Algorithme ECM de factorisation d'entiers

1 L'algorithme $p-1$ de Pollard

1.1 Principe

Voici tout d'abord un algorithme quelques similitudes avec celui de l'algorithme ECM. Le principe, si on cherche à décomposer N, est de trouver un élément g non inversible dans $\mathbb{Z}/N\mathbb{Z}$ et de regarder $\text{pgcd}(g, N)$. Ce pgcd sera différent de 1 (car g est non inversible) ; il reste à espérer qu'il sera différent de 0 $\mod N$. En outre il y a quelques restrictions sur les entiers à décomposer.

Définition 1. *Soient $B \in \mathbb{N}^*$. Un entier $N = \prod_p p^{\alpha_p}$ est dit* B-superlisse *(traduction française de* B-powersmooth*) si :* $\forall p,\ p^{\alpha_p} \leqslant B$.

Plus précisément, l'algorithme pourra trouver un facteur p de N si $p-1$ est B-superlisse, avec B fixé au préalable.

En effet, soit $N = p^r m$ avec $p-1$ B-superlisse. Alors $p-1$ divise $B!$.
Alors pour tout a premier avec p ($a \wedge p = 1$), $a^{B!} = \left(a^{(p-1)}\right)^k \equiv 1[p]$.

Le bon candidat est donc $g = a^{B!} - 1$: il est divisible par p et donc p divise $\text{pgcd}(g, N)$. Si ce pgcd n'est pas nul modulo N, on a gagné.

L'algorithme fonctionne donc de la sorte :

- On choisit un B, appelé borne de lissage.
- On choisit un point $a \in \{2...B\}$ de départ. Si $\text{pgcd}(a, N) = d > 1$, on a de la chance et on a terminé.
- Sinon, on calcule les puissances successives de a. Pour t de 2 à B : $a \leftarrow a^t \mod N$ en vérifiant à chaque étape si $\text{pgcd}(a-1, N)$ est un facteur non trivial de N.
- Si on a parcouru tout jusqu'à $t = B$ sans succès, alors on recommence avec un autre a. Ceci est assez rare car on sait que nécessairement on va trouver un facteur de N qui n'est pas 1. L'échec survient quand, par malchance, ce facteur est multiple de N.

2 Avec des courbes elliptiques

L'algorithme ECM présenté maintenant met en jeu le même principe fondamental : trouver un élément g non premier avec N à décomposer. La différence est qu'au lieu de prendre des itérations successives d'un point de $(\mathbb{Z}/N\mathbb{Z})^*$, on va se placer dans le groupe d'une courbe elliptique. Cela appelle donc pour commencer un rappel sur cette structure.

2.1 Loi de groupe sur les courbes elliptiques

2.1.1 Quelques rappels sur les courbes elliptiques

Rappelons-nous pour commencer la défiition d'une courbe elliptique :

Définition 2. *Soit k un corps de caractéristique différente de 2 ou 3. Une courbe elliptique sur K est une courbe projective d'équation affine $y^2 = x^3 + ax + b$, avec $a, b \in k$.*

On rappelle qu'une courbe elliptique est naturellement munie d'une loi de groupe dont on note $\mathcal{O}$ le neutre.

Remarque. Dans le cas d'une courbe elliptique répondant à une équation du type $y^2 = x^3 + ax + b$, on peut interpréter graphiquement cette loi de groupe en remarquant que la courbe est symétrique par rapport à l'axe des abscisses :

- $P + Q$ correspond au symétrique de $P * Q$ par rapport à l'axe horizontal.
- $-P$ correspond au symétrique de P par rapport à l'axe horizontal.
- $\mathcal{O}$ correspond au point à l'infini dans la direction de l'axe vertical.

2.1.2 Formules explicites

On peut construire de manière théorique une structure de groupe sur notre courbe elliptique. Mais il est intéressant pour le problème qui nous occupe d'avoir des expressions explicites des différentes opérations. On se place dans le cas d'une courbe définie (sur k de caractéristique $p > 3$) par l'équation affine $y^2 = x^3 + ax + 1$ sur laquelle on considère deux points $P(x_P, y_P)$ et $Q(x_Q, y_Q)$.

Proposition 1 (Formule de l'inverse)**.** *L'inverse de P est son symétrique par rapport à l'axe horizontal. Donc* $\begin{cases} x_{-P} = x_P \\ y_{-P} = -y_P \end{cases}$

Proposition 2 (Formule de l'addition de P et Q distincts)**.**
- *Si $x_P = x_Q$ alors P et Q étant distincts, on a $P + Q = \mathcal{O}$.*
- *Si $x_P \neq x_Q$ alors* $\begin{cases} x_{P+Q} = \lambda^2 - x_P - x_Q \\ y_{P+Q} = \lambda(x_P - x_{P+Q}) - y_P \end{cases}$ *avec* $\lambda = \dfrac{y_P - y_Q}{x_P - x_Q}$

Il reste à donner la formule de l'addition de P avec lui-même.

Proposition 3 (Formule du doublement)**.**
- *Si $y_P = O$ alors la tangente en P est verticale et on a $2P = \mathcal{O}$.*
- *Si $y_P \neq 0$ alors* $\begin{cases} x_{2P} = \lambda^2 - 2x_P \\ y_{2P} = \lambda(x_P - x_{2P}) - y_P \end{cases}$ *avec* $\lambda = \dfrac{3x_P^2 + a}{2y_p}$

2.2 L'algorithme

On se donne une courbe elliptique E définie par une équation affine de la forme $y^2 = x^3 + ax + 1$, avec pour commencer $a = 1$. On prend un point P sur cette courbe. Par exemple $[0, 1]$ marche pour tout a et quel que soit le corps de base.

On fixe comme dans le cas de l'algorithme de Pollard une borne de lissage B. Soit $N = p^\alpha q$ à décomposer avec p premier tel que $\sharp E(\mathbb{F}_p)$ soit B-superlisse. Cette condition est assez

naturelle : Dans le $p-1$ de Pollard, la condition était la superlissité de $p-1$, ie. de $\sharp\mathbb{F}_p^*$. Ici il se trouve simplement qu'on travaille avec un groupe un peu plus compliqué : $E(\mathbb{F}_p)$.

Notons $m = \sharp E(\mathbb{F}_p)$. Alors $m|B!$.

L'idée est d'utiliser que dans $E(\mathbb{F}_p)$ qui est un groupe fini, on a $mP = \mathcal{O}$. Le problème est qu'on ne connaît pas p *a priori* et donc on ne connaît pas m non plus. Il faudra donc viser plus large que m. On va utiliser que $(B!)P = \mathcal{O}$ sur $E(\mathbb{F}_p)$. Et puis comme on ne connaît pas p, il va falloir faire tous les calculs sur $E(\mathbb{Z}/N\mathbb{Z})$. Mais dans les formules d'addition, on risque de diviser par des éléments non inversibles ($\mathbb{Z}/N\mathbb{Z}$ n'est pas un corps), notamment en calculant le λ... Ce qui est remarquable, c'est que c'est précisément ça qu'on recherche.

Reprenons les choses dans l'ordre :

- La première chose à faire est de se convaincre qu'un certain kP peut se calculer simplement en doublant ou en ajoutant P successivement. Par exemple pour calculer $5P$, on décompose 5 en binaire (101), on part de $\mathcal{O}$ et on lit 101 de gauche à droite. Quand on lit un 0, on double le résultat ; quand on lit un 1, on le double et on ajoute P. Ainsi ici $\mathcal{O} \to P \to 2P \to (2.2+1)P$.
- On se place dans $E(\mathbb{Z}/N\mathbb{Z})$ où on prend un point P ($[0,1]$ convient très bien comme on l'a déjà dit).
- Comme $p|N$, $P \in E(\mathbb{F}_p)$ et donc $mP = \mathcal{O} = (B!)P$ dans $E(\mathbb{F}_p)$
- (l'astuce est là) Quand on obtient une itération de P égale à $\mathcal{O}$ dans $E(\mathbb{F}_p)$, cela signifie qu'en faisant les calculs dans $E(\mathbb{Z}/N\mathbb{Z})$, on a appliqué une formule illicite. En effet, quand le résultat d'une somme ou d'un doublement appartient à $E(\mathbb{Z}/N\mathbb{Z})$, ses coordonnées sont données par les mêmes formules que sur $E(\mathbb{F}_p)$. Le fait de sortir de $E(\mathbb{Z}/N\mathbb{Z})$ correnspond donc à un calcul illicite. En d'autres termes, on a divisé par quelque chose qui est nul modulo N. Quand ça arrive, il suffit de retrouver ce petit facteur non inversible ; notons-le g. Alors on retrouve le même principe que pour l'algorithme $p-1$ de Pollard. On regarde $\text{pgcd}(g, N)$ qui est un excellent candidat de facteur propre de N. Il le sera presque toujours, sauf s'il est congru à 0 modulo N, auquel cas il faudra recommencer avec une autre courbe (un autre a).

Une question se pose encore : Quel itération de P calculer ?

Réponse : Comme $(B!)P = \mathcal{O}$ dans $E(\mathbb{F}_p)$, on est sûr en allant jusqu'à $B!$ d'avoir un candidat g. Mais $B!$ c'est vraiment énorme alors on va faire un peu mieux : soient $p_1, \ldots, p_k$ tous les nombres premiers inférieurs ou égaux à B. On calcule pour chacun une puissance s_i telle que $p_i^{s_i} \leqslant B < p_i^{s_i+1}$. Et alors $m | \prod_{i=1}^{k} p_i^{s_i}$ et on peut donc s'arrêter à cette quantité pour le calcul des itérés successifs de P.

Questions.

1. Implémenter l'algorithme de Pollard.
2. Démontrer la formule pour la somme de deux points.
3. Implmenter la loi de groupe d'une courbe elliptique d'équation $y^2 = x^3 + ax + 1$.
4. Écrire un algorithme de factorisation par courbes elliptiques. En évaluer la complexité.
5. Comparer les performances avec l'algorithme de Pollard.