FACTORISATION

ALEXANDRA BRUASSE-BAC

Table des matières

Voir également les chapitres sur les corps finis, sur la primalité, ainsi que sur le calcul du logarithme discret.

En cas de problème, voici la liste des notations utilisées.

Dans ce chapitre, nous allons nous intéresser à une question cruciale vis à vis de la sécurité du cryptosystème RSA : la factorisation des grands nombres entiers. Fort heureusement pour RSA, cette question est très difficile. Cependant, on ne sait toujours pas à l'heure actuelle si le problème de la factorisation est lui-même difficile à résoudre (NP complet par exemple) ou si l'on manque juste d'un bon algorithme.

Les algorithme les plus efficaces permettant - à l'heure actuelle - d'attaquer ce problème sont :

- le crible quadratique (*quadratic sieve*),
- des méthodes utilisant les courbes elliptiques,
- le crible sur le corps des nombres (*number field sieve*).

Notre objectif sera de présenter l'algorithme du crible quadratique (que l'on retrouvera disséqué en détails dans [Buchmann]).

1. La méthode $p-1$

Il existe des algorithmes de factorisation qui sont particulièrement efficaces pour des entiers ayant une forme bien spécifique. Il est donc important de les éviter lorsque l'on choisit un module RSA ! Un exemple de tel algorithme est la méthode $p-1$ de John Pollard que nous allons présenter ici.

Cette méthode est particulièrement adaptée pour trouver des facteurs premiers p tels que $p-1$ n'a que de petits diviseurs premiers.

Soit n un entier que l'on souhaite factoriser. Soit $a \in \{1 \dots n-1\}$ tel que $\text{pgcd}(a,n) = 1$. Supposons que p soit un facteur premier de n, et soit q un entier tel que $p-1 \mid q$ (ie. $q = (p-1)\cdot m$). Comme $\text{pgcd}(a,n) = 1$ et comme $p \mid n$, on

Date: Année 2003-2004.

a également $\text{pgcd}(a,p)=1$. Par conséquent, d'après le petit théorème de Fermat, on a :

$$a^q = (a^{p-1})^m \equiv 1\ [p]$$

d'où $p \mid a^q - 1$. Or on a également $p \mid n$, puis $p \mid \text{pgcd}(a^q-1, n)$, ainsi, $\text{pgcd}(a^q - 1, n) > 1$.

Si l'on a de plus $a^q - 1 \not\equiv 0\ [n]$, alors $\text{pgcd}(a^q-1,n)$ est un diviseur propre de n.

Reste une question en suspend : comment choisir q ? En fait, l'algorithme utilise une borne B qui est choisie en début de calcul. On pose alors :

$$q = \prod_{\substack{p^e \leq B \\ p \text{ premier} \\ e \in \mathbb{N}}} p^e \tag{1}$$

Si tous les facteurs de $p-1$ sont inférieurs à B, alors on obtient bien que $p-1$ divise q ainsi choisi.

Pour factoriser un entier n, l'algorithme $p-1$ est donc le suivant :

- on se fixe tout d'abord une base a telle que $\text{pgcd}(a,n)=1$,
- on choisit une borne B (et on prie très fort pour que, si p est le facteur que l'on recherche, tous les facteurs de $p-1$ soient plus petits que B),
- on calcule l'entier q donné par l'équation (1),
- si $\text{pgcd}(a^q-1,n)$ est un diviseur propre de n, on a terminé, sinon, il faut choisir une borne B plus grande.

Exemple 1.1. Soit $n = 1241143$ l'entier à factoriser. On va choisir comme base $a = 2$ (qui est bien premier avec n).

Reste maintenant à choisir la borne B. Nous allons tout d'abord être très optimiste et essayer $B = 7$. Les puissances de nombres premiers inférieures à 7 sont $1, 2, 3, 5, 7$, par conséquent, on a

$$q = 4 \cdot 3 \cdot 5 \cdot 7 = 420$$

On a : $a^q - 1 \equiv 0\ [n]$. Par conséquent, la borne choisie est trop petite.

Prenons maintenant : $B = 13$. On a alors :

$$q = 8 \cdot 9 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 360360$$

Puis : $a^q - 1 \equiv 861525$, et $\text{pgcd}(a^q - 1, n) = 547$. On calcule alors $1241143 = 547 \cdot 2269$. Grâce au test de Rabin (par exemple), on montre alors que 2269 est premier.

On notera qu'entre autres, tous les nombres de la forme $2^n + 1$ sont donc facilement factorisables ! Il faudra donc les éviter comme modules RSA.

2. Le crible quadratique

C'est en fait l'une des méthodes les plus efficaces de factorisation.

2.1. Idée de base.

On essaye de factoriser un nombre composite impair ou, plus précisément, d'en trouver un diviseur propre. C'est en fait suffisant pour "casser" l'algorithme RSA puisque les modules RSA sont le produit de deux grands premiers.

La méthode du crible quadratique trouve deux entiers x et y tels que

$$x^2 \equiv y^2\ [n] \tag{2}$$

et

$$x \not\equiv \pm y\ [n] \tag{3}$$

Alors n divise $(x-y)(x+y)$ mais ne divise ni $x+y$ ni $x-y$. Par conséquent, $\text{pgcd}(x-y,n)$ est un diviseur propre de n.

On notera que cette idée est également utilisée dans la méthode du crible des corps de nombres, cependant la façon de trouver x et y est alors différente.

2.2. Les entiers x et y.

Nous allons maintenant décrire une méthode pour trouver des entiers x et y satisfaisant les équations (2) et (3). Soit

$$m = [\sqrt{n}]$$

et soit

$$P(X) = (X+m)^2 - n$$

Exemple 2.1. Nous allons tout d'abord présenter la méthode à l'oeuvre sur un exemple, puis nous présenterons le cas général. Soit $n = 7429$ l'entier à factoriser. Prenons $m = 86$, on a $P(X) = (X+86)^2 - 7429$. Par ailleurs :

$$\begin{array}{llll} P(-3) & = 83^2 - 7429 & = -540 & = -1 \cdot 2^2 \cdot 3^3 \cdot 5 \\ P(1) & = 87^2 - 7429 & = 140 & = 2^2 \cdot 5 \cdot 7 \\ P(2) & = 88^2 - 7429 & = 315 & = 3^2 \cdot 5 \cdot 7 \end{array}$$

D'où :

$$\begin{aligned} 83^2 &\equiv -1 \cdot 2^2 \cdot 3^3 \cdot 5 && mod 7429 \\ 87^2 &\equiv 2^2 \cdot 5 \cdot 7 && mod 7429 \\ 88^2 &\equiv 3^2 \cdot 5 \cdot 7 && mod 7429 \end{aligned}$$

En multipliant les deux dernières congruences, on obtient :

$$(87 \cdot 88)^2 \equiv (2 \cdot 3 \cdot 5 \cdot 7)^2 \ [7429]$$

On peut donc choisir $x = 87 \cdot 88 \ [n] = 227$ et $y = 2 \cdot 3 \cdot 5 \cdot 7 \ [n] = 210$.

En fait, le point majeur de cet exemple est que pour certains nombres s, $P(s)$ n'a que de petits facteurs premiers. On peut alors utiliser la congruence

$$(s+m)^2 \equiv P(s) \ [n]$$

en choisissant bien les nombres s pour trouver x et y.

Nous pouvons maintenant présenter la méthode générale de construction de x et y. On choisit un entier B positif et on cherche des entiers s tels que $P(s)$ n'ait que des facteurs premiers appartenant à une base de factorisation :

$$F(B) = \{p \text{ premier} \,;\, p \leq B\} \cup \{-1\}$$

De telles valeurs de $P(s)$ sont appelées B-lisses.

Il faut alors trouver autant de valeurs de s que $F(B)$ n'a d'éléments (disons l, voir la section suivante pour cette étape). On a alors l éléments $s_1, \ldots, s_l$ tels que $P(s_1), \ldots, P(s_l)$ soient B-lisses, et $F(B) = \{p_1, \ldots, p_l\}$. Pour tout $i \in \{1 \ldots l\}$, comme $P(s_i)$ est B-lisse, les facteurs premiers de $P(s_i)$ sont les éléments de $F(B)$, dont $P(s_i)$ s'écrit :

$$P(s_i) = \prod_{j=1}^{l} p_j^{\alpha_j^i}$$

On va chercher un produit de ces éléments qui soit un carré. Il s'écrit :

$$P(s_1)^{\lambda_1} \cdots P(s_l)^{\lambda_l} = \prod_{j=1}^{l} p_j^{\lambda_1 \alpha_j^1 + \cdots + \lambda_l \alpha_j^l}$$

pour $\lambda_1, \dots, \lambda_l \in \mathbb{N}$. Ce produit est un carré si et seulement si $\lambda_1, \dots, \lambda_l$ est une solution du système linéaire suivant :

$$\begin{pmatrix} \alpha_1^1 & \cdots & \alpha_1^l \\ \alpha_2^1 & \cdots & \alpha_2^l \\ \vdots & & \vdots \\ \alpha_l^1 & \cdots & \alpha_l^l \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_l \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} [2] \\ [2] \\ \vdots \\ [2] \end{matrix}$$

Ce système est un système linéaire sur $\mathbb{Z}/2\mathbb{Z}$ et peut donc être résolu par le pivot de Gauss.

2.3. **Le criblage.**

Toute la question est maintenant : *comment trouver les entiers s tels que $P(s)$ soit B-lisse ?* Eh bien : en criblant !

Essayer tous les entiers s un par un et vérifier si l'on peut les factoriser en utilisant les éléments de $F(B)$ est clairement très coûteux. Une technique plus efficace consiste à utiliser des techniques de criblage.

L'idée est la suivante : on se fixe un intervalle de criblage :

$$S = \{-C, -C+1, \dots, 0, \dots, C-1, C\}$$

C est appelé la borne de l'intervalle de criblage. Pour tout $s \in S$, on calcule la valeur $P(s)$, puis successivement, pour chaque $p \in F(B)$, on divise les valeurs $P(s)$ par la plus grande puissance possible de p. Les nombres B-lisses sont ceux pour lesquels il reste 1 ou -1 à la fin de ce processus.

Cependant, si l'on doit tester la divisibilité de chaque $P(s)$ par chaque p, la tâche reste très coûteuse. Supposons que l'on connaisse les zéros[1] de $P(X)$ modulo un nombre premier $p \in F(B)$, c'est-à-dire que l'on ait déterminé tous les $s \in \{1, \dots, n\}$ tels que $P(s)$ soit divisible par p. Alors les nombres $s \in S$ tels que p divise $P(s)$ sont exactement les racines de $P(X)$ modulo p plus un multiple de p (on n'a pas à effectuer de divisions inutiles). Cette technique est appelée le criblage avec p (voir [Buchmann] p.177 pour plus de détails).

3. Efficacité des algorithmes de factorisation

Dans cette partie, nous allons brièvement nous intéresser à la question de l'efficacité et de la complexité du crible quadratique et des autres algorithmes de cryptage. Un fait surprenant : jusqu'à ce jour, personne n'a encore été capable de donner un analyse complète et rigoureuse de la méthode du crible quadratique. Sous certaines hypothèses (que nous verrons plus tard), il a été possible d'estimer sa complexité, mais ce résultat n'est pas valable en toute généralité.

Commençons par définir une fonction qui nous servira à mesurer la complexité des différents algorithmes. Etant donnés n, u, v des nombres, on définit :

$$L_n(u, v) = e^{v(\log n)^u (\log\log n)^{1-u}}$$

On a, en particulier :

$$L_n(0, v) = (\log n)^v$$

et :

$$L_n(1, v) = e^{v \log n}$$

Ainsi, si un algorithme qui factorise l'entier n (dont la décomposition binaire est de longueur $[\log_2 n]$ s'exécute en temps $L_n(0, v)$, alors cette algorithme est polynômial en la taille de l'entrée. S'il s'exécute en temps $L_n(1, v)$, alors il est exponentiel en

[1] Le calcul des zéros d'un polynôme dans $\mathbb{F}_p$ se fait de manière assez efficace. On pourra se reporter à [LN,86] p. 150-159.

la taille de son entrée. Enfin, s'il s'exécute en temps $L_n(u, v)$ avec $0 < u < 1$, alors il est sous-exponentiel.

On choisit les bornes B et C de sorte que le nombre de le nombre d'éléments B-lisses et le nombre d'éléments de $F(B)$ soient à peut près égaux. L'*hypothèse d'analyse* (qui n'est pas prouvée, mais qui semble expérimentalement satisfaite) est que la proportion d'entiers B-lisses $s \in S$ est la même que la proportion d'entiers B-lisses inférieurs à $\sqrt{n}$. On montre alors (voir [Buchmann]) que le crible quadratique s'exécute en temps

$$L_n(\frac{1}{2}, 1 + o(1))$$

Sous l'hypothèse que nous avons faite, le crible quadratique est donc sous-exponentiel.

Mentionnons brièvement la complexité de quelques autres algorithmes :

- L'algorithme le plus efficace dont la complexité a pu être prouvée est un algorithme probabiliste utilisant des formes quadratique. Il s'exécute en $L_n(1/2, 1+o(1))$. En pratique, cependant, le crible quadratique est plus efficace.
- La méthode des courbes elliptiques (proche de la méthode $p-1$ de Pollard) s'exécute en $L_p(1/2, \sqrt{1/2})$ où p est le plus petit facteur premier de n. Ainsi, si la méthode du crible quadratique dépend de la taille de n, la méthode des courbes elliptiques sera beaucoup plus rapide quand n à un petit facteur premier. Lorsque les facteurs premiers sont de la taille de $\sqrt{n}$, la complexité est de l'ordre de $L_n(1/2, 1)$, mais en pratique, le crible quadratique est tout de même plus efficace.
- Enfin, en 1988, John Pollard (eh oui, toujours lui !) a montré que l'on pouvait descendre au-dessous de la complexité $L_n(1/2, 1)$ avec le crible sur les corps de nombres dont la complexité est de $L_n(1/3, (64/9)^{1/3})$. Cet algorithme se rapproche donc beaucoup plus que ses pères d'un algorithme polynômial.

Bibliographie

[Buchmann] J.A. Buchmann. *Introduction to cryptography*. Springer, 2001.

[LN,86] R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications*. Cambridge University Press, 1986.