALGEBRE ET THEORIE DES NOMBRES

Question : Le RSA est-il sûr ?

Les attaques actuelles du RSA se font essentiellement en factorisant l'entier N de la clé publique. La sécurité du RSA repose donc sur la difficulté de factoriser des grands entiers.

Le record établi en 1999, avec l'algorithme le plus performant et des moyens matériels considérables, est la factorisation d'un entier à 155 chiffres (soit une clé de 512 bits ~10^{155}. Il faut donc, pour garantir une certaine sécurité, choisir des clés plus grandes ; les spécialistes recommandent l'utilisation des clés de taille 768 bits pour un usage privé et des clés de 1024, voire 2048 bits, pour un usage sensible.

2eme Application sur le Cryptosysteme "EL GAMAL".

1.1 . El Gamal:

Le cryptosystème El Gamal est un système de chiffrement asymétrique. La sécurité de ce type de chiffrement repose en partie sur le problème du logarithme discret.

En effet, dans $(\mathbb{Z}/p\mathbb{Z})^*$ tel que p assez grand, l'exponentiation est réalisable relativement rapidement. Au contraire, le logarithme discret est énormément plus difficile et donc plus long. Nous présentons le fonctionnement du chiffrement ElGamal :

Paramètres publiques : Soit p , un grand nombre premier et g un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$

Initialisation : générer aléatoirement $x \in (\mathbb{Z}/p\mathbb{Z})^*$

$y \equiv g^x \bmod p$

Clé secrète : $K_s = x$

Clé publique : $K_p = y$

Message : M, un élément de $(\mathbb{Z}/p\mathbb{Z})^*$

Chiffrement : génère aléatoirement $r \in (\mathbb{Z}/p\mathbb{Z})^*$

$C = (u \equiv g^r \bmod p,\ v \equiv M \cdot y^r \bmod p)$

Déchiffrement : $M = v.u^{-x}$

Une particularité du chiffrement d'El Gamal est qu'il n'est pas déterministe. En effet, pour des paramètres identiques, tels que p, g est la paire de clés, deux chiffrements du même message clair M donneront deux messages chiffrés diffèrents. *Mohamed* veut communiquer avec *Ali* en utilisant un système de cryptographie symétrique.

Exemple 40 :

Soit le nombre premier p =181, un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$ est g = 23

Nous choisissons s = 7

Nous avons alors : x = 57 tel que $57 \equiv g^s \pmod p$

Nous obtenons la clé publique : (181, 23, 57)

et la clé privée : (18, 7)

Supposons que le message à envoyer soit une date de naissance :

Chiffrement :

p est de longueur 3, il nous faut prendre des blocs de longueur trois [7, 12, 90]

Qui pour l'ordinateur devient: [07, 12, 90]

Nous choisissons un nombre k au hasard compris entre 2 et p -1, et nous calculons

$$\alpha = g^k = 23^6 \equiv 152 (\text{mod} 181),$$

Puis nous chiffrons chaque élément de ce tableau et $x^k \equiv 57^6 (\text{mod}181)$:

$$7\times 57^6 \equiv 146 \ (\text{mod}181)$$

$$12\times 57^6 \equiv 121 \ (\text{mod}181)$$

$$90\times 57^6 \equiv 93 \ (\text{mod}181)$$

Ce qui nous donne : [146, 121, 93]

Déchiffrement :

Nous recevons le message précédent que nous décompose en bloc de longueur 3 ce qui donne [146, 121, 93].

Nous calculons alors l'inverse de α^s modulo p, et nous obtenons $\alpha^{-s} = 36$

Pour chaque élément *a* du tableau nous calculons $a.\alpha^{-s}$ mod 181,

Nous obtenons le tableau: [7, 12, 90]

il reste à compléter chaque bloc par des 0.

Nous obtenons [07, 12,90]

Ce qui correspond à notre message 071290

2/ à l'aide du logiciel "Mathematica"

```
<< NumberTheory`NumberTheoryFunctions`
```

p = Prime 180

1069

p = NextPrime 180

181

g = 23

23

s = 7

7

```
x= Mod g^s, p
```

57

```
m = 7,12,90
```

```
m = 7,12,90
```

152

a = Modg^6,p

c = Mod g^6*m, p

mod148035889m,181

m_1 = 07

7

m_2 = 12

12

m_3 = 90

90

C_1=Mod x^m_1, p

146

C_2=Mod x^6*m_2, p

121

C_3=Mod x^6*m_3, p

93

i=Mod a^(-1) , p

$$\frac{1}{152}$$

i= **PowerMod** a^(-s) , p

36

m_1 =Mod c_1*i,p

7

m_2 =Mod c_2*i,p

12

m_3 =Mod c_3*i,p

90

Exemple 48 : le texte qui sera chiffré est donc

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMADIENNE

Nous utilisons la clé publique:

$$[\,2038074743,\ 179424673,\ 179424673^{1788785923}\,]$$

Nous obtenons le message chiffré suivant:

1657044171,897170612,1519017923,309829863,793650926,275323301,
1519017923,966183736,793650926,1726786490,102790491,793650926,
275323301,1726786490,275323301,1450004799,1519017923,793650926
,897170612,1450004799,793650926,275323301,1726786490,793650926,
966183736,1726786490,102790491,793650926,1726786490,1553524485,
68283929,1726786490,966183736,793650926,1450004799,828157488,
897170612,1588031047,1553524485,1588031047,137297053,151901792
3,793650926,1726786490,828157488,1588031047,1657044171,6828392
9,1622537609,1519017923,1726786490,759144364,1588031047,165704
4171,206310177,793650926,102790491,1519017923,793650926,897170
612, 897170612,793650926.

2/ à l'aide du logiciel "Mathematica"

```
<< NumberTheory`NumberTheoryFunctions`
```

p = Prime[10^8]

2038074743

g = Prime[10^7]

179424673

s = Prime[10^6]

15485863

x = PowerMod[g,s,p]

17887859 23

m= ToCharacterCode["UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMADIENNE"]

85, 78, 73, 86, 69, 83, 73, 84, 69, 32, 68, 69, 83, 32, 83, 67, 73, 69, 78, 67,
69, 83, 32, 69, 84, 32, 68, 69, 32, 76, 65, 32, 84, 69, 67, 72, 78, 79, 76, 79, 71,
73, 69, 32, 72, 79, 85, 65, 82, 73, 32, 66, 79, 85, 77, 69, 68, 73, 69, 78, 78, 69

K = 2008

alpha = PowerMod [g, k, p]

811805588

cryptogramme = Mod [x^k , p]

1657044171,897170612,1519017923,309829863,793650926,275323301,
1519017923,966183736,793650926,1726786490,102790491,793650926,
275323301,1726786490,275323301,1450004799,1519017923,793650926,
897170612,1450004799,793650926,275323301,1726786490,793650926,
966183736,1726786490,102790491,793650926,1726786490,1553524485,
68283929,1726786490,966183736,793650926,1450004799,828157488,
897170612,1588031047,1553524485,1588031047,137297053,1519017923,
793650926,1726786490,828157488,1588031047,1657044171,68283929,
1622537609,1519017923,1726786490,759144364,1588031047,165704417
1,206310177,793650926,102790491,1519017923,793650926,897170612,
897170612,793650926

inverse = PowerMod[alpha,-s,p]

1693693O9

clair = Mod[crypptogramme * inverse , p]

85,78,73,86,69,83,73,84,69,32,68,69,83,32,83,67,73,69,78,67,
69,83,32,69,84,32,68,69,32,76,65,32,84,69,67,72,78,79,76,79,71,
73,69,32,72,79,85,65,82,73,32,66,79,85,77,69,68,73,69,78,78,69

FromCharacterCode [clair]

UNIVESITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENNE

Conclusion :

Les meilleurs algorithmes de factorisation actuels peuvent factoriser des nombres jusqu'à 230 chiffres en plusieurs mois de calculs avec quelques centaines de machines. Cela semble être la limite actuelle de calcul. De plus, cette limite est régulièrement reculée entre 1999 et 2000, le record de factorisation pour les clés RSA est passé de 140 à 155 chiffres décimaux.

Pour cela, et à travers ce mémoire, nous soulignons à présent l'importance de la course aux nombres premiers.

Cette ***recherche****, au début est devenue avec le développement dans le domaine de la cryptographie une véritable course entre les différents laboratoires et même chez les* ***chercheurs****, pour chaque nouveau grand* ***nombre premier*** *découvert son auteur est recomposé. Cette recherche évolue en deux directions :*

- *La première s'effectue à l'aide d'****ordinateurs*** *que des gens n'hésitent pas à faire tourner des mois et des mois pour découvrir un nouveau nombre premier.*
- *La seconde se tourne vers l'algorithmique; l'algorithme d'****Agrawal Kayal*** *et* ***Saxena*** *(AKS) étant le dernier algorithme. Cependant cet algorithme bien que très innovant, au sens où il est le premier à avoir une complexité polynomiale, a néanmoins le défaut non négligeable d'avoir une portée pratique très limitée : en effet, les algorithmes classiques de primalité tels que le test de* ***Miller-Rabin*** *sont beaucoup plus rapides pour les entiers que l'on peut tester avec la puissance actuelle de nos ordinateurs ; malheureusement, ces algorithmes sont* ***probabilistes****.*

Il reste donc encore beaucoup de recherches dans le domaine de la primalité en théorie des nombres pour surmonter toutes les difficultés puis les appliquer en ***cryptographie****.*

Références :

[1] L. ADLEMAN, C. POMERANCE and RUMELY. *On distinguishing prime numbers from composite numbers*. Ann. of Math.(2),117:173-206,1983;

[2] Xavier ALEXANDRE & Steve Bennoun. *Méthodes de factorisation,* 18 juin 2004; Sous la direction du Dr Diego Kuonen, - Dr Gérard Maze et de Lorenz Minder.

[3] Tom. M . APOSTOL. *Introduction to Analytic Number Theory*. New York Heidelberg Berlin 1976;

[4] F. ARNAULT. *Sur quelques tests probabilistes de primalité* , Thèse de Doctorat Université Poitiers Décembre 1993;

[5] Eric BACH et JEFFREY SHALLIT. *Algorithmic Numbers Theory Volume 01: Efficient Algorithms*, The MIT Press Cambridge, Massachusetts-London England,1996;

[6] Schneider BRUCE. *Cryptographie Appliquée, Protocoles Algorithmes et Codes Sources en C,* Edition corrigée et traduite par Marc VAUCLAIR, Paris, 1995;

[7] Johannes . A. BUCHMANN. *Introduction to cryptography*. Springer, 2001;

[8] Steve BURNETT and STEPHEN PAINE. *RSA security's official guide to cryptography* Mc Graw-Hill Company California USA 2001 ;

[9] W.W. L. CHEN. *Elementary and Analytic Number Theory*, University of London,1981;

[10] H. COHEN . *A course in algorithmic algebraic number theory*, volume 138 of graduate texts in Mathematics Verlag, Third printing, 1996.

[11] H. Cohen and A.K.LENSTRA. *Implementation of new primality test*. Math. comp, 48(177)103-121,1987;

[12] G.Cohen, A.LOBSTEIN et J.P.BARTHELEMY. *Complexité Algorithmique et problèmes de communication*, Masson 1992;

[13] Richard CRANDALL and Carl POMERANCE. *Prime numbers. A computational perspective*. Springer- *Second Edition*, 2005;

[14] Jean Paul DELAHAYE. *Merveilleux nombres premiers,voyage en cœur de l'arithmétique*, Paris Amazon, 2000;

[15] Etienne FOUVRY. *Théorème de Brun- Titchmarsh; application au théorème de Fermat*, Invent. Math. 79, 383-407, 1985;

[16] K.IRELAND and M. ROSEN . *A classical introduction to modern number theory*, volume 84, Of Graduate Texts in Mathematics Springer, 1982;

[17] Antoine JOUX & Reynald LERCIER. *Algorithmes pour résoudre le problème du logarithme discret dans les corps finis*, In Nouvelles Méthodes Mathématiques en Cryptographie, Fascicule *Journées Annuelles*, pages : 23–53. Société Mathématique de France, Juin 2007;

[18] H.X.Mel et DORIS BAKER - collaboration de steve Burnett et Johen Kinyon. *La cryptographie décryptée*, juillet 2001;

[19] A.MENEZES, P.VAN OORSCHOT and S.VANSTONE. *Handbook of Applied Cryptography*, CRC Press 1997;

[20] Benjamin ODGERS. *The Distribution of Prime Numbers & The Riemann Zeta Function*, 2002;

[21] Paulo RIBENBOIM. *Nombres premiers : mystères et records,* France, Décembre 1994 *;*

[22] William STEIN. *An Explicit Approach to Elementary Numbers Theory*, Harvard University Fall 2001;

[23] Douglas STINSON. *Cryptographie- Théorie et pratique*, vuibert, 2003;

[24] Steve TEIXEIRA and Xavier PACHECO: *Delphi* ^{TM} *6 Developer's Guide* version Traduite par Olivier ENGLER, SANS Publishing indianapolis Indiana, USA 2001 ;

[25] Lawrance C.WASHINGTON . *Elliptic Curves Number Theory and Cryptography* second Edition , University of Maryland College Park Maryland USA 2008 ;

[26] Jason WOJCIECHOWSKI. *Modern Primality Tests and the Agrawal-Kayal – Saxena Algorithm* Hampshire College, Amherst, Massachusetts. Kell Hoffman, David Kelly. April 2003.

Annexe :

Le langage utilisé *Delphi* est un outil de développement puissant pour la programmation d'applications pour *Windows* il ne contient pas seulement des composants permettant de créer facilement des interface d'applications, mais aussi des zones de listes, des boutons ou des boîtes de dialogue toutes prêtes. *Delphi* est un outil à double usage puisqu'il permet de créer des applications de deux façons différentes. La première méthode est visuelle et utilise le glisseur et déplaceur à l'écran. La deuxième méthode consiste en une programmation traditionnelle, celle où l'on produit du code source.

Description de l'interface graphique :

L'interface de notre programme est orientée vers la démonstration des générations et chiffrement/déchiffrement RSA et notre implémentation de cet algorithme en cryptographie. Elle est présentée en deux applications indépendantes :

(1) La première version est consacrée à la génération de clés utilisées dans l'opération chiffrement / déchiffrement.

(2) La deuxième application version: 1.0.0/ 2009 USTHB est conçue spécialement pour l'opération chiffrement /déchiffrement avec deux couples de clés (e, N), (d, N) générées d'une manière aléatoire à l'aide de la commende 'Random'.

Application.1 :

Cette première application consiste à générer toutes les clés utilisées dans l'opération chiffrement et déchiffrement. L'affichage de ces couples de nombres (e, N) et (d, N) est en Hexadécimal, ce qui permet d'afficher des grands nombres écrits dans le système décimal ou binaire.

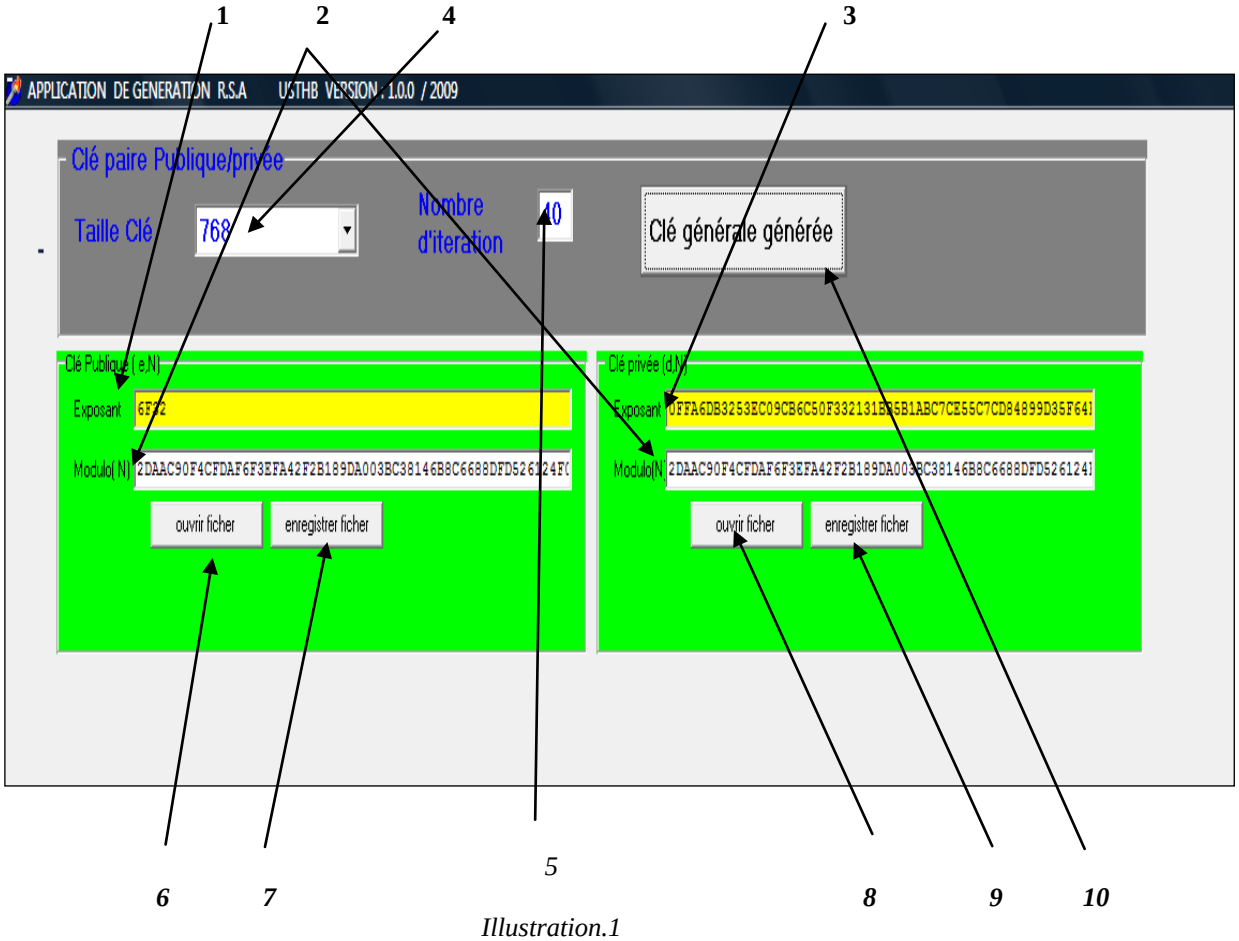

Illustration.1

(1) exposant e en Hexadécimal de la clé publique incluse dans le chiffrement de RSA .

(2) N est produit de deux nombres premiers p et q en Hexadécimal.

(3) exposant d en Hexadécimal de la clé privée incluse dans le déchiffrement de RSA.

(4) La taille de la clé varie entre 128, 256, 512, 768 et 1024 bits.

(5) Nombre d'itérations souhaitables dans la génération de la clé générale

(6) Bouton d'ouverture d'un fichier contenant une clé publique (e, N)

(7) Bouton d'enregistrement d'un fichier contenant une clé publique (e, N)

(8) Bouton d'ouverture d'un fichier contenant une clé publique (d, N)

(9) Bouton d'enregistrement d'un fichier contenant une clé publique (e, N)

(10) Bouton de génération d'une clé générale entre dans l'opération chiff / déchiff

Application.2 :

Cette deuxième application consiste à chiffrer et à déchiffrer un texte en clair par une clé générée automatiquement à l'aide d'une commande 'Random', cette clé varie entre 128,256,512, 768 et 1024 bits.

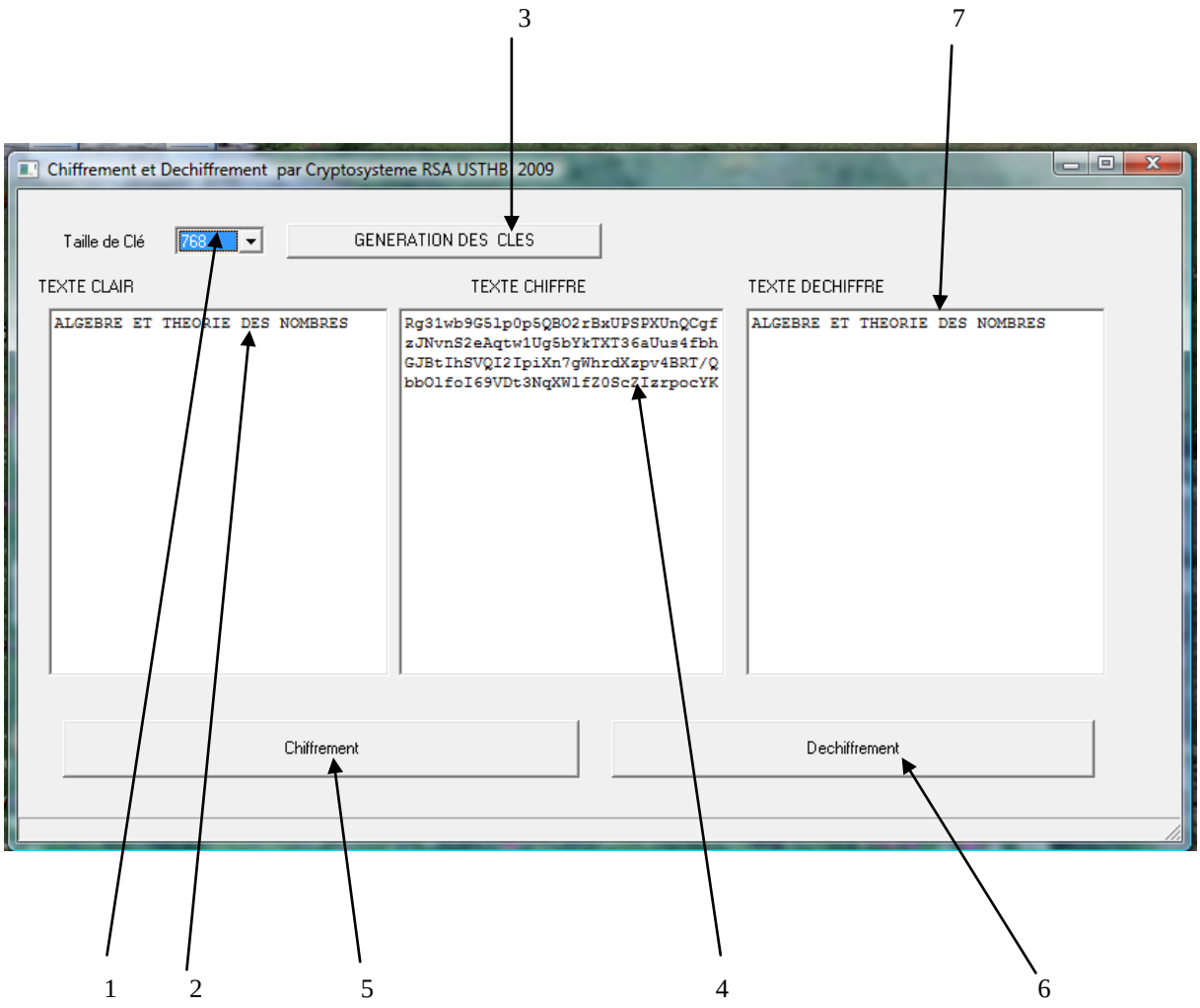

Illustration.2

(1) Taille de la clé choisie parmi les 128, 256, 512, 768 et 1024 bits;

(2) Le texte en clair;

(3) Bouton de la clé générale générée automatiquement ;

(4) Le texte chiffré par le premier couple (e , N);

(5) Bouton de l'opération chiffrement;

(6) Bouton de l'opération de déchiffrement (d , N);

(7) Le texte en clair , résultat de l'opération déchiffrement.