Tests de primalité, RSA, logarithme discret, Diffie-Hellman et El Gamal classique

Abdelmalek Azizi

Ecole de Recherche CIMPA-Mauritanie
Théorie Algorithmique des Nombres et Cryptographie
Nouakchott 15-26 Février 2016

12 février 2016

Plan du Cours

Plan du Cours

1. Introduction,

Plan du Cours

1. Introduction,
2. Généralités d'Arithmétique,

Plan du Cours

1. Introduction,
2. Généralités d'Arithmétique,
3. Tests de primalité,

Plan du Cours

1. Introduction,
2. Généralités d'Arithmétique,
3. Tests de primalité,
4. Méthode RSA,

Plan du Cours

1. Introduction,
2. Généralités d'Arithmétique,
3. Tests de primalité,
4. Méthode RSA,
5. Logarithme Discret, Diffie-Hellman et ElGamal,

Plan du Cours

1. Introduction,
2. Généralités d'Arithmétique,
3. Tests de primalité,
4. Méthode RSA,
5. Logarithme Discret, Diffie-Hellman et ElGamal,
6. Signatures et Attaques.

Introduction

Introduction

Comme disait Kronecker(1823 - 1852) ; Dieu a crée les entiers naturels et l'homme a fait le reste.

Introduction

Comme disait Kronecker(1823 - 1852) ; Dieu a crée les entiers naturels et l'homme a fait le reste. Dieu a crée les entiers en même temps que l'univers.

Introduction

Comme disait Kronecker(1823 - 1852) ; Dieu a crée les entiers naturels et l'homme a fait le reste. Dieu a crée les entiers en même temps que l'univers. L'homme a su, avec son génie, comprendre et utiliser les entiers et d'autres nombres.

Introduction

Comme disait Kronecker(1823 - 1852) ; Dieu a crée les entiers naturels et l'homme a fait le reste. Dieu a crée les entiers en même temps que l'univers. L'homme a su, avec son génie, comprendre et utiliser les entiers et d'autres nombres.
L'homme s'est intéressé à l'étude de certains problèmes de nombres depuis des périodes très reculées.

Introduction

Comme disait Kronecker(1823 - 1852) ; Dieu a crée les entiers naturels et l'homme a fait le reste. Dieu a crée les entiers en même temps que l'univers. L'homme a su, avec son génie, comprendre et utiliser les entiers et d'autres nombres.
L'homme s'est intéressé à l'étude de certains problèmes de nombres depuis des périodes très reculées. Les problèmes étudiés provenaient aussi bien de son activité économique (commerce, poids et mesure) que de ses préoccupations astronomiques (calendrier, astrologie).

Introduction

Comme disait Kronecker(1823 - 1852) ; Dieu a crée les entiers naturels et l'homme a fait le reste. Dieu a crée les entiers en même temps que l'univers. L'homme a su, avec son génie, comprendre et utiliser les entiers et d'autres nombres.
L'homme s'est intéressé à l'étude de certains problèmes de nombres depuis des périodes très reculées. Les problèmes étudiés provenaient aussi bien de son activité économique (commerce, poids et mesure) que de ses préoccupations astronomiques (calendrier, astrologie).
Les Babyloniens, Les Romains, les Grecs, les Arabes et d'autres civilations ont mis la théorie des nombres sur le bon chemin. Les Européens des six derniers siècles ont bien développés cette théorie. Ils ont résolu plusieurs problèmes , ils ont laissé d'autres sous forme de conjectures et ils ont orienté cette théorie vers plusieurs axes de recherches théoriques et appliqués. Parmis les axes appliqués on trouve la Cryptographie.

Introduction

Introduction

Une méthode arithmétique de cryptographie avait été utilisé dans le Maghreb à la fin du 16 ième siècle :

Introduction

Une méthode arithmétique de cryptographie avait été utilisé dans le Maghreb à la fin du 16 ième siècle : cette méthode, a été inventé par le Sultan Al Mansour au Maroc,

Introduction

Une méthode arithmétique de cryptographie avait été utilisé dans le Maghreb à la fin du 16 ième siècle : cette méthode, a été inventé par le Sultan Al Mansour au Maroc, et consiste à utiliser des transformations arithmétiques sur le texte clair(multiplication, addition et factorisation),

Introduction

Une méthode arithmétique de cryptographie avait été utilisé dans le Maghreb à la fin du 16 ième siècle : cette méthode, a été inventé par le Sultan Al Mansour au Maroc, et consiste à utiliser des transformations arithmétiques sur le texte clair(multiplication, addition et factorisation), et n'avait pas à ma connaissance d'équivalent, ni dans le Moyen-Orient, ni en Europe, avant 1690 où on trouve, après cette date, plusieurs travaux sur la substitution polyalphabétique

Introduction

Une méthode arithmétique de cryptographie avait été utilisé dans le Maghreb à la fin du 16 ième siècle : cette méthode, a été inventé par le Sultan Al Mansour au Maroc, et consiste à utiliser des transformations arithmétiques sur le texte clair(multiplication, addition et factorisation), et n'avait pas à ma connaissance d'équivalent, ni dans le Moyen-Orient, ni en Europe, avant 1690 où on trouve, après cette date, plusieurs travaux sur la substitution polyalphabétique et en particulier le chiffre de Vigenère décrit par le scientifique français Claude Comiers en utilisant l'addition d'une clé avec le texte clair, modulo 26.

Soient a et b deux entiers naturels non nuls ; d'après la division Euclidienne, si b est plus grand que a il existe q et r deux entiers tels que $b = a.q + r$ où $r < a$.

Soient a et b deux entiers naturels non nuls ; d'après la division Euclidienne, si b est plus grand que a il existe q et r deux entiers tels que $b = a.q + r$ où $r < a$.
On dit que a divise b si et seulement si il existe d un entier naturel tel que $b = ad$.

Soient a et b deux entiers naturels non nuls ; d'après la division Euclidienne, si b est plus grand que a il existe q et r deux entiers tels que $b = a.q + r$ où $r < a$.
On dit que a divise b si et seulement si il existe d un entier naturel tel que $b = ad$.
Posons $r_0 = b$ et $r_1 = a$ on calcule jusqu à obtenir un reste nul.

Soient a et b deux entiers naturels non nuls ; d'après la division Euclidienne, si b est plus grand que a il existe q et r deux entiers tels que $b = a.q + r$ où $r < a$.
On dit que a divise b si et seulement si il existe d un entier naturel tel que $b = ad$.
Posons $r_0 = b$ et $r_1 = a$ on calcule jusqu à obtenir un reste nul.

1. si $r_1 \neq 0$ on a $r_0 = r_1 q_1 + r_2$ avec $r_2 < r_1$;

Soient a et b deux entiers naturels non nuls ; d'après la division Euclidienne, si b est plus grand que a il existe q et r deux entiers tels que $b = a.q + r$ où $r < a$.
On dit que a divise b si et seulement si il existe d un entier naturel tel que $b = ad$.
Posons $r_0 = b$ et $r_1 = a$ on calcule jusqu à obtenir un reste nul.

1. si $r_1 \neq 0$ on a $r_0 = r_1 q_1 + r_2$ avec $r_2 < r_1$;
2. si $r_2 \neq 0$ on a $r_1 = r_2 q_2 + r_3$ avec $r_3 < r_2$;

Soient a et b deux entiers naturels non nuls ; d'après la division Euclidienne, si b est plus grand que a il existe q et r deux entiers tels que $b = a.q + r$ où $r < a$.
On dit que a divise b si et seulement si il existe d un entier naturel tel que $b = ad$.
Posons $r_0 = b$ et $r_1 = a$ on calcule jusqu à obtenir un reste nul.

1. si $r_1 \neq 0$ on a $r_0 = r_1 q_1 + r_2$ avec $r_2 < r_1$;
2. si $r_2 \neq 0$ on a $r_1 = r_2 q_2 + r_3$ avec $r_3 < r_2$;

$\vdots$

Soient a et b deux entiers naturels non nuls ; d'après la division Euclidienne, si b est plus grand que a il existe q et r deux entiers tels que $b = a.q + r$ où $r < a$.
On dit que a divise b si et seulement si il existe d un entier naturel tel que $b = ad$.
Posons $r_0 = b$ et $r_1 = a$ on calcule jusqu à obtenir un reste nul.

1. si $r_1 \neq 0$ on a $r_0 = r_1 q_1 + r_2$ avec $r_2 < r_1$;
2. si $r_2 \neq 0$ on a $r_1 = r_2 q_2 + r_3$ avec $r_3 < r_2$;

$\vdots$

3. si $r_{n-2} \neq 0$ on a $r_{n-2} = r_{n-1} q_{n-1} + r_n$ avec $r_n < r_{n-1}$;
4. si $r_{n-1} \neq 0$ on a $r_{n-1} = r_n q_n + r_{n+1}$ avec $r_{n+1} = 0$.

En particulier, on a :

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.
2. Il existe u et v dans Z tels que $a.u + b.v = r_n$ et u et v se calculent à l'aide des équations d'Euclide précédentes.

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.
2. Il existe u et v dans Z tels que $a.u + b.v = r_n$ et u et v se calculent à l'aide des équations d'Euclide précédentes.
3. si a et b sont premiers entre eux, leurs PGCD est égal à 1 et on obtient l'identité de Bezout :

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.
2. Il existe u et v dans Z tels que $a.u + b.v = r_n$ et u et v se calculent à l'aide des équations d'Euclide précédentes.
3. si a et b sont premiers entre eux, leurs PGCD est égal à 1 et on obtient l'identité de Bezout :
   Il existe u et v dans Z tels que $a.u + b.v = 1$.

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.
2. Il existe u et v dans Z tels que $a.u + b.v = r_n$ et u et v se calculent à l'aide des équations d'Euclide précédentes.
3. si a et b sont premiers entre eux, leurs PGCD est égal à 1 et on obtient l'identité de Bezout :
   Il existe u et v dans Z tels que $a.u + b.v = 1$.
4. Soient n un entier naturel supérieur à 2 et $\mathbf{Z}/n\mathbf{Z}$ l'ensemble des classes d'équivalences modulo n.

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.
2. Il existe u et v dans Z tels que $a.u + b.v = r_n$ et u et v se calculent à l'aide des équations d'Euclide précédentes.
3. si a et b sont premiers entre eux, leurs PGCD est égal à 1 et on obtient l'identité de Bezout :
   Il existe u et v dans Z tels que $a.u + b.v = 1$.
4. Soient n un entier naturel supérieur à 2 et $\mathbf{Z}/n\mathbf{Z}$ l'ensemble des classes d'équivalences modulo n. Un entier a est inversible modulo n si et seulement si a est premier avec n ;

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.
2. Il existe u et v dans Z tels que $a.u + b.v = r_n$ et u et v se calculent à l'aide des équations d'Euclide précédentes.
3. si a et b sont premiers entre eux, leurs PGCD est égal à 1 et on obtient l'identité de Bezout :
   Il existe u et v dans Z tels que $a.u + b.v = 1$.
4. Soient n un entier naturel supérieur à 2 et $\mathbf{Z}/n\mathbf{Z}$ l'ensemble des classes d'équivalences modulo n. Un entier a est inversible modulo n si et seulement si a est premier avec n ; ce qui est équivalent d'après l'identité de Bezout à l'existence de u et v dans $\mathbf{Z}$ tels que
   $$a.u + n.v = 1.$$

En particulier, on a :

1. Le PGCD de a et b est le dernier reste non nul.
2. Il existe u et v dans Z tels que $a.u + b.v = r_n$ et u et v se calculent à l'aide des équations d'Euclide précédentes.
3. si a et b sont premiers entre eux, leurs PGCD est égal à 1 et on obtient l'identité de Bezout :
   Il existe u et v dans Z tels que $a.u + b.v = 1$.
4. Soient n un entier naturel supérieur à 2 et $\mathbf{Z}/n\mathbf{Z}$ l'ensemble des classes d'équivalences modulo n. Un entier a est inversible modulo n si et seulement si a est premier avec n ; ce qui est équivalent d'après l'identité de Bezout à l'existence de u et v dans $\mathbf{Z}$ tels que

$$a.u + n.v = 1.$$

   Alors, on déduit de ce qui précède que u est l'inverse de a modulo n.

Parmi les plus anciens théorèmes d'arithmétique on trouve le théorème des restes chinois :

Parmi les plus anciens théorèmes d'arithmétique on trouve le théorème des restes chinois :

Théorème

Théorème Chinois : Soient n_1, n_2,..., n_r des entiers naturels premiers entre eux deux à deux, a_1, a_2,..., a_r des entiers relatifs ; alors il existe un entier x unique modulo $N = n_1 n_2 \ldots n_r$ tel que $x \equiv a_i \bmod n_i$ pour tout $i = 1, \ldots, r$. Soient $N_i = N/n_i$ et u_i l'inverse de N_i modulo n_i, alors

$$x = \sum_{1}^{r} a_i u_i N_i.$$

Parmi les plus anciens théorèmes d'arithmétique on trouve le théorème des restes chinois :

Théorème

Théorème Chinois : Soient n_1, n_2,..., n_r des entiers naturels premiers entre eux deux à deux, a_1, a_2,..., a_r des entiers relatifs ; alors il existe un entier x unique modulo $N = n_1 n_2 \ldots n_r$ tel que $x \equiv a_i \bmod n_i$ pour tout $i = 1, \ldots, r$. Soient $N_i = N/n_i$ et u_i l'inverse de N_i modulo n_i, alors

$$x = \sum_{1}^{r} a_i u_i N_i.$$

On peut voir ce dernier théorème, comme conséquence du théorème suivant :

Parmi les plus anciens théorèmes d'arithmétique on trouve le théorème des restes chinois :

Théorème

Théorème Chinois : Soient n_1, n_2,..., n_r des entiers naturels premiers entre eux deux à deux, a_1, a_2,..., a_r des entiers relatifs ; alors il existe un entier x unique modulo $N = n_1 n_2 \ldots n_r$ tel que $x \equiv a_i \bmod n_i$ pour tout $i = 1, \ldots, r$. Soient $N_i = N/n_i$ et u_i l'inverse de N_i modulo n_i, alors

$$x = \sum_{1}^{r} a_i u_i N_i.$$

On peut voir ce dernier théorème, comme conséquence du théorème suivant :

Théorème

Soient deux entiers n et m premiers entre eux. Alors les anneaux $Z/Zn \times Z/mZ$ et Z/nmZ sont isomorphes.

On désigne par $\mathbf{Z}_n^*$ le groupe des éléments inversibles de l'anneau $\mathbf{Z}/n\mathbf{Z}$.

On désigne par $\mathbf{Z}_n^*$ le groupe des éléments inversibles de l'anneau $\mathbf{Z}/n\mathbf{Z}$. En utilisant le fait que l'ordre d'une classe d'équivalence modulo n dans $\mathbf{Z}_n^*$ divise l'ordre du groupe $\mathbf{Z}_n^*$, on obtient les théorèmes suivants :

On désigne par $\mathbf{Z}_n^*$ le groupe des éléments inversibles de l'anneau $\mathbf{Z}/n\mathbf{Z}$. En utilisant le fait que l'ordre d'une classe d'équivalence modulo n dans $\mathbf{Z}_n^*$ divise l'ordre du groupe $\mathbf{Z}_n^*$, on obtient les théorèmes suivants :

Théorème

Soient p un nombre premier et a un entier positif inférieur strictement à p, alors on a :

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

On désigne par $\mathbf{Z}_n^*$ le groupe des éléments inversibles de l'anneau $\mathbf{Z}/n\mathbf{Z}$. En utilisant le fait que l'ordre d'une classe d'équivalence modulo n dans $\mathbf{Z}_n^*$ divise l'ordre du groupe $\mathbf{Z}_n^*$, on obtient les théorèmes suivants :

Théorème

Soient p un nombre premier et a un entier positif inférieur strictement à p, alors on a :

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

Ce dernier théorème (connu sous le nom du petit théorème de Fermat) a été généralisé, pour un entier n quelconque, en 1760 par Euler :

Théorème

Soient n un entier positif supérieur à 2, a un entier premier avec n et $\phi(n)$ l'ordre du groupe $\mathbf{Z}_n^*$, alors on a :

$$\phi(n) = (p_1 - 1)p_1^{r_1 - 1} \cdots (p_s - 1)p_s^{r_s - 1} \text{ si } n = p_1^{r_1} \cdots p_s^{r_s}$$

$$\text{et } a^{\phi(n)} \equiv 1 \bmod n.$$

Théorème

Soient n un entier positif supérieur à 2, a un entier premier avec n et $\phi(n)$ l'ordre du groupe $\mathbf{Z}_n^*$, alors on a :

$$\phi(n) = (p_1 - 1)p_1^{r_1 - 1} \cdots (p_s - 1)p_s^{r_s - 1} \text{ si } n = p_1^{r_1} \cdots p_s^{r_s}$$

$$\text{et } a^{\phi(n)} \equiv 1 \bmod n.$$

En général, si a et n ne sont pas premiers entre eux on n'a pas toujours $a^{\phi(n)} \equiv 1 \bmod n$.

Théorème

Soient n un entier positif supérieur à 2, a un entier premier avec n et $\phi(n)$ l'ordre du groupe $\mathbf{Z}_n^*$, alors on a :

$$\phi(n) = (p_1 - 1)p_1^{r_1 - 1} \cdots (p_s - 1)p_s^{r_s - 1} \text{ si } n = p_1^{r_1} \cdots p_s^{r_s}$$

$$\text{et } a^{\phi(n)} \equiv 1 \bmod n.$$

En général, si a et n ne sont pas premiers entre eux on n'a pas toujours $a^{\phi(n)} \equiv 1 \bmod n$. Mais on a le résultat très important suivant :

Théorème

Soient n un entier positif supérieur à 2, a un entier premier avec n et $\phi(n)$ l'ordre du groupe $\mathbf{Z}_n^*$, alors on a :

$$\phi(n) = (p_1 - 1)p_1^{r_1 - 1} \cdots (p_s - 1)p_s^{r_s - 1} \text{ si } n = p_1^{r_1} \cdots p_s^{r_s}$$

$$\text{et } a^{\phi(n)} \equiv 1 \bmod n.$$

En général, si a et n ne sont pas premiers entre eux on n'a pas toujours $a^{\phi(n)} \equiv 1 \bmod n$. Mais on a le résultat très important suivant :

Théorème

Soient n un entier sans facteurs carrés et a un entier positif inférieur strictement à n ; alors on a :

$$\forall k \in \mathbf{Z},\ a^{k\phi(n)+1} \equiv a \bmod n$$

Théorème

Soient n un entier positif supérieur à 2, a un entier premier avec n et $\phi(n)$ l'ordre du groupe $\mathbf{Z}_n^*$, alors on a :

$$\phi(n) = (p_1 - 1)p_1^{r_1 - 1} \cdots (p_s - 1)p_s^{r_s - 1} \text{ si } n = p_1^{r_1} \cdots p_s^{r_s}$$

$$\text{et } a^{\phi(n)} \equiv 1 \bmod n.$$

En général, si a et n ne sont pas premiers entre eux on n'a pas toujours $a^{\phi(n)} \equiv 1 \bmod n$. Mais on a le résultat très important suivant :

Théorème

Soient n un entier sans facteurs carrés et a un entier positif inférieur strictement à n ; alors on a :

$$\forall k \in \mathbf{Z},\ a^{k\phi(n)+1} \equiv a \bmod n$$

Exercice. Démontrez le théorème précédent.

Symbole de Legendre (1752 - 1833).

Symbole de Legendre (1752 - 1833).

Soit p un nombre premier impair et a un entier naturel premier avec p.

Symbole de Legendre (1752 - 1833).

Soit p un nombre premier impair et a un entier naturel premier avec p. Le symbole $\left(\frac{}{p}\right)$ est défini par :

Symbole de Legendre (1752 - 1833).

Soit p un nombre premier impair et a un entier naturel premier avec p. Le symbole $(\frac{}{p})$ est défini par :$(\frac{a}{p}) = 1$ si l'équation $x^2 \equiv a \bmod p$ possède une solution dans $\mathbb{N}$.

Symbole de Legendre (1752 - 1833).

Soit p un nombre premier impair et a un entier naturel premier avec p. Le symbole $(\frac{}{p})$ est défini par :$(\frac{a}{p}) = 1$ si l'équation $x^2 \equiv a \bmod p$ possède une solution dans $\mathbb{N}$. Dans le cas contraire on a $(\frac{a}{p}) = -1$.

Symbole de Legendre (1752 - 1833).

Soit p un nombre premier impair et a un entier naturel premier avec p. Le symbole $(\frac{}{p})$ est défini par :$(\frac{a}{p}) = 1$ si l'équation $x^2 \equiv a \bmod p$ possède une solution dans $\mathbb{N}$. Dans le cas contraire on a $(\frac{a}{p}) = -1$. On prolonge le symbole $(\frac{}{p})$ par zéro sur $\mathbb{N}$.

Symbole de Legendre (1752 - 1833).

Soit p un nombre premier impair et a un entier naturel premier avec p. Le symbole $(\frac{}{p})$ est défini par :$(\frac{a}{p}) = 1$ si l'équation $x^2 \equiv a \bmod p$ possède une solution dans $\mathbb{N}$. Dans le cas contraire on a $(\frac{a}{p}) = -1$. On prolonge le symbole $(\frac{}{p})$ par zéro sur $\mathbb{N}$. Ce symbole a été défini par Legendre pour les nombres premiers impairs et par Kronecker (1823 - 1893) pour le nombre 2.

Symbole de Legendre (1752 - 1833).

Soit p un nombre premier impair et a un entier naturel premier avec p. Le symbole $(\frac{}{p})$ est défini par :$(\frac{a}{p}) = 1$ si l'équation $x^2 \equiv a \bmod p$ possède une solution dans $\mathbb{N}$. Dans le cas contraire on a $(\frac{a}{p}) = -1$. On prolonge le symbole $(\frac{}{p})$ par zéro sur $\mathbb{N}$. Ce symbole a été défini par Legendre pour les nombres premiers impairs et par Kronecker (1823 - 1893) pour le nombre 2.

Théorème

Critère d'Euler.

Soit p un nombre premier impair ; alors pour tout entier a premier avec p on a : $a^{(p-1)/2} \equiv \pm 1 \bmod p$.

Les nombres composés qui vérifient le lèmme d'Euler sont les nombres pseudopremiers d'Euler tandis que les nombres composés qui vérifient le théorème de Fermat sont appelés les nombres de Carmichael

Remarques

Remarques

1. Le symbole de Legendre $(\frac{\cdot}{p})$ vérifie de plus :

Remarques

1. Le symbole de Legendre $(\frac{\cdot}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p.$

Remarques

1. Le symbole de Legendre $(\frac{\cdot}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour a et b quelconques dans* $\mathbb{N}$.

Remarques

1. Le symbole de Legendre $(\frac{\cdot}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour a et b quelconques dans* $\mathbb{N}$.

2. Le symbole de Legendre a été généralisé, par Jacobi (1804 - 1851), comme suit :

Remarques

1. Le symbole de Legendre $(\frac{a}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour a et b quelconques dans* $\mathbb{N}$.

2. Le symbole de Legendre a été généralisé, par Jacobi (1804 - 1851), comme suit :

Le symbole de Jacobi $(\frac{a}{n})$ *est défini par :*

Remarques

1. Le symbole de Legendre $(\frac{\cdot}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour a et b quelconques dans* $\mathbb{N}$.

2. Le symbole de Legendre a été généralisé, par Jacobi (1804 - 1851), comme suit :

Le symbole de Jacobi $(\frac{a}{n})$ *est défini par :*

C'est le symbole de Legendre si $n = p$, *un nombre premier impair ;*

Remarques

1. Le symbole de Legendre $(\frac{\cdot}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour a et b quelconques dans* $\mathbb{N}$.

2. Le symbole de Legendre a été généralisé, par Jacobi (1804 - 1851), comme suit :

Le symbole de Jacobi $(\frac{a}{n})$ *est défini par :*

C'est le symbole de Legendre si $n = p$, *un nombre premier impair ;*

c'est le symbole de Kronecker pour $n = 2$ *:*

Remarques

1. Le symbole de Legendre $(\frac{a}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour a et b quelconques dans* $\mathbb{N}$.

2. Le symbole de Legendre a été généralisé, par Jacobi (1804 - 1851), comme suit :

Le symbole de Jacobi $(\frac{a}{n})$ *est défini par :*

C'est le symbole de Legendre si $n = p$, *un nombre premier impair ;*

c'est le symbole de Kronecker pour $n = 2$ *:*

$$(\frac{a}{2}) = \begin{cases} 1 \text{ si } a \equiv 1 \bmod 8, \\ -1 \text{ si } a \equiv 5 \bmod 8, \\ 0 \text{ si } a \text{ est divisible par } 4, \\ \text{et il n'est pas défini pour les autres valeurs de } a. \end{cases}$$

Remarques

1. Le symbole de Legendre $(\frac{\cdot}{p})$ *vérifie de plus :*

i) $(\frac{a}{p}) \equiv a^{(p-1)/2} \bmod p$.

ii) $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour a et b quelconques dans* $\mathbb{N}$.

2. Le symbole de Legendre a été généralisé, par Jacobi (1804 - 1851), comme suit :

Le symbole de Jacobi $(\frac{a}{n})$ *est défini par :*

C'est le symbole de Legendre si $n = p$, *un nombre premier impair ;*

c'est le symbole de Kronecker pour $n = 2$ *:*

$$(\frac{a}{2}) = \begin{cases} 1 \text{ si } a \equiv 1 \bmod 8, \\ -1 \text{ si } a \equiv 5 \bmod 8, \\ 0 \text{ si } a \text{ est divisible par } 4, \\ \text{et il n'est pas défini pour les autres valeurs de } a. \end{cases}$$

Et si $n = \prod_{i=1}^{i=r} p_i^{m_i}$, *alors* $(\frac{a}{n}) = \prod_{i=1}^{i=r} (\frac{a}{p_i})^{m_i}$.

Tests de Primalité

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ;

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers.

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps.

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps. On peut toujours essayer de chercher parmi les nombres inférieurs à ce nombre ses diviseurs,

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps. On peut toujours essayer de chercher parmi les nombres inférieurs à ce nombre ses diviseurs, essayer la méthode du crible d'*Eratosthène*,

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps. On peut toujours essayer de chercher parmi les nombres inférieurs à ce nombre ses diviseurs, essayer la méthode du crible d'*Eratosthène*, le test de Fermat,

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps. On peut toujours essayer de chercher parmi les nombres inférieurs à ce nombre ses diviseurs, essayer la méthode du crible d'*Eratosthène*, le test de Fermat, celui d'Euler

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps. On peut toujours essayer de chercher parmi les nombres inférieurs à ce nombre ses diviseurs, essayer la méthode du crible d'*Eratosthène*, le test de Fermat, celui d'Euler ou bien d'autres.

Tests de Primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de déterminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps. On peut toujours essayer de chercher parmi les nombres inférieurs à ce nombre ses diviseurs, essayer la méthode du crible d'*Eratosthène*, le test de Fermat, celui d'Euler ou bien d'autres. Cependant, à part l'algorithme AKS, la plus part de ces algorithmes restent incapables de déterminer, en un temps raisonnable, la factorisation d'un grand nombre.

Tests de Primalité

Tests de Primalité

Algorithme Probabiliste : c‘est un algorithme qui donne un résultat probable.

Tests de Primalité

Algorithme Probabiliste : c‘est un algorithme qui donne un résultat probable.
Algorithme déterministe : c’est un algorithme qui donne une réponse oui ou non.

Tests de Primalité

Algorithme Probabiliste : c‘est un algorithme qui donne un résultat probable.
Algorithme déterministe : c’est un algorithme qui donne une réponse oui ou non.
Algorithme Inconditionnel : C’est un algorithme qui ne dépend de aucune Condition.

Algorithme Probabiliste : c'est un algorithme qui donne un résultat probable.
Algorithme déterministe : c'est un algorithme qui donne une réponse oui ou non.
Algorithme Inconditionnel : C'est un algorithme qui ne dépend de aucune Condition.
Algorithme polynomial : algorithme nécessitant n opérations et dont le temps $t(n)$, vérifie $t(n) \leq cP(n)$ à partir d'un certain N où $P(x)$ est un pôlynome et c est une constante positif. Dans le cas contraire il peut être exponentiel ou sous-exponentiel.

Algorithme Probabiliste : c'est un algorithme qui donne un résultat probable.
Algorithme déterministe : c'est un algorithme qui donne une réponse oui ou non.
Algorithme Inconditionnel : C'est un algorithme qui ne dépend de aucune Condition.
Algorithme polynomial : algorithme nécessitant n opérations et dont le temps $t(n)$, vérifie $t(n) \leq cP(n)$ à partir d'un certain N où $P(x)$ est un pôlynome et c est une constante positif. Dans le cas contraire il peut être exponentiel ou sous-exponentiel.
Le test de primalité de *Miller-Rabin* est un test de primalité probabiliste : c'est-à-dire un algorithme qui détermine si un nombre donné est probablement premier, de façon similaire au test de primalité de *Fermat*.

Tests de Primalité

Critère de *Fermat* :

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Euler* :

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Euler* : Ce critère, repose sur le Critère d'Euler. On prend un entier a au hasard, et on calcule $a^{\frac{n-1}{2}} \, mod \, n$; si $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \, [mod \, n]$ alors n n'et pas premier.

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Euler* : Ce critère, repose sur le Critère d'Euler. On prend un entier a au hasard, et on calcule $a^{\frac{n-1}{2}} \, mod \, n$; si $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \, [mod \, n]$ alors n n'et pas premier.

Critère de *Miller-Rabin* :

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Euler* : Ce critère, repose sur le Critère d'Euler. On prend un entier a au hasard, et on calcule $a^{\frac{n-1}{2}} \, mod \, n$; si $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \, [mod \, n]$ alors n n'et pas premier.

Critère de *Miller-Rabin* :

Propriété :

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Euler* : Ce critère, repose sur le Critère d'Euler. On prend un entier a au hasard, et on calcule $a^{\frac{n-1}{2}} \, mod \, n$; si $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \, [mod \, n]$ alors n n'et pas premier.

Critère de *Miller-Rabin* :

Propriété :

Soit $p > 2$, un nombre premier. Écrivons $p - 1 = 2^s.t$ avec t impair.

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Euler* : Ce critère, repose sur le Critère d'Euler. On prend un entier a au hasard, et on calcule $a^{\frac{n-1}{2}} \, mod \, n$; si $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \, [mod \, n]$ alors n n'et pas premier.

Critère de *Miller-Rabin* :

Propriété :

Soit $p > 2$, un nombre premier. Écrivons $p - 1 = 2^s.t$ avec t impair.

Soit a un entier non divisible par p.

Critère de *Fermat* : Ce critère, repose sur le petit théorème de Fermat. On prend un entier a au hasard, et on calcule $a^{n-1} \, mod \, n$; si $a^{n-1} \not\equiv 1 \, [mod \, n]$ alors n n'et pas premier.

Critère de *Euler* : Ce critère, repose sur le Critère d'Euler. On prend un entier a au hasard, et on calcule $a^{\frac{n-1}{2}} \, mod \, n$; si $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \, [mod \, n]$ alors n n'et pas premier.

Critère de *Miller-Rabin* :

Propriété :

Soit $p > 2$, un nombre premier. Écrivons $p - 1 = 2^s.t$ avec t impair. Soit a un entier non divisible par p. Alors ou bien $a^t \equiv 1 [mod \, p]$, ou bien il existe un entier i tel que $i < s$ et $a^{2^i \cdot t} \equiv -1 [mod \, n]$.

Tests de Primalité

Corrolaire

Soit $n < 1$ un entier impair. Écrivons $n-1 = 2^s.t$ avec t impair. Supposons qu'il existe un entier a avec $1 < a < n$, $a^t \not\equiv 1[mod\ n]$ et $a^{2^i.t} \not\equiv -1[mod\ n]$ pour $i = 0, 1, ..., s-1$. Alors n est composé. (Appelons un tel entier a un témoin de Miller).

Corrolaire

Soit $n < 1$ un entier impair. Écrivons $n-1 = 2^s.t$ avec t impair. Supposons qu'il existe un entier a avec $1 < a < n$, $a^t \not\equiv 1[mod\ n]$ et $a^{2^i.t} \not\equiv -1[mod\ n]$ pour $i = 0, 1, \ldots, s-1$. Alors n est composé. (Appelons un tel entier a un témoin de Miller).

Propriété : Si le nombre impair n est composé, au moins les trois quarts des $n-2$ entiers a tels que $1 < a < n$ sont des témoins de *Miller* pour n.

Corrolaire

Soit $n < 1$ un entier impair. Écrivons $n-1 = 2^s.t$ avec t impair. Supposons qu'il existe un entier a avec $1 < a < n$, $a^t \not\equiv 1[mod\ n]$ et $a^{2^i.t} \not\equiv -1[mod\ n]$ pour $i = 0, 1, \ldots, s-1$. Alors n est composé. (Appelons un tel entier a un témoin de Miller).

Propriété : Si le nombre impair n est composé, au moins les trois quarts des $n-2$ entiers a tels que $1 < a < n$ sont des témoins de *Miller* pour n.

Théorème

Théorème(Rabin) Soit n un entier impair composé tel que $n > 9$. Posons $n-1 = 2^s.t$ avec t impair.

Corrolaire

Soit $n < 1$ un entier impair. Écrivons $n-1 = 2^s.t$ avec t impair. Supposons qu'il existe un entier a avec $1 < a < n$, $a^t \not\equiv 1[mod\ n]$ et $a^{2^i.t} \not\equiv -1[mod\ n]$ pour $i = 0, 1, ..., s-1$. Alors n est composé. (Appelons un tel entier a un témoin de Miller).

Propriété : Si le nombre impair n est composé, au moins les trois quarts des $n-2$ entiers a tels que $1 < a < n$ sont des témoins de *Miller* pour n.

Théorème

Théorème(Rabin) Soit n un entier impair composé tel que $n > 9$. Posons $n-1 = 2^s.t$ avec t impair. Les entiers a compris entre 1 et n et qui satisfont à la condition $a^t \equiv 1[mod\ n]$ ou à l'une des conditions $a^{2^i.t} \equiv -1[mod\ n]$ pour $i = 0, 1, ..., s-1$ sont en nombre au plus $\varphi(n)/4$ (avec $\varphi(n)$ l'indicateur d'Euler).

Tests de Primalité de Rabin-Miller

Tests de Primalité de Rabin-Miller

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :

Tests de Primalité de Rabin-Miller

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :

(1) Non, il est composé,

Tests de Primalité de Rabin-Miller

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :

(1) Non, il est composé,

(2) La probabilité qu'il soit premier est x.

Tests de Primalité de Rabin-Miller

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :
(1) Non, il est composé,
(2) La probabilité qu'il soit premier est x.
On écrit $n-1$ sous la forme $n-1=2^s t$, où t est impair. On choisit au hasard un entier b dans l'intervalle $[1, n-1]$ et on calcule les résidus dans $[0, n-1]$ des puissances suivantes de b modulo n :

Tests de Primalité de Rabin-Miller

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :
(1) Non, il est composé,
(2) La probabilité qu'il soit premier est x.
On écrit $n-1$ sous la forme $n-1=2^s t$, où t est impair. On choisit au hasard un entier b dans l'intervalle $[1, n-1]$ et on calcule les résidus dans $[0, n-1]$ des puissances suivantes de b modulo n :
$(S)\ \ b^t (mod\ n),\quad b^{2t} (mod\ n),\quad .....,\quad b^{2^{s-1}t} (mod\ n),\quad b^{n-1} (mod\ n).$

Tests de Primalité de Rabin-Miller

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :
(1) Non, il est composé,
(2) La probabilité qu'il soit premier est x.
On écrit $n-1$ sous la forme $n-1=2^s t$, où t est impair. On choisit au hasard un entier b dans l'intervalle $[1, n-1]$ et on calcule les résidus dans $[0, n-1]$ des puissances suivantes de b modulo n :
$(S)\quad b^t (mod\ n),\quad b^{2t} (mod\ n),\quad .....,\quad b^{2^{s-1}t} (mod\ n),\quad b^{n-1} (mod\ n).$

Définition

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :
(1) Non, il est composé,
(2) La probabilité qu'il soit premier est x.
On écrit $n-1$ sous la forme $n-1=2^s t$, où t est impair. On choisit au hasard un entier b dans l'intervalle $[1, n-1]$ et on calcule les résidus dans $[0, n-1]$ des puissances suivantes de b modulo n :
$(S)\ \ b^t (mod\ n),\quad b^{2t} (mod\ n),\quad .....,\quad b^{2^{s-1}t} (mod\ n),\quad b^{n-1} (mod\ n).$

Définition

On dit que n passe le test de primalité de Rabin-Miller en base b si les deux résultats suivants sont vérifiés :

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :
(1) Non, il est composé,
(2) La probabilité qu'il soit premier est x.
On écrit $n-1$ sous la forme $n-1=2^s t$, où t est impair. On choisit au hasard un entier b dans l'intervalle $[1, n-1]$ et on calcule les résidus dans $[0, n-1]$ des puissances suivantes de b modulo n :
$(S)\ \ b^t (mod\ n),\quad b^{2t} (mod\ n),\quad .....,\quad b^{2^{s-1}t} (mod\ n),\quad b^{n-1} (mod\ n).$

Définition

On dit que n passe le test de primalité de Rabin-Miller en base b si les deux résultats suivants sont vérifiés :
(i) $b^{n-1} \equiv 1 [mod\ n]$,

Tests de Primalité de Rabin-Miller

Soit n un entier impair donné. A la question : n est-il premier, le test de *Rabin-Miller* donne l'une des réponses suivantes :
(1) Non, il est composé,
(2) La probabilité qu'il soit premier est x.
On écrit $n-1$ sous la forme $n-1=2^s t$, où t est impair. On choisit au hasard un entier b dans l'intervalle $[1, n-1]$ et on calcule les résidus dans $[0, n-1]$ des puissances suivantes de b modulo n :
$(S)\ \ b^t(mod\ n),\quad b^{2t}(mod\ n),\quad .....,\quad b^{2^{s-1}t}(mod\ n),\quad b^{n-1}(mod\ n).$

Définition

On dit que n passe le test de primalité de Rabin-Miller en base b si les deux résultats suivants sont vérifiés :
(i) $b^{n-1} \equiv 1[mod\ n]$,
(ii) Si le premier élément de (S) n'est pas égale a 1, et $b^{2^r t}$ *est le premier élément égale a 1, alors l'élément précédent* $b^{2^{r-1}t}(mod\ n)$ *est* $n-1$.

Tests de Primalité

Remarque

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller.

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller. C'est essentiellement le théorème de Fermat dans le corps $(\mathbb{Z}/n\mathbb{Z})$.

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller. C'est essentiellement le théorème de Fermat dans le corps $(\mathbb{Z}/n\mathbb{Z})$.
(2) Si le résultat (i) n'est pas vérifié, alors le théorème de Fermat n'est pas vrai dans $(\mathbb{Z}/n\mathbb{Z})^*$, donc n n'est pas premier.

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller. C'est essentiellement le théorème de Fermat dans le corps $(\mathbb{Z}/n\mathbb{Z})$.
(2) Si le résultat (i) n'est pas vérifié, alors le théorème de Fermat n'est pas vrai dans $(\mathbb{Z}/n\mathbb{Z})^*$, *donc n n'est pas premier.*
(3) Si le résultat (ii) n'est pas vérifié, c'est qu'il existe dans l'anneau $(\mathbb{Z}/n\mathbb{Z})$ *un élément u tel que* $u \neq \pm 1$ *et* $u^2 = 1$, *ce qui n'est pas possible dans un corps, donc n n'est pas premier.*

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller. C'est essentiellement le théorème de Fermat dans le corps $(\mathbb{Z}/n\mathbb{Z})$.
(2) Si le résultat (i) n'est pas vérifié, alors le théorème de Fermat n'est pas vrai dans $(\mathbb{Z}/n\mathbb{Z})^*$, *donc n n'est pas premier.*
(3) Si le résultat (ii) n'est pas vérifié, c'est qu'il existe dans l'anneau $(\mathbb{Z}/n\mathbb{Z})$ *un élément u tel que* $u \neq \pm 1$ *et* $u^2 = 1$, *ce qui n'est pas possible dans un corps, donc n n'est pas premier.*
(4) Si n passe le test dans une base b, alors l'entier n est premier avec une probabilité supérieure à $\frac{3}{4}$.

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller. C'est essentiellement le théorème de Fermat dans le corps $(\mathbb{Z}/n\mathbb{Z})$.
(2) Si le résultat (i) n'est pas vérifié, alors le théorème de Fermat n'est pas vrai dans $(\mathbb{Z}/n\mathbb{Z})^*$, *donc n n'est pas premier.*
(3) Si le résultat (ii) n'est pas vérifié, c'est qu'il existe dans l'anneau $(\mathbb{Z}/n\mathbb{Z})$ *un élément u tel que* $u \neq \pm 1$ *et* $u^2 = 1$, *ce qui n'est pas possible dans un corps, donc n n'est pas premier.*
(4) Si n passe le test dans une base b, alors l'entier n est premier avec une probabilité supérieure à $\frac{3}{4}$.

Test de primalité de *lehmer* :

Grâce au test de Fermat, d'autres variétés de test ont été mises au point.

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller. C'est essentiellement le théorème de Fermat dans le corps $(\mathbb{Z}/n\mathbb{Z})$.
(2) Si le résultat (i) n'est pas vérifié, alors le théorème de Fermat n'est pas vrai dans $(\mathbb{Z}/n\mathbb{Z})^*$, *donc n n'est pas premier.*
(3) Si le résultat (ii) n'est pas vérifié, c'est qu'il existe dans l'anneau $(\mathbb{Z}/n\mathbb{Z})$ *un élément u tel que* $u \neq \pm 1$ *et* $u^2 = 1$, *ce qui n'est pas possible dans un corps, donc n n'est pas premier.*
(4) Si n passe le test dans une base b, alors l'entier n est premier avec une probabilité supérieure à $\frac{3}{4}$.

Test de primalité de *lehmer* :

Grâce au test de Fermat, d'autres variétés de test ont été mises au point. Dans ce test on suppose donnée une décomposition en facteurs premiers de $p-1$.

Remarque

(1) Si n est premier, alors n passe le test de Rabin-Miller. C'est essentiellement le théorème de Fermat dans le corps $(\mathbb{Z}/n\mathbb{Z})$.
(2) Si le résultat (i) n'est pas vérifié, alors le théorème de Fermat n'est pas vrai dans $(\mathbb{Z}/n\mathbb{Z})^*$, *donc n n'est pas premier.*
(3) Si le résultat (ii) n'est pas vérifié, c'est qu'il existe dans l'anneau $(\mathbb{Z}/n\mathbb{Z})$ *un élément u tel que* $u \neq \pm 1$ *et* $u^2 = 1$, *ce qui n'est pas possible dans un corps, donc n n'est pas premier.*
(4) Si n passe le test dans une base b, alors l'entier n est premier avec une probabilité supérieure à $\frac{3}{4}$.

Test de primalité de *lehmer* :

Grâce au test de Fermat, d'autres variétés de test ont été mises au point. Dans ce test on suppose donnée une décomposition en facteurs premiers de $p-1$.

Tests de Primalité

Théorème

(Critère de lehmer)

Théorème

(Critère de lehmer) Soit $n > 1$ un enter impair tel qu'on connaît tous les facteurs premiers de $n - 1$.

Théorème

(Critère de lehmer) Soit $n > 1$ un enter impair tel qu'on connaît tous les facteurs premiers de $n-1$. Les conditions suivantes sont équivalentes :

Théorème

(Critère de lehmer) Soit $n > 1$ un enter impair tel qu'on connaît tous les facteurs premiers de $n-1$. Les conditions suivantes sont équivalentes :
(i) n est premier.

Théorème

(Critère de lehmer) Soit $n > 1$ un enter impair tel qu'on connaît tous les facteurs premiers de $n-1$. Les conditions suivantes sont équivalentes :
(i) n est premier.
(ii) Il existe un entier a tel que $a^{n-1} \equiv 1[mod\ n]$ et $a^{(n-1)/q} \not\equiv 1[mod\ n]$ pour tout facteur premier q de $n-1$.

Exercice. Démontrez le théorème précédent.

Théorème

(Critère de lehmer) Soit $n > 1$ un enter impair tel qu'on connaît tous les facteurs premiers de $n-1$. Les conditions suivantes sont équivalentes :
(i) n est premier.
(ii) Il existe un entier a tel que $a^{n-1} \equiv 1[mod\ n]$ et $a^{(n-1)/q} \not\equiv 1[mod\ n]$ pour tout facteur premier q de $n-1$.

Exercice. Démontrez le théorème précédent.

Corrolaire

Soit $n > 2$ un entier impair. Les conditions suivantes sont équivalentes :

Théorème

(Critère de lehmer) Soit $n > 1$ un enter impair tel qu'on connaît tous les facteurs premiers de $n-1$. Les conditions suivantes sont équivalentes :
(i) n est premier.
(ii) Il existe un entier a tel que $a^{n-1} \equiv 1[mod\ n]$ et $a^{(n-1)/q} \not\equiv 1[mod\ n]$ pour tout facteur premier q de $n-1$.

Exercice. Démontrez le théorème précédent.

Corrolaire

Soit $n > 2$ un entier impair. Les conditions suivantes sont équivalentes :
(i) n est premier.

Théorème

(Critère de lehmer) Soit $n > 1$ un enter impair tel qu'on connaît tous les facteurs premiers de $n-1$. Les conditions suivantes sont équivalentes :
(i) n est premier.
(ii) Il existe un entier a tel que $a^{n-1} \equiv 1[mod\ n]$ et $a^{(n-1)/q} \not\equiv 1[mod\ n]$ pour tout facteur premier q de $n-1$.

Exercice. Démontrez le théorème précédent.

Corrolaire

Soit $n > 2$ un entier impair. Les conditions suivantes sont équivalentes :
(i) n est premier.
(ii) il existe un entier a tel que $a^{(n-1)/2} \equiv -1[mod\ n]$ et $a^{(n-1)/q} \not\equiv 1[mod\ n]$ pour tout facteur premier impair q de $n-1$.

Tests de Primalité

Les nombres de Fermat

Les nombres de Fermat

Pour les nombres de *Fermat* $F_n = 2^{2^n} + 1$, *le critère de Lehmer devient :*

Les nombres de Fermat

Pour les nombres de *Fermat* $F_n = 2^{2^n} + 1$, *le critère de Lehmer devient :*

Lemme

Pour que F_n *soit premier, il faut et il suffit qu'il existe a avec*

$$a^{(F_n-1)/2} \equiv -1 [mod\ F_n].$$

Les nombres de Fermat

Pour les nombres de *Fermat* $F_n = 2^{2^n} + 1$, *le critère de Lehmer devient :*

Lemme

Pour que F_n *soit premier, il faut et il suffit qu'il existe a avec*

$$a^{(F_n-1)/2} \equiv -1 [mod\ F_n].$$

On a aussi un test de Péepin qui date de 1877.

Les nombres de Fermat

Pour les nombres de *Fermat* $F_n = 2^{2^n} + 1$, *le critère de Lehmer devient :*

Lemme

Pour que F_n *soit premier, il faut et il suffit qu'il existe a avec*

$$a^{(F_n-1)/2} \equiv -1 [mod\ F_n].$$

On a aussi un test de Péepin qui date de 1877.

Théorème

Pour $n \geq 1$, *on a :* F_n *est premier* $\Leftrightarrow 3^{(F_n-1)/2} \equiv -1 (mod\ F_n)$.

Tests de Primalité

Les nombres de Mersenne

Les nombres de Mersenne

Théorème

(Théorème de Lucas-Lehmer sur les nombres de Mersenne*)*
Soient s un nombre premier impair, $n = 2^s - 1$*, a un entier tel que n soit premier avec* $a^2 - 4$*.*
On définit par récurrence une suite d'entiers, (L_i) *où* $i \geq 1$*, dite suite majeure de Lucas, comme suit :* $L_1 = a$*,* $L_{i+1} = {L_i}^2 - 2$*. Alors on a :* $L_{s-1} \equiv 0 (mod\ n) \Leftrightarrow$ *n est premier.*

Tests de Primalité

L'algorithme AKS

En 2002, Manindra Agrawal de l'Indian Institute of Technology à Kanpur et deux de ses étudiants Neeraj Kayal et Nitin Saxena ont trouvé un algorithme simple et polynômial qui teste la primalité de n.

L'algorithme AKS

En 2002, Manindra Agrawal de l'Indian Institute of Technology à Kanpur et deux de ses étudiants Neeraj Kayal et Nitin Saxena ont trouvé un algorithme simple et polynômial qui teste la primalité de n. Ce dernier algorithme se base sur le petit théeorème de Fermat et repose sur le résultat suivant,

L'algorithme AKS

En 2002, Manindra Agrawal de l'Indian Institute of Technology à Kanpur et deux de ses étudiants Neeraj Kayal et Nitin Saxena ont trouvé un algorithme simple et polynômial qui teste la primalité de n. Ce dernier algorithme se base sur le petit théeorème de Fermat et repose sur le résultat suivant, qui est une généralisation du petit théorème de Fermat.

L'algorithme AKS

En 2002, Manindra Agrawal de l'Indian Institute of Technology à Kanpur et deux de ses étudiants Neeraj Kayal et Nitin Saxena ont trouvé un algorithme simple et polynômial qui teste la primalité de n. Ce dernier algorithme se base sur le petit théeorème de Fermat et repose sur le résultat suivant, qui est une généralisation du petit théorème de Fermat.

Théorème

Soient n un entier naturel strictement supérieur à 2 et a un entier relatif premier avec n. Alors n est premier si, et seulement si,

$$(X+a)^n \equiv X^n + a \bmod n.$$

L'algorithme AKS

En 2002, Manindra Agrawal de l'Indian Institute of Technology à Kanpur et deux de ses étudiants Neeraj Kayal et Nitin Saxena ont trouvé un algorithme simple et polynômial qui teste la primalité de n. Ce dernier algorithme se base sur le petit théeorème de Fermat et repose sur le résultat suivant, qui est une généralisation du petit théorème de Fermat.

Théorème

Soient n un entier naturel strictement supérieur à 2 et a un entier relatif premier avec n. Alors n est premier si, et seulement si,

$$(X+a)^n \equiv X^n + a \bmod n.$$

Pour appliquer ce Théorème sur un entier n, il suffit de choisir un entier a premier avec n et ensuite voir si la congruence est satisfaite.

Tests de Primalité

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.
Soit n un entier supérieur ou égal à 2.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.
Soit n un entier supérieur ou égal à 2.
1. Si $n = a^b$ pour a et b deux entiers tels que $b > 1$, alors n est Composé.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.
Soit n un entier supérieur ou égal à 2.
1. Si $n = a^b$ pour a et b deux entiers tels que $b > 1$, alors n est Composé.
2. Déterminer le plus petit entier r tel que l'ordre de n dans Z/rZ soit supérieur à $4log(n)^2$.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.
Soit n un entier supérieur ou égal à 2.
1. Si $n = a^b$ pour a et b deux entiers tels que $b > 1$, alors n est Composé.
2. Déterminer le plus petit entier r tel que l'ordre de n dans Z/rZ soit supérieur à $4log(n)^2$.
3. Si $1 < pdcd(a, n) < n$ pour un entier $a \leq r$, alors n est Composé.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.
Soit n un entier supérieur ou égal à 2.
1. Si $n = a^b$ pour a et b deux entiers tels que $b > 1$, alors n est Composé.
2. Déterminer le plus petit entier r tel que l'ordre de n dans Z/rZ soit supérieur à $4log(n)^2$.
3. Si $1 < pdcd(a, n) < n$ pour un entier $a \leq r$, alors n est Composé.
4. Si $n \leq r$, alors n est Premier.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.
Soit n un entier supérieur ou égal à 2.
1. Si $n = a^b$ pour a et b deux entiers tels que $b > 1$, alors n est Composé.
2. Déterminer le plus petit entier r tel que l'ordre de n dans Z/rZ soit supérieur à $4log(n)^2$.
3. Si $1 < pdcd(a, n) < n$ pour un entier $a \leq r$, alors n est Composé.
4. Si $n \leq r$, alors n est Premier.
5. Pour $a = 1$ à $\lfloor 2\sqrt{\phi(r)}log(n) \rfloor$: si $(X + a)^n \not\equiv X^n + a$ dans $Z/nZ[X]/(X^r - 1)Z/nZ[X]$, alors n est Composé.

Pour gagner sur le temps de calcul l'idée c'est d'évaluer les deux membres de la congruence modulo un polynôme de la forme $X^r - 1$ pour un certain entier r plus petit que n.
Ainsi l'algorithme de primalité AKS, qui est déterministe, polynômial et inconditionnel ; est le suivant.
Soit n un entier supérieur ou égal à 2.
1. Si $n = a^b$ pour a et b deux entiers tels que $b > 1$, alors n est Composé.
2. Déterminer le plus petit entier r tel que l'ordre de n dans Z/rZ soit supérieur à $4log(n)^2$.
3. Si $1 < pdcd(a, n) < n$ pour un entier $a \leq r$, alors n est Composé.
4. Si $n \leq r$, alors n est Premier.
5. Pour $a = 1$ à $\lfloor 2\sqrt{\phi(r)}log(n) \rfloor$: si $(X + a)^n \not\equiv X^n + a$ dans $Z/nZ[X]/(X^r - 1)Z/nZ[X]$, alors n est Composé.
6. Autrement n est Premier.

Généralités sur la Cryptographie

Cryptographie : Art et Science de l'étude des systèmes propres à résoudre les problèmes de confidentialité et d'authentification.

Cryptographie : Art et Science de l'étude des systèmes propres à résoudre les problèmes de confidentialité et d'authentification.
Cryptanalyse : Science permettant d'étudier les systèmes cryptographiques en vue de les tester ou de les casser.

Cryptographie : Art et Science de l'étude des systèmes propres à résoudre les problèmes de confidentialité et d'authentification.
Cryptanalyse : Science permettant d'étudier les systèmes cryptographiques en vue de les tester ou de les casser.
Cryptologie : Branche qui traite de la cryptographie et de la cryptanalyse.

Cryptographie : Art et Science de l'étude des systèmes propres à résoudre les problèmes de confidentialité et d'authentification.
Cryptanalyse : Science permettant d'étudier les systèmes cryptographiques en vue de les tester ou de les casser.
Cryptologie : Branche qui traite de la cryptographie et de la cryptanalyse.
Chiffrement d'un texte clair : transformation du texte clair en un texte incompréhensif.

Cryptographie : Art et Science de l'étude des systèmes propres à résoudre les problèmes de confidentialité et d'authentification.
Cryptanalyse : Science permettant d'étudier les systèmes cryptographiques en vue de les tester ou de les casser.
Cryptologie : Branche qui traite de la cryptographie et de la cryptanalyse.
Chiffrement d'un texte clair : transformation du texte clair en un texte incompréhensif.
Déchiffrement : l'opération inverse du chiffrement.

Types d'algorithmes et leurs sécurités

Types d'algorithmes et leurs sécurités

Deux types d'Algorithmes :

Types d'algorithmes et leurs sécurités

Deux types d'Algorithmes :
- Algorithmes à clé secrète (crypto-systèmes symétriques) : On a une clé K1 pour chiffrer et une clé K2 pour déchiffrer ; K1 peut être calculer à partir de K2 et vice versa. On a souvent K1=K2 ; les clés K1 et K2 doivent être secrètes. Exemples : Substitutions Affines, DES, AES...

Types d'algorithmes et leurs sécurités

Deux types d'Algorithmes :
- Algorithmes à clé secrète (crypto-systèmes symétriques) : On a une clé K1 pour chiffrer et une clé K2 pour déchiffrer ; K1 peut être calculer à partir de K2 et vice versa. On a souvent K1=K2 ; les clés K1 et K2 doivent être secrètes. Exemples : Substitutions Affines, DES, AES...
- Algorithmes à clé publique(crypto-systèmes asymétriques) Deux clés K1 et K2 ; K2 ne peut pas être calculer à partir de K1. La clé K1 peut être publique et la clé K2 doit être secrète (clef privée). Exemples : RSA, ElGamal...

Types d'algorithmes et leurs sécurités

Deux types d'Algorithmes :
- Algorithmes à clé secrète (crypto-systèmes symétriques) : On a une clé K1 pour chiffrer et une clé K2 pour déchiffrer ; K1 peut être calculer à partir de K2 et vice versa. On a souvent K1=K2 ; les clés K1 et K2 doivent être secrètes. Exemples : Substitutions Affines, DES, AES...
- Algorithmes à clé publique(crypto-systèmes asymétriques) Deux clés K1 et K2 ; K2 ne peut pas être calculer à partir de K1. La clé K1 peut être publique et la clé K2 doit être secrète (clef privée). Exemples : RSA, ElGamal...

La sécurité des Algorithmes repose en général sur certains problèmes difficiles à résoudre.

Types d'algorithmes et leurs sécurités

Deux types d'Algorithmes :
- Algorithmes à clé secrète (crypto-systèmes symétriques) : On a une clé K1 pour chiffrer et une clé K2 pour déchiffrer ; K1 peut être calculer à partir de K2 et vice versa. On a souvent K1=K2 ; les clés K1 et K2 doivent être secrètes. Exemples : Substitutions Affines, DES, AES...
- Algorithmes à clé publique(crypto-systèmes asymétriques) Deux clés K1 et K2 ; K2 ne peut pas être calculer à partir de K1. La clé K1 peut être publique et la clé K2 doit être secrète (clef privée). Exemples : RSA, ElGamal...

La sécurité des Algorithmes repose en général sur certains problèmes difficiles à résoudre. En particulier, le problème de factoriser un grand nombre dans un temps convenable et le problème de calculer le logarithme discret d'un élément d'un groupe dans un temps convenable font partie de ces problèmes difficiles.

RSA (Rivest, Shamir et Adleman 1978)

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :

i. $n_X = pq$ où p et q sont des nombres premiers,

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :

i. $n_X = pq$ où p et q sont des nombres premiers,

ii. p et q sont gardés secrets par chacun,

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :

i. $n_X = pq$ où p et q sont des nombres premiers,

ii. p et q sont gardés secrets par chacun,

iii. l'entier s_X est premier avec l'entier $(p-1)(q-1)$. Par suite il existe t_X tel que

$$1 \leq t_X < (p-1)(q-1) \text{ et } s_X t_X \equiv 1 \bmod (p-1)(q-1).$$

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :

i. $n_X = pq$ où p et q sont des nombres premiers,

ii. p et q sont gardés secrets par chacun,

iii. l'entier s_X est premier avec l'entier $(p-1)(q-1)$. Par suite il existe t_X tel que

$$1 \leq t_X < (p-1)(q-1) \text{ et } s_X t_X \equiv 1 \bmod (p-1)(q-1).$$

On publie la clé publique (s_X, n_X).

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :
i. $n_X = pq$ où p et q sont des nombres premiers,
ii. p et q sont gardés secrets par chacun,
iii. l'entier s_X est premier avec l'entier $(p-1)(q-1)$. Par suite il existe t_X tel que

$$1 \leq t_X < (p-1)(q-1) \text{ et } s_X t_X \equiv 1 \bmod (p-1)(q-1).$$

On publie la clé publique (s_X, n_X).
Chaque personne garde secrètement la clé privée (t_X, n_X).

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :
i. $n_X = pq$ où p et q sont des nombres premiers,
ii. p et q sont gardés secrets par chacun,
iii. l'entier s_X est premier avec l'entier $(p-1)(q-1)$. Par suite il existe t_X tel que

$$1 \leq t_X < (p-1)(q-1) \text{ et } s_X t_X \equiv 1 \bmod (p-1)(q-1).$$

On publie la clé publique (s_X, n_X).
Chaque personne garde secrètement la clé privée (t_X, n_X).
Fonction de chiffrement : $E(M) = M^{s_X} = C \bmod n$.

RSA (Rivest, Shamir et Adleman 1978)

Soit une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers n_X et s_X vérifiant les conditions suivantes :
i. $n_X = pq$ où p et q sont des nombres premiers,
ii. p et q sont gardés secrets par chacun,
iii. l'entier s_X est premier avec l'entier $(p-1)(q-1)$. Par suite il existe t_X tel que

$$1 \leq t_X < (p-1)(q-1) \text{ et } s_X t_X \equiv 1 \bmod (p-1)(q-1).$$

On publie la clé publique (s_X, n_X).
Chaque personne garde secrètement la clé privée (t_X, n_X).
Fonction de chiffrement : $E(M) = M^{s_X} = C \bmod n$.
Fonction de déchiffrement : $D(C) = C^{t_X} = M \bmod n$.

Cryptosystème RSA : Remarques

Remarques

1. Les nombres premiers p et q doivent être bien choisies : ils doivent être très grands, et leurs différence doit être grande car sinon ils seront proches de $\sqrt{n_X}$ et donc faciles à trouver.

Remarques

1. Les nombres premiers p et q doivent être bien choisies : ils doivent être très grands, et leurs différence doit être grande car sinon ils seront proches de $\sqrt{n_X}$ et donc faciles à trouver.

2. **Exercice.** *Soient A et B deux personnes qui veulent s'échanger des messages à l'aide de la méthode RSA et E un espion. On désigne par (s_x, n_x) la clè publique d'une personne X et t_x sa clé privée :*

Remarques

1. Les nombres premiers p et q doivent être bien choisies : ils doivent être très grands, et leurs différence doit être grande car sinon ils seront proches de $\sqrt{n_X}$ *et donc faciles à trouver.*

2. **Exercice.** *Soient A et B deux personnes qui veulent s'échanger des messages à l'aide de la méthode RSA et E un espion. On désigne par* (s_x, n_x) *la clè publique d'une personne X et* t_x *sa clé privée :*

i. On suppose que E a pu trouver $\phi(n_a)$*, vérifiez qu'il peut déterminer la décomposition de* n_a*.*

Remarques

1. Les nombres premiers p et q doivent être bien choisies : ils doivent être très grands, et leurs différence doit être grande car sinon ils seront proches de $\sqrt{n_X}$ et donc faciles à trouver.

2. **Exercice.** *Soient A et B deux personnes qui veulent s'échanger des messages à l'aide de la méthode RSA et E un espion. On désigne par (s_x, n_x) la clè publique d'une personne X et t_x sa clé privée :*

i. On suppose que E a pu trouver $\phi(n_a)$, vérifiez qu'il peut déterminer la décomposition de n_a.

ii. On suppose que E a pu remarquer qu'un message m a été envoyé à A et à B et que $n_a = n_b$ et e_a et e_b sont premiers entre eux ; vérifiez que E peut trouver le message m.

Remarques

1. Les nombres premiers p et q doivent être bien choisies : ils doivent être très grands, et leurs différence doit être grande car sinon ils seront proches de $\sqrt{n_X}$ et donc faciles à trouver.

2. **Exercice.** *Soient A et B deux personnes qui veulent s'échanger des messages à l'aide de la méthode RSA et E un espion. On désigne par (s_x, n_x) la clè publique d'une personne X et t_x sa clé privée :*

i. On suppose que E a pu trouver $\phi(n_a)$, vérifiez qu'il peut déterminer la décomposition de n_a.

ii. On suppose que E a pu remarquer qu'un message m a été envoyé à A et à B et que $n_a = n_b$ et e_a et e_b sont premiers entre eux ; vérifiez que E peut trouver le message m.

iii. On suppose cette fois ci qu'une société a envoyé un message m aux trois personnes A, B et C et que l'espion E s'aperçoit que $e_a = e_b = e_c = 3$ et que n_a, n_b, et n_c sont premiers entre eux deux à deux. On suppose que $m < n_x$ pour $x = a$, b et c ; montrer comment E

Logarithme discret

Logarithme discret

Soient G un groupe cyclique fini dont la loi de composition est notée multiplicativement et g un générateur de G.

Logarithme discret

Soient G un groupe cyclique fini dont la loi de composition est notée multiplicativement et g un générateur de G. Si l'ordre de G est égal à n et e est l'élément neutre de G ; alors $g^n = e$ et

$$G = \{e, g, g^2, g^3, ..., g^{n-1}\}.$$

Logarithme discret

Soient G un groupe cyclique fini dont la loi de composition est notée multiplicativement et g un générateur de G. Si l'ordre de G est égal à n et e est l'élément neutre de G ; alors $g^n = e$ et

$$G = \{e, g, g^2, g^3, ..., g^{n-1}\}.$$

Ainsi, pour tout élément h de G, il éxiste un entier naturel $m < n$ tel que $h = g^m$.

Logarithme discret

Soient G un groupe cyclique fini dont la loi de composition est notée multiplicativement et g un générateur de G. Si l'ordre de G est égal à n et e est l'élément neutre de G ; alors $g^n = e$ et

$$G = \{e, g, g^2, g^3, ..., g^{n-1}\}.$$

Ainsi, pour tout élément h de G, il éxiste un entier naturel $m < n$ tel que $h = g^m$. L'entier m est appelé le logarithme discret de h relativement à la base g et on note alors

$$dlog_g(h) = m.$$

Logarithme discret

Soient G un groupe cyclique fini dont la loi de composition est notée multiplicativement et g un générateur de G. Si l'ordre de G est égal à n et e est l'élément neutre de G ; alors $g^n = e$ et

$$G = \{e, g, g^2, g^3, ..., g^{n-1}\}.$$

Ainsi, pour tout élément h de G, il éxiste un entier naturel $m < n$ tel que $h = g^m$. L'entier m est appelé le logarithme discret de h relativement à la base g et on note alors

$$dlog_g(h) = m.$$

Il n'est pas facile en général de trouver le logarithme d'un élément quelconque de G. Cette dificulté de résoudre ce problème dans certains groupes G est utilisée en cryptolographie pour chiffrer des messages.

Logarithme discret : Remarque

Logarithme discret : Remarque

Remarque

Le groupe $G = (Z/pZ)$ muni de la somme est un groupe où le problème du logarithme discret est facile à résoudre

Logarithme discret : Remarque

Remarque

Le groupe $G = (Z/pZ)$ muni de la somme est un groupe où le problème du logarithme discret est facile à résoudre tandis que Le groupe $G = (Z/pZ)^*$ muni de la multiplication est un groupe où le problème du logarithme discret est difficile à résoudre si le nombre premier p est trés grand et est bien choisi.

Protocole de Diffie et Hellman

Protocole de Diffie et Hellman

Deux personnes veulent s'échanger une clé ; alors ils se mettent d'accord sur un groupe cyclique G, g un générateur de G et n l'ordre de G.

Protocole de Diffie et Hellman

Deux personnes veulent s'échanger une clé ; alors ils se mettent d'accord sur un groupe cyclique G, g un générateur de G et n l'ordre de G.
Alice choisit un grand nombre entier aléatoire x inférieur à n et envoie à Bachir le résultat du calcul : $A = g^x$.

Protocole de Diffie et Hellman

Deux personnes veulent s'échanger une clé ; alors ils se mettent d'accord sur un groupe cyclique G, g un générateur de G et n l'ordre de G.
Alice choisit un grand nombre entier aléatoire x inférieur à n et envoie à Bachir le résultat du calcul : $A = g^x$.
Bachir choisit un grand nombre entier aléatoire y inférieur à n et envoie à Alice le résultat du calcul : $B = g^y$.

Protocole de Diffie et Hellman

Deux personnes veulent s'échanger une clé ; alors ils se mettent d'accord sur un groupe cyclique G, g un générateur de G et n l'ordre de G.
Alice choisit un grand nombre entier aléatoire x inférieur à n et envoie à Bachir le résultat du calcul : $A = g^x$.
Bachir choisit un grand nombre entier aléatoire y inférieur à n et envoie à Alice le résultat du calcul : $B = g^y$.
Alice calcule $k = B^x$;

Protocole de Diffie et Hellman

Deux personnes veulent s'échanger une clé ; alors ils se mettent d'accord sur un groupe cyclique G, g un générateur de G et n l'ordre de G.
Alice choisit un grand nombre entier aléatoire x inférieur à n et envoie à Bachir le résultat du calcul : $A = g^x$.
Bachir choisit un grand nombre entier aléatoire y inférieur à n et envoie à Alice le résultat du calcul : $B = g^y$.
Alice calcule $k = B^x$;
Bachir calcule $k' = A^y$;

Protocole de Diffie et Hellman

Deux personnes veulent s'échanger une clé ; alors ils se mettent d'accord sur un groupe cyclique G, g un générateur de G et n l'ordre de G.
Alice choisit un grand nombre entier aléatoire x inférieur à n et envoie à Bachir le résultat du calcul : $A = g^x$.
Bachir choisit un grand nombre entier aléatoire y inférieur à n et envoie à Alice le résultat du calcul : $B = g^y$.
Alice calcule $k = B^x$;
Bachir calcule $k' = A^y$;
Les valeurs k et k' sont toutes les deux égales à g^{xy} qui est la clé échangée entre les deux personnes.

Protocole de Diffie et Hellman

Deux personnes veulent s'échanger une clé ; alors ils se mettent d'accord sur un groupe cyclique G, g un générateur de G et n l'ordre de G.
Alice choisit un grand nombre entier aléatoire x inférieur à n et envoie à Bachir le résultat du calcul : $A = g^x$.
Bachir choisit un grand nombre entier aléatoire y inférieur à n et envoie à Alice le résultat du calcul : $B = g^y$.
Alice calcule $k = B^x$;
Bachir calcule $k' = A^y$;
Les valeurs k et k' sont toutes les deux égales à g^{xy} qui est la clé échangée entre les deux personnes.
Si g, A, et B sont connus ; trouver k est un problème appelé problème de Diffie et Hellman.

Cryptosystème d'ElGamal

Cryptosystème d'ElGamal

Le système de cryptage d'ElGamal est un exemple de cryptage en liaison avec l'échange de clè de Diffie-Hellman défini sur le groupe $(Z/pZ)^*$. La sécurité de ce système de cryptage est basée sur la difficulté de résoudre le problème de Diffie-Hellman dans le groupe $(Z/pZ)^*$.

Cryptosystème d'ElGamal

Le système de cryptage d'ElGamal est un exemple de cryptage en liaison avec l'échange de clè de Diffie-Hellman défini sur le groupe $(Z/pZ)^*$. La sécurité de ce système de cryptage est basée sur la difficulté de résoudre le problème de Diffie-Hellman dans le groupe $(Z/pZ)^*$.
Alors si Alice veut envoyer un message m à Bachir, elle calcule $c = g^{xy}\, m$ et envoie (X, c) à Bachir.

Cryptosystème d'ElGamal

Le système de cryptage d'ElGamal est un exemple de cryptage en liaison avec l'échange de clè de Diffie-Hellman défini sur le groupe $(Z/pZ)^*$. La sécurité de ce système de cryptage est basée sur la difficulté de résoudre le problème de Diffie-Hellman dans le groupe $(Z/pZ)^*$.
Alors si Alice veut envoyer un message m à Bachir, elle calcule $c = g^{xy}\, m$ et envoie (X, c) à Bachir.
Pour décrypter ce message, Bachir doit tout simplement multiplier par l'inverse de la clé dans le groupe G :

$$m = g^{p-1-xy}\, c = g^{p-1-xy}\, g^{xy}\, m.$$

Cryptosystème d'ElGamal : Remarques

Remarques

1. Tout ce qui précède reste vrais pour un groupe cyclique fini G quelconque. La sécurité de ce système de cryptage est basée sur la difficulté de résoudre le problème de Diffie-Hellman.

Remarques

1. Tout ce qui précède reste vrais pour un groupe cyclique fini G quelconque. La sécurité de ce système de cryptage est basée sur la difficulté de résoudre le problème de Diffie-Hellman.

2. Il est clair que casser le protocol d'échange de clés de Diffie-Hellman c'est casser le cryptosystème d'ElGamal.

Remarques

1. Tout ce qui précède reste vrais pour un groupe cyclique fini G quelconque. La sécurité de ce système de cryptage est basée sur la difficulté de résoudre le problème de Diffie-Hellman.
2. Il est clair que casser le protocol d'échange de clés de Diffie-Hellman c'est casser le cryptosystème d'ElGamal. Inversement, si on sait casser le cryptosystème d'ElGamal, c'est qu'on sait déchiffrer tout message chiffré par la méthode d'ElGamal, en particulier si le message chiffré est $c = 1$ *alors de l'égalité* $1 = g^{xy} m$*, on trouve que la clé est* $g^{xy} = m^{-1}$.

Remarques

1. Tout ce qui précède reste vrais pour un groupe cyclique fini G quelconque. La sécurité de ce système de cryptage est basée sur la difficulté de résoudre le problème de Diffie-Hellman.

2. Il est clair que casser le protocol d'échange de clés de Diffie-Hellman c'est casser le cryptosystème d'ElGamal. Inversement, si on sait casser le cryptosystème d'ElGamal, c'est qu'on sait déchiffrer tout message chiffré par la méthode d'ElGamal, en particulier si le message chiffré est $c = 1$ alors de l'égalité $1 = g^{xy} m$, on trouve que la clé est $g^{xy} = m^{-1}$.

3. Pour le cryptosystème d'ElGamal dans le cas d'un groupe cyclique quelconque on représente un message m par un élément g_m du groupe et le message chiffré est $c = K.g_m$. Une question qui se pose, c'est comment représenter le message m par l'élément g_m ?

Signature et vérification

Signature et vérification

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.

Signature et vérification

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.
Soient I_m un ensemble de messages, I_s un ensemble de signatures et I_k un ensemble de clés.

Signature et vérification

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.
Soient I_m un ensemble de messages, I_s un ensemble de signatures et I_k un ensemble de clés.

Définition

Un procédé de signature est la donnée pour chaque clé $K \in I_k$ de deux fonctions calculables en temps polynomial :

Signature et vérification

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.
Soient I_m un ensemble de messages, I_s un ensemble de signatures et I_k un ensemble de clés.

Définition

Un procédé de signature est la donnée pour chaque clé $K \in I_k$ de deux fonctions calculables en temps polynomial :
$S_K : I_m \longrightarrow I_s$ une fonction de signature (secrète)

Signature et vérification

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.
Soient I_m un ensemble de messages, I_s un ensemble de signatures et I_k un ensemble de clés.

Définition

Un procédé de signature est la donnée pour chaque clé $K \in I_k$ de deux fonctions calculables en temps polynomial :
$S_K : I_m \longrightarrow I_s$ une fonction de signature (secrète)
$V_K : I_m \times I_s \longrightarrow \{$ vrai , faux $\}$ est la fonction de vérification(publique)
telles que $V_K(M; S) =$ vrai si $S = S_K(M)$ et faux dans le cas contraire.

Signature et vérification

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.
Soient I_m un ensemble de messages, I_s un ensemble de signatures et I_k un ensemble de clés.

Définition

Un procédé de signature est la donnée pour chaque clé $K \in I_k$ de deux fonctions calculables en temps polynomial :
$S_K : I_m \longrightarrow I_s$ une fonction de signature (secrète)
$V_K : I_m \times I_s \longrightarrow \{$ vrai , faux $\}$ est la fonction de vérification(publique)
telles que $V_K(M; S) =$ vrai si $S = S_K(M)$ et faux dans le cas contraire.

La signature RSA

Signature et vérification

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.
Soient I_m un ensemble de messages, I_s un ensemble de signatures et I_k un ensemble de clés.

Définition

Un procédé de signature est la donnée pour chaque clé $K \in I_k$ de deux fonctions calculables en temps polynomial :
$S_K : I_m \longrightarrow I_s$ une fonction de signature (secrète)
$V_K : I_m \times I_s \longrightarrow \{$ vrai , faux $\}$ est la fonction de vérification(publique)
telles que $V_K(M; S) =$ vrai si $S = S_K(M)$ et faux dans le cas contraire.

La signature RSA

Soient $n = pq$, c l'exposant de chiffrèrent et d celui de déchiffrement qui est la clé privé.

La signature électronique est un moyen permettant de remplir la même fonction que la signature ordinaire.
Soient I_m un ensemble de messages, I_s un ensemble de signatures et I_k un ensemble de clés.

Définition

Un procédé de signature est la donnée pour chaque clé $K \in I_k$ de deux fonctions calculables en temps polynomial :
$S_K : I_m \longrightarrow I_s$ une fonction de signature (secrète)
$V_K : I_m \times I_s \longrightarrow \{$ vrai , faux $\}$ est la fonction de vérification(publique) telles que $V_K(M; S) =$ vrai si $S = S_K(M)$ et faux dans le cas contraire.

La signature RSA

Soient $n = pq$, c l'exposant de chiffrèrent et d celui de déchiffrement qui est la clé privé. Pour signer un message $m \in Z/nZ$, Alice calcule la fonction $S(m) = m^d \bmod n$ et la fonction de vérification associée est $V(m; s) =$ vrai si $m = s^c \bmod n$ et faux dans le cas contraire.

Signature et vérification

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

Alice publie p ; g et A.

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

Alice publie p ; g et A. Pour un $k \in (Z/(p-1)Z)^*$ on définit la fonction de signature par :

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

Alice publie p ; g et A. Pour un $k \in (Z/(p-1)Z)^*$ on définit la fonction de signature par :

$sig(m) = (K; S)$ où $K = g^k \mod p$ et $S = (m - aK)k^{-1} \mod p-1$,

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

Alice publie p ; g et A. Pour un $k \in (Z/(p-1)Z)^*$ on définit la fonction de signature par :

$sig(m) = (K; S)$ où $K = g^k \mod p$ et $S = (m - aK)k^{-1} \mod p-1$, et la fonction de vérification par :

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

Alice publie p ; g et A. Pour un $k \in (Z/(p-1)Z)^*$ on définit la fonction de signature par :

$sig(m) = (K;S)$ où $K = g^k \mod p$ et $S = (m - aK)k^{-1} \mod p-1$, et la fonction de vérification par :

$V(m;K;S) = vrai$ si et seulement si $A^K K^S = g^m \mod p$.

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

Alice publie p ; g et A. Pour un $k \in (Z/(p-1)Z)^*$ on définit la fonction de signature par :

$sig(m) = (K;S)$ où $K = g^k \mod p$ et $S = (m - aK)k^{-1} \mod p-1$, et la fonction de vérification par :

$V(m;K;S) = vrai$ si et seulement si $A^K K^S = g^m \mod p$. Cette fonction de vérification permet bien d'authentifier toute signature :

La signature El Gamal

Soient p un grand nombre premier, g un générateur de $(Z/pZ)^*$, a un entier compris entre 0 et $p-2$ et $A = g^a$.

Alice publie p ; g et A. Pour un $k \in (Z/(p-1)Z)^*$ on définit la fonction de signature par :

$sig(m) = (K;S)$ où $K = g^k \mod p$ et $S = (m - aK)k^{-1} \mod p-1$, et la fonction de vérification par :

$V(m;K;S) = vrai$ si et seulement si $A^K K^S = g^m \mod p$. Cette fonction de vérification permet bien d'authentifier toute signature :

Dans Z/pZ on a $K^S = g^{kS} = g^{m-aK}$ puisque $g^{p-1} = 1$ et $A^K = g^{aK}$ donc $A^K K^S = g^m \mod p$.

Signature et vérification

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal :

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ;

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard).

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard). Avec la DSS on obtient une signature plus courte qu'avec El Gamal pour une sécurité identique.

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard). Avec la DSS on obtient une signature plus courte qu'avec El Gamal pour une sécurité identique.

Soient p de taille de 512 ou 1024 bits, q de taille 160 bits et g un élément d'ordre q dans $(Z/pZ)^*$.

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard). Avec la DSS on obtient une signature plus courte qu'avec El Gamal pour une sécurité identique.

Soient p de taille de 512 ou 1024 bits, q de taille 160 bits et g un élément d'ordre q dans $(Z/pZ)^*$. On travaille dans le sous-groupe engendré par g. On a que $p \equiv 1 \bmod q$(ce qui est assuré par l'existence de l'élément g d'ordre q dans $(Z/pZ)^*$).

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard). Avec la DSS on obtient une signature plus courte qu'avec El Gamal pour une sécurité identique.

Soient p de taille de 512 ou 1024 bits, q de taille 160 bits et g un élément d'ordre q dans $(Z/pZ)^*$. On travaille dans le sous-groupe engendré par g. On a que $p \equiv 1 \bmod q$(ce qui est assuré par l'existence de l'élément g d'ordre q dans $(Z/pZ)^*$). Soient $a \in N$; $1 \leq a \leq q-1$ et $A = g^a \bmod p$.

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard). Avec la DSS on obtient une signature plus courte qu'avec El Gamal pour une sécurité identique.

Soient p de taille de 512 ou 1024 bits, q de taille 160 bits et g un élément d'ordre q dans $(Z/pZ)^*$. On travaille dans le sous-groupe engendré par g. On a que $p \equiv 1 \bmod q$(ce qui est assuré par l'existence de l'élément g d'ordre q dans $(Z/pZ)^*$). Soient $a \in N$; $1 \leq a \leq q-1$ et $A = g^a \bmod p$. Les entiers p, q, g et A sont publiques tandis que a est secret.

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard). Avec la DSS on obtient une signature plus courte qu'avec El Gamal pour une sécurité identique.

Soient p de taille de 512 ou 1024 bits, q de taille 160 bits et g un élément d'ordre q dans $(Z/pZ)^*$. On travaille dans le sous-groupe engendré par g. On a que $p \equiv 1 \bmod q$(ce qui est assuré par l'existence de l'élément g d'ordre q dans $(Z/pZ)^*$). Soient $a \in N$; $1 \leq a \leq q-1$ et $A = g^a \bmod p$. Les entiers p, q, g et A sont publiques tandis que a est secret.

La fonction de signature est : $sig(m) = (K; S)$ où

$K = (g^k \bmod p) \bmod q$ avec k un entier quelconque $1 \leq k \leq q-1$ et $S = (m + aK)k^{-1} \bmod q$.

La signature DSS

L'algorithme DSS est une amélioration de la signature de El Gamal : dans un premier temps, il s'est appelé DSA (Digital Signature Algorithm) ; en suite son nom est devenu DSS (Digital Signature Standard). Avec la DSS on obtient une signature plus courte qu'avec El Gamal pour une sécurité identique.

Soient p de taille de 512 ou 1024 bits, q de taille 160 bits et g un élément d'ordre q dans $(Z/pZ)^*$. On travaille dans le sous-groupe engendré par g. On a que $p \equiv 1 \bmod q$(ce qui est assuré par l'existence de l'élément g d'ordre q dans $(Z/pZ)^*$). Soient $a \in N$; $1 \leq a \leq q-1$ et $A = g^a \bmod p$. Les entiers p, q, g et A sont publiques tandis que a est secret.

La fonction de signature est : $sig(m) = (K; S)$ où
$K = (g^k \bmod p) \bmod q$ avec k un entier quelconque $1 \leq k \leq q-1$ et $S = (m + aK)k^{-1} \bmod q$.

La fonction de vérification est : $V(m; K; S) =$ vrai si et seulement si $A^{KS^{-1}} g^{mS^{-1}} \bmod p \bmod q = K$ où S^{-1} est l'inverse de S modulo q.

Attaques du logarithme discret

Attaques du logarithme discret

On suppose que G est un groupe cyclique d'ordre n engendré par g ; on cherche à calculer le logarithme d'un élément $h \in G$. L'attaque la plus simple est la recherche par force brute ou par énumération : on teste tous les entiers $x \in \{0; 1; \ldots; n-1\}$ jusqu'à ce que l'égalité $g^x = h$ soit satisfaite. La complexité de cette méthode est en $O(n)$ opérations, ce qui n'a d'intérêt que pour les groupes G de très petites tailles. On va décrire ici des attaques qui se basent surtout sur le théorème des restes chinois ou sur le paradoxe des anniversaires.

Attaques du logarithme discret

Réduction de Pohlig-Hellman Soit $n = \prod_{i=1}^{N} p_i^{\alpha_i}$ la décomposition en facteurs premiers de l'ordre de G. Si le logarithme discret de h modulo chacun des $p_i^{\alpha_i}$ est connu, on peut retrouver son logarithme relativement à la base g, en utilisant le théorème des restes chinois.

Réduction de Pohlig-Hellman Soit $n = \prod_{i=1}^{N} p_i^{\alpha_i}$ la décomposition en facteurs premiers de l'ordre de G. Si le logarithme discret de h modulo chacun des $p_i^{\alpha_i}$ est connu, on peut retrouver son logarithme relativement à la base g, en utilisant le théorème des restes chinois. Pour tout premier p divisant n on pose $n_p = n/p^{\alpha}$, $g_p = g^{n_p}$ et $h_p = h^{n_p}$; alors g_p est d'ordre p^{α} et $g_p^x = h_p$.

Réduction de Pohlig-Hellman Soit $n = \prod_{i=1}^{N} p_i^{\alpha_i}$ la décomposition en facteurs premiers de l'ordre de G. Si le logarithme discret de h modulo chacun des $p_i^{\alpha_i}$ est connu, on peut retrouver son logarithme relativement à la base g, en utilisant le théorème des restes chinois. Pour tout premier p divisant n on pose $n_p = n/p^{\alpha}$, $g_p = g^{n_p}$ et $h_p = h^{n_p}$; alors g_p est d'ordre p^{α} et $g_p^x = h_p$. Par suite $x_p = x$ modulo p^{α} est le logarithme discret de h_p dans la base g_p.

Réduction de Pohlig-Hellman Soit $n = \prod_{i=1}^{N} p_i^{\alpha_i}$ la décomposition en facteurs premiers de l'ordre de G. Si le logarithme discret de h modulo chacun des $p_i^{\alpha_i}$ est connu, on peut retrouver son logarithme relativement à la base g, en utilisant le théorème des restes chinois. Pour tout premier p divisant n on pose $n_p = n/p^{\alpha}$, $g_p = g^{n_p}$ et $h_p = h^{n_p}$; alors g_p est d'ordre p^{α} et $g_p^x = h_p$. Par suite $x_p = x$ modulo p^{α} est le logarithme discret de h_p dans la base g_p. Réciproquement si x_p est le logarithme discret de h_p dans la base g_p pour tout p divisant n ; alors, d'après le théorème des restes chinois, il existe x unique modulo n tel que $x \equiv x_p \bmod p^{\alpha}$.

Réduction de Pohlig-Hellman Soit $n = \prod_{i=1}^{N} p_i^{\alpha_i}$ la décomposition en facteurs premiers de l'ordre de G. Si le logarithme discret de h modulo chacun des $p_i^{\alpha_i}$ est connu, on peut retrouver son logarithme relativement à la base g, en utilisant le théorème des restes chinois. Pour tout premier p divisant n on pose $n_p = n/p^{\alpha}$, $g_p = g^{n_p}$ et $h_p = h^{n_p}$; alors g_p est d'ordre p^{α} et $g_p^x = h_p$. Par suite $x_p = x$ modulo p^{α} est le logarithme discret de h_p dans la base g_p. Réciproquement si x_p est le logarithme discret de h_p dans la base g_p pour tout p divisant n ; alors, d'après le théorème des restes chinois, il existe x unique modulo n tel que $x \equiv x_p \bmod p^{\alpha}$. Puisque $(g^{-x}h)^{n_p} = g_p^{-x_p} h_p = 1$ pour tout premier p divisant n ; alors l'ordre de $g^{-x}h$ est un diviseur de n_p pour chaque p, ainsi c'est un diviseur de leurs pgcd qui est égal à 1 et ainsi $g^x = h$.

Attaques du logarithme discret

De même on réduit le calcul du logarithme discret dans le cas d'un groupe cyclique d'ordre une puissance d'un nombre premier au calcul du logarithme discret dans le cas d'un groupe cyclique d'ordre un nombre premier de la façon suivante :
Soit $g^x = h$ où g est d'ordre p^α ; alors on peut écrire
$x = x_0 + x_1 p + \ldots + x_{\alpha-1} p^{\alpha-1}$ avec $0 \leq x_i < p$ et on a :

$$(g^x)^{p^{\alpha-1}} = h^{p^{\alpha-1}} = g^{xp^{\alpha-1}} = (g^{p^{\alpha-1}})^{x_0}.$$

L'élément $g^{p^{\alpha-1}}$ est d'ordre p, donc le calcul de x_0 c'est le calcul du logarithme discret de $h^{p^{\alpha-1}}$ dans un groupe d'ordre p. On fait la même chose pour déterminer les autres x_j : en supposant que x_0,...,x_{j-1} sont déterminés on a :

$$g^{x_j p^j + \ldots + x_{\alpha-1} p^{\alpha-1}} = h g^{-x_0 + \ldots + -x_{j-1} p^{j-1}} = h_1.$$

En élevant la dernière équation à la puissance $p^{\alpha-j-1}$; on trouve que x_j est le logarithme discret de l'élément

Attaques du logarithme discret

Pas-de-bébé pas-de-géant

Cette méthode vise d'accélérer la recherche par énumération en essayant de trouver un certain équilibre entre le temps et la mémoire. Soient G un groupe cyclique d'ordre n et $d < n$ un entier ; si $x \in \{0; \ldots; n-1\}$ est le logarithme de h dans la base g, alors il s'écrit de façon unique $x = ad + r$ avec $0 \leq r < d$ et $0 \leq a \leq \lfloor n/d \rfloor$. En particulier, a est l'unique entier entre 0 et $\lfloor n/d \rfloor$ tel que

$$h(g^{-d})^a$$

appartienne à l'ensemble $\{g^i : 0 \leq i < d\}$. Notre algorithme consiste à construire l'ensemble L_d des couples $(i; g^i)$ pour $0 \leq i < d$ dans un premier temps et ensuite calculer

$$h(g^{-d})^k$$

pour $k = 0$ jusqu'à ce qu'on obtient un élément de la forme g^s où $0 \leq s < d$, qui correspond à un unique couple $(s; g^s)$ de L_d. On a alors

$$h(g^{-d})^k = g^s$$

Attaques du logarithme discret

La méthode rho de Pollard

Cette méthode vise à améliorer la complexité en mémoire. C'est Pollard qui avait introduit en 1978 un algorithme probabiliste de calcul du logarithme discret dont la complexité temporelle reste en $O(\sqrt{n})$. Son idée est d'itérer une fonction $F : G \longrightarrow G$ vérifiant les propriétés suivantes :

1. F doit être simple à calculer,
2. étant donnés α et $\beta \in Z/nZ$, on trouve facilement α' et $\beta' \in Z/nZ$ tels que $F(g^{\alpha}h^{\beta}) = g^{\alpha'}h^{\beta'}$,
3. le comportement de F doit être suffisamment proche de celui d'une fonction aléatoire.

Les fonctions simples vérifiant les points 1 : et 2 : sont du type $x \mapsto x^k$ avec k un entier petit, $x \mapsto gx$, $x \mapsto hx$, ou bien des composées de ces trois primitives.

L'approche de Pollard consiste à alterner entre plusieurs fonctions de ce type de la façon suivante : on partitionne G en trois sous-ensembles G_1 ; G_2 et G_3 de tailles comparables, et on définit

$$F(x) = \begin{cases} x^2 & \text{si} \quad x \in G_1 \, ; \\ gx & \text{si} \quad x \in G_2 \, ; \\ hx & \text{si} \quad x \in G_3 . \end{cases}$$

L'approche de Pollard consiste à alterner entre plusieurs fonctions de ce type de la façon suivante : on partitionne G en trois sous-ensembles G_1 ; G_2 et G_3 de tailles comparables, et on définit

$$F(x) = \begin{cases} x^2 & \text{si} \quad x \in G_1 ; \\ gx & \text{si} \quad x \in G_2 ; \\ hx & \text{si} \quad x \in G_3. \end{cases}$$

En partant d'un élément $u_0 = g^{\alpha_0} h^{\beta_0}$, on calcule la suite (u_i) ainsi que les suites (α_i) ; (β_i) de telle sorte que $u_i = F(u_{i-1}) = F^i(u_0) = g^{\alpha_i} h^{\beta_i}$. Comme le groupe est d'ordre fini, la suite (u_i) est périodique : il existe deux entiers i_0 et $l > 0$ tels que $u_{i_0} = u_{i_0+l}$ (et donc $u_i = u_{i+l}$ pour tout $i \geq i_0$).

La collision entre u_{i_0} et $u_{j_0} = u_{i_0+l}$($u_{i_0} = u_{i_0+l}$) entraîne que

$$g^{\alpha_{i_0}} h^{\beta_{i_0}} = g^{\alpha_{j_0}} h^{\beta_{j_0}};$$

si $\beta_{i_0} - \beta_{j_0}$ est premier avec n, l'ordre de G ; on en déduit que le logarithme discret de h relativement à la base g est

$$-(\alpha_{i_0} - \alpha_{j_0})(\beta_{i_0} - \beta_{j_0})^{-1} \bmod n.$$

Si $\beta_{i_0} - \beta_{j_0} \bmod n$ n'est pas inversible, alors il faut recommencer avec un autre élément u_0.

La collision entre u_{i_0} et $u_{j_0} = u_{i_0+l}$($u_{i_0} = u_{i_0+l}$) entraîne que

$$g^{\alpha_{i_0}} h^{\beta_{i_0}} = g^{\alpha_{j_0}} h^{\beta_{j_0}};$$

si $\beta_{i_0} - \beta_{j_0}$ est premier avec n, l'ordre de G ; on en déduit que le logarithme discret de h relativement à la base g est

$$-(\alpha_{i_0} - \alpha_{j_0})(\beta_{i_0} - \beta_{j_0})^{-1} \bmod n.$$

Si $\beta_{i_0} - \beta_{j_0} \bmod n$ n'est pas inversible, alors il faut recommencer avec un autre élément u_0.
On peut montrer que si on itère une fonction aléatoire de G dans G, le temps moyen de parcourir un cycle (collision) est en O($\sqrt{n}$) ; en effet, si F est une fonction uniformément aléatoire, les éléments u_0 ; $u_1 = F(u_0)$, ... forment une suite aléatoire uniformément distribuée dans G jusqu'à la première collision.

Une analyse du type "paradoxe des anniversaires" montre alors que cela arrive au bout de O($\sqrt{n}$) itérations. En pratique F n'est pas aléatoire, mais son comportement est presque similaire pour que cette analyse reste valide.
L'autre difficulté à traiter, est comment détecter les cycles : si on doit stocker toutes les valeurs des u_i jusqu'à obtenir une collision, la complexité en mémoire reste encore en O($\sqrt{n}$). Il existe plusieurs méthodes pour détecter les cycles dont le coût mémoire est en O(1), on présente la plus simple et la plus ancienne due à Floyd. On considère en plus de la suite (u_i), la suite (v_i) telle que $v_i = u_{2i}$ et on itère jusqu'à trouver une collision $u_i = v_i$. Une telle collision se produit dès que i est un multiple de l (la longueur du cycle) et est supérieur à i_0 (l'entrée du cycle), donc pour une valeur de i nécessairement plus petite que $i_0 + l$.