Factorisation avec l'algorithme ECM

JEDDI Skander, MAZELET Florent

October 22, 2023

1 Introduction

Factoriser des grands nombres en produit de nombres premiers est un sujet au coeur de la cryptographie moderne. Il existe de nombreux algorithmes et procédés de factorisation, plus ou moins sophistiqués, comme le crible d'Eratosthène, la méthode rho de Pollard (1975), ou le crible de corps des nombres généralisé(Pollard 1988), qui est l'algorithme le plus rapide pour des grands nombres connu à ce jour. Nous allons présenter la méthode de factorisation ECM découverte par A.Lenstra (1987) et ses similarités avec la méthode (p-1)-pollard. L'algorithme ECM (elliptic-curve method) est le 3e algorithme de factorisation le plus rapide derrière l'algoritme du crible quadratique et le cribile de corps des nombres généralisé. Il reste l'algorithme le plus efficace pour des diviseurs de moins de 50 chiffres. Son temps d'exécution est dépendant du plus petit facteur p du nombre n à factoriser, ce qui le rend très utile pour exhiber des "petits" facteurs premiers de grands nombres, et comme il est indépendant de la taille de n, on peut l'appliquer sur un nombre de grande taille sur lequel les cribles ne pourraient pas s'appliquer.

Nous allons avant tout rappeler quelques notions d'algèbre importantes pour la compréhension du sujet traité.

1.1 Rappels

Propriété 1.1.1. *Quelques propriétés*

- *Si* $n = \prod_{i=1}^{r} p_i^{e_i}$, $(\mathbb{Z}/n\mathbb{Z}) \cong \prod_{i=1}^{r} (\mathbb{Z}/p_i^{e_i}\mathbb{Z})$
- *Soit* $a \in (\mathbb{Z}/n\mathbb{Z})$. *a est inversible si et seulement si* $pgcd(a, n) = 1$
- *Petit théorème de Fermat : Soit p un nombre premier et a un entier non divisible par p, alors*

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

- *Ordre d'un groupe : Un groupe G est d'ordre n si il contient n élements.*
- *Ordre d'un élement : Un élement a d'un groupe G est d'ordre k si k est le plus petit entier tel que* $a^k = 1$ *(noté multiplicativement), où* 1 *est l'élement neutre de G*
- *Théorème de Lagrange : Pour tout groupe fini G et tout sous groupe H de G, l'ordre de H divise celui de G.*

Définition 1.1.2. *Friabilité*

- *Un entier est dit B-friable si tous ses diviseurs premiers sont inférieurs ou égaux à B*
- *Un entier est dit B-ultrafriable si tous ses diviseurs de la forme* p^n, *avec p premier sont inférieurs ou égaux à B.*

Dans ce document nous parlerons toujours d'entiers friables et non ultrafriables. Les definitions et propriétés restent cependant adaptables à des entiers ultrafriables.

2 Methode (p-1)-pollard

L'algorithme trouve un diviseur non trivial d'un nombre n donné.
Soit n un entier divisible par un nombre premier p. Prenons un entier $a \in (\mathbb{Z}/p\mathbb{Z})^*$. Alors on sait que $a^{p-1} \equiv 1 \pmod{p}$ d'après le petit théorème de Fermat. Cela implique que pour tout multiple M de $p-1$, on a que

$$a^M \equiv 1 \pmod{p} \tag{1}$$

De plus, on sait que si $a \equiv b \pmod{n}$, alors $a \equiv b \pmod{p}$.
Le principe de l'algorithme repose sur le fait que si $p-1$ est $B-$friable, ou B est une borne choisie, alors en prenant par exemple $M = ppcm(1, .., B)$ ou $M = B!$, on aura forcément $a^M \equiv 1 \pmod{p}$. Ainsi en calculant $b \equiv a^M \pmod{n}$ et en calculant le $pgcd(b-1, n)$, on peut trouver un facteur de n. Il peut cependant arriver que le $pgcd(b-1, n)$ soit égal à 1 ou n. Dans ce cas on doit diminuer ou augmenter la borne B, calculer M autrement(par exemple une factorielle, ou un produit de nombres premiers élevés à une certaine puissance), ou changer a.

L'algorithme pour la méthode p-1 est donc le suivant :

Algorithm 1 Algorithme pour la méthode p-1 de Pollard

```
a ← Rand(1, n)
M ← B!
p ← pgcd(a^M (mod n) − 1, n)
if p ≠ 1 and p ≠ n then
  return p
else
  return 0 {On retourne échec, il faut changer a,M, ou B}
end if
```

Nous allons voir dans la suite de ce document que la méthode ECM est similaire à la methode (p-1)-Pollard.

3 Courbes elliptiques et algorithme ECM

3.1 Définitions

Soit K un corps. Une courbe elliptique sur K est une paire d'elements $a, b \in K$ tel que $4a^3 + 27b^2 \neq 0$. Ces elements sont les coefficients de l'equation de Weierstrass

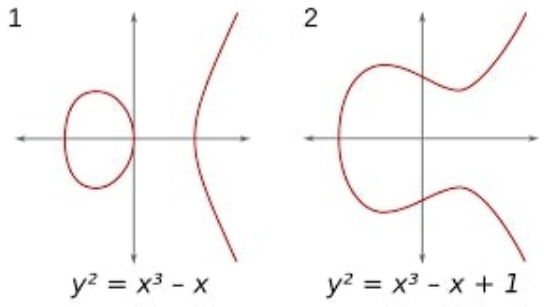
$$y^2 = x^3 + ax + b \tag{2}$$

On appelle la courbe elliptique (a, b) par $E_{a,b}$ ou E. L'ensemble des points $E(K)$ d'une telle courbe elliptique sur K est défini par

$$E(K) = \{(x, y, z) \in P^2(K) : y^2 z = x^3 + axz^2 + bz^3\}, \tag{3}$$

Où $P^2(K)$ est le plan projectif sur K : ce sont les classes d'équivalences des triplets $(x, y, z) \in K \times K \times K$ non tous nuls tel que (x, y, z) et (x', y', z') sont équivalents si il existe $c \in K^*$ tel que $cx = x'$, $cy = y'$ et $cz = z'$. La classe d'équivalence de (x, y, z) sera notée $(x : y : z)$

Exemple 3.1.1. *Voici un exemple de deux courbes elliptiques*



Définition 3.1.2. *Soit E une courbe elliptique sur un corps K. Alors $E(K)$ ne contient qu'un seul point $(x : y : z)$ pour lequel $z = 0$, c'est le point $(0 : 1 : 0)$. On appelera ce point le point à l'infini, que l'on notera 0. Les autres points de $E(K)$ sont les points $(x : y : 1)$, tel que x, y satisfont $y^2 = x^3+ax+b$.*

On peut montrer que $E(K)$ a une structure de groupe abélien pour une loi bien choisie. Explicitons cette loi et comment la construire a l'aide de la représentation graphique de cette courbe (en prenant $K = \mathbb{R}$). Pour aider le lecteur des image sont disponibles sur la page suivante:

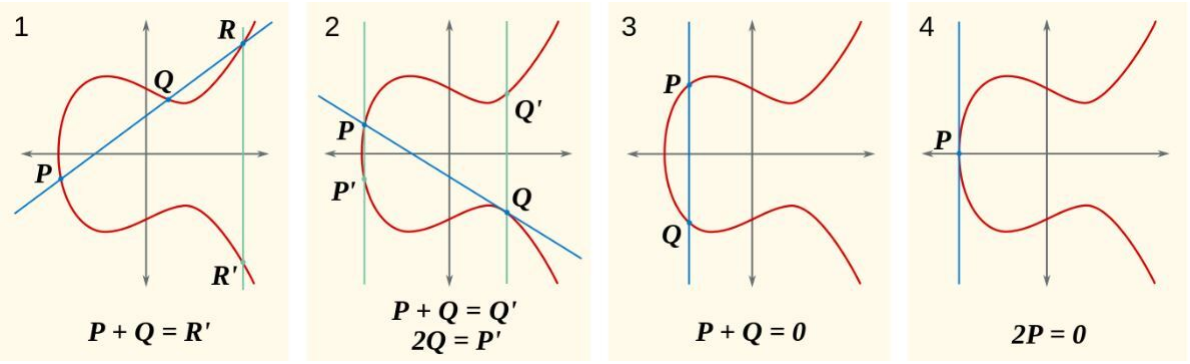
Pour définir l'addition de deux points distincts P et Q sur la courbe elliptique $E(K)$, considérons la droite qui passe par ces deux points. Cette droite coupe la courbe E en un troisième point R. Ce point R est un bon candidat pour être défini comme la somme de P et Q. Cependant, ce choix ne permet pas d'obtenir les propriétés attendues d'un groupe additif abélien. Par exemple, l'élément "zéro" (éléments additif neutre) n'est pas bien défini. On considère alors le symétrique de ce point R par rapport à l'axe horizontal comme la "somme" de P et Q. Cela revient à prendre un point dont l'abscisse x est la même que R et dont l'ordonnée est opposée à R, point qui appartient bien à la courbe. Nous notons alors ce point comme $P + Q$. Nous verrons plus tard pourquoi cette notation est valide, c'est-à-dire que la structure produite a toutes les propriétés que attendues de l'addition dans un groupe abélien.

Définition 3.1.3. *Soient P,Q deux points sur une courbe elliptique. On définit la somme $P + Q$ comme suit:*

1. *Si la droite (PQ) est verticale, le point R intersection de la droite et de la courbe est le point à l'infini, et il est son propre symétrique par rapport à l'axe des abscisses. On a donc $P + Q = 0$*
2. *Si la droite (PQ) est tangente à la courbe en l'un des deux points, le point R est le point de tangence, et donc la somme de P et Q est le symétrique de ce point.*
3. *La somme d'un point P avec lui même se fait en considérant la tangente à la courbe passant par P. La tangente coupe la courbe en un point R distinct ou non de P. Le point résultant de l'addition est encore le symétrique de ce point.*
4. *Sinon, le point R correspond à l'intersection de la droite (PQ) avec la courbe, et la somme $P+Q$ est égale à l'opposé de R par rapport à l'axe des abscisses.*

Cette définition de l'addition invite naturellement au choix du point à l'infini comme élément neutre pour l'addition. Soit 0 ce point à l'infini. Pour trouver $P + 0$ selon la méthode décrite, nous devons tracer une ligne passant par les points P et 0. C'est la verticale passant par P. Elle coupe la courbe elliptique exactement au point P' symétrique de P par rapport à l'axe des abscisses. Le point $P + 0$ cherché, par définition de l'addition, est le symétrique de ce point P', donc c'est P lui-même. On a bien trouvé que $P+0 = P$, ce qui correspond bien à ce qu'on attend d'un élément neutre pour l'addition.

Exemple 3.1.4. *Voici un exemple des définitions énoncées ci-dessus.*



Cette définition de l'addition s'adapte parfaitement en passant aux calculs sur les coordonnées des points de la courbe elliptique. Les nouvelles coordonées d'un point $R = (x_R, y_R)$ somme de deux points $P = (x_P, y_P)$ et $Q = (x_Q, y_Q)$ appartenant à la courbe d'équation $y^2 = x^3 + ax + b$ s'obtiennent de cette façon par le calcul :

Propriété 3.1.5 (Addition)**.** *L'addition se fait de la manière suivante :*

- *Si* $x_P \neq x_Q$ *:*

$$x_R = \lambda^2 - x_P - x_Q$$
$$y_R = y_P + \lambda(x_R - x_P)$$

  où $\lambda = (y_P - y_Q)/(x_P - x_Q)$ *est la pente.*

- *Si* $x_P = x_Q$ *et* $y_P = -y_Q$*, alors* $R = 0$*.*
- *si* $x_P = x_Q$ *et* $y_P = y_Q \neq 0$*, alors :*

$$x_R = \lambda^2 - 2x_P$$
$$y_R = y_P + \lambda(x_R - x_P)$$

  où $\lambda = (3x_P^2 + a)/(2y_P)$*.*

Nous venons donc de définir une loi de composition sur les points de la courbe $E(K)$, notée $+$, ce qui nous donne le théorème suivant:

Theorème 3.1.6. *L'ensemble des points de* $E(K)$ *(en incluant le point à l'infini), muni de cette loi de composition, forme un groupe commutatif.*

Démonstration:

- Si deux points P et Q sont des points de la courbe ou le point à l'infini, il en est de même de $P+Q$ et de $-P$, et la loi définie est bien une loi de composition interne. En effet $P+Q$ appartient à la courbe par construction, et Si $P = (x, y)$ alors $-P = (x, -y)$, et on a que

$$(-y)^2 = y^2 = x^3 + ax + b \tag{4}$$

  et $-P$ appartient bien à la courbe

- Le fait que le point à l'infini est élément neutre a été déjà vérifié plus haut.

- La droite joignant un point quelconque P et son symétrique par rapport à l'axe des abscisses, noté $-P$, est une droite verticale, le 3e point d'intersection avec la courbe est donc le point à l'infini (qui est son propre symétrique par rapport à l'axe des abscisses), d'où $P + (-P) = 0$;

- La commutativité est évidente, par définition d'une droite; on peut aussi la prouver par le calcul : Soit $P = (x_p, y_p), Q = (x_q, y_q)$. Alors on a

$$x_{p+q} = \lambda_1^2 - x_p - x_q \tag{5}$$

et

$$x_{q+p} = \lambda_2^2 - x_q - x_p \tag{6}$$

On remarque que $\lambda_1 = \lambda_2$ et donc que $x_{p+q} = x_{q+p}$.
De plus

$$y_{p+q} = y_p + \lambda(x_{p+q} - x_p) \tag{7}$$

et

$$y_{q+p} = y_q + \lambda(x_{p+q} - x_q) \tag{8}$$

En soustrayant les deux équations, on a

$$y_{p+q} - y_{q+p} = y_p - y_q + \lambda(x_{p+q} - x_p) - \lambda(x_{p+q} - x_q) \tag{9}$$

et donc

$$y_{p+q} - y_{q+p} = y_p - y_q + \lambda(x_q - x_p) = y_p - y_q - y_p + y_q = 0 \tag{10}$$

Si $x_p = x_q$ la preuve est similaire. On a donc bien que la loi est commutative;

- La seule propriété difficile à démontrer est en fait l'associativité, c'est-à-dire que, pour trois points quelconques:

$$(P + Q) + R = P + (Q + R),$$

ce qui peut se faire directement par le calcul, mais cela devient vite très lourd à écrire. Une idée de preuve alternative(cf réference 7) utilise un Lemme disant que si une courbe cubique passe par 8 points du plan projectif, alors il existe un 9ème point tel que toute courbe passant par les 8 premiers points passe aussi par le 9e point. On peut grace à ce Lemme s'arranger pour construire 8 points tel que le 9e point soit égal à $-((P + Q) + R)$ selon une courbe, et égal à $-(P + (Q + R))$ selon une autre. Le lemme nous donne que ces deux points sont égaux et prouve l'associativité.

On a vu que lorsqu'on peut tracer une courbe sur le plan l'addition est graphiquement représentable, mais les formules sont algébriques et définies sur $\mathbb{Z}$ donc elles sont valables lorsque K est un corps fini $\mathbb{F}_p$. Cependant pour l'algorithme ECM nous regardons les courbes elliptiques sur $(\mathbb{Z}/n\mathbb{Z})$, où n est un entier positif. Cet ensemble n'est pas un corps si n est composé. Pour a, b dans $(\mathbb{Z}/n\mathbb{Z})$ on considère la courbe cubique $E = E_{a,b}$ définie sur$(\mathbb{Z}/n\mathbb{Z})$ par l'équation

$$y^2 = x^3 + ax + b \tag{11}$$

L'ensemble V des points d'une telle courbe sur $(\mathbb{Z}/n\mathbb{Z})$ est défini par :

$$V = \{(x : y : z) \in P^2(\mathbb{Z}/n\mathbb{Z}) : y^2 = x^3 + axz^2 + bz^3)\} \tag{12}$$

Si $6(4a^3 + 27b^2) \in (\mathbb{Z}/n\mathbb{Z})^*$ alors l'ensemble a une structure de groupe abélien, mais elle n'est pas utilisée pour l'algorithme ECM car elle est trop compliquée (cf reférence 1). On travaillera donc avec une "pseudo-addition" sur un sous ensemble de V que l'on appelera $E(\mathbb{Z}/n\mathbb{Z})$.

$$E(\mathbb{Z}/n\mathbb{Z}) = \{(x : y : 1) : x, y \in \mathbb{Z}/n\mathbb{Z}\} \cup 0,$$

où 0 est le point à l'infini $(0 : 1 : 0)$. Pour cet ensemble nous utiliserons l'addition décrite précedemment mais nous allons voir qu'elle n'est pas toujours possible.

3.2 Propriétés de $E(\mathbb{F}_p)$

Nous allons voir dans cette section des propriétés utiles de $E(\mathbb{F}_p)$ pour ECM. Les propriétés et théorèmes énoncés ne seront pas démontrés dans ce rapport. Toutes les réferences seront données à la fin.

Theorème 3.2.1. *Théorème de Hasse* (1936) *:*

$$\#E(\mathbb{F}_p) = p + 1 - t \tag{13}$$

où $t \in \mathbb{Z}$ *et* $|t| \leq 2\sqrt{p}$

Le théorème nous donne une borne sur le nombre de points d'une certaine courbe elliptique sur $\mathbb{F}_p$, et donc sur l'ordre de chaque point. Ce théorème est au centre de l'algorithme ECM, car ils nous permet d'estimer un nombre k pour lequel pour tout $P \in E(\mathbb{F}_p)$, $kP = 0_{\mathbb{F}_p}$

Si $n = p_1 p_2 ... p_r$, on peut projeter les coordonnées d'un élement P_1 de $E(\mathbb{Z}/n\mathbb{Z})$ sur un élement P_2 de $E_{\overline{a},\overline{b}}(\mathbb{F}_{p_i})$ en réduisant ses coordonnées modulo p_i. On pourrait penser que similairement à l'anneau $(\mathbb{Z}/n\mathbb{Z})$, il existe un isomorphisme entre $E(\mathbb{Z}/n\mathbb{Z})$ et $E(\mathbb{F}_{p_1}) \times E(\mathbb{F}_{p_2}) \times ... \times E(\mathbb{F}_{p_r})$. Cependant ce n'est pas le cas.

Lemme 3.2.2. *Soit* $n = p_1 p_2 ... p_r$*, avec les* p_i *deux à deux distincts,*$r > 2$*, alors l'application*

$$\phi\colon \begin{cases} E(\mathbb{Z}/n\mathbb{Z}) \longrightarrow E(\mathbb{F}_{p_1}) \times E(\mathbb{F}_{p_2}) \times ... \times E(\mathbb{F}_{p_r}) \\ P \longmapsto (P_{p_1}, P_{p_2}, ..., P_{p_r}) \end{cases} \tag{14}$$

où P_{p_i} *est la projection de* P *sur* $E(\mathbb{F}_{p_i})$*, pour* $0 < i \leq r$ *n'est pas surjective.*

En particulier si $n = pq$ et $Q \in E(\mathbb{F}_q) \neq 0_{E(\mathbb{F}_q)}$, il n'y a pas de correspondance entre l'élement $(0_{E(\mathbb{F}_p)}, Q)$ et un point de $E(\mathbb{Z}/n\mathbb{Z})$. Si $n = p_1 p_2 ... p_r$, les élements $(P_1, P_2, ..., P_r)$ où au moins un P_i est nul et où les P_i sont non tous nuls sont en fait les seuls élements qui ne sont pas atteints.

Theorème 3.2.3. *Soit* $n = p_1 p_2 ... p_r$*,* $P \in E(\mathbb{Z}/n\mathbb{Z}), k \in \mathbb{Z}$*,* P_{p_i} *la projection sur* $E(\mathbb{F}_{p_i})$ *pour* $0 < i \leq r$*. Si il existe* j, m *tels que*

$$k * P_{p_j} = 0_{E(\mathbb{F}_{p_j})} \text{ et } k * P_{p_m} = Q \neq 0_{E(\mathbb{F}_{p_m})} \tag{15}$$

alors on ne peut pas calculer $k * P$ *dans* $E(\mathbb{Z}/n\mathbb{Z})$*.*

Ce théorème donne que si on additionne le point P un certain nombre de fois dans $E(\mathbb{Z}/n\mathbb{Z})$ de manière à ce qu'à un moment, les deux conditions du théorème soient vérifiées, alors l'addition sur $E(\mathbb{Z}/n\mathbb{Z})$ échoue, à cause du lemme précédent. Nous allons voir pourquoi cela nous permet d'exhiber un facteur de n.

4 Algorithme ECM

4.1 Principe de l'algorithme

Nous avons vu précédemment la méthode pour additionner deux points P et Q sur une courbe elliptique. Cette méthode fait intervenir une division lors du calcul de la pente :

$$\lambda = (y_P - y_Q)/(x_P - x_Q) \tag{16}$$

Pour pouvoir diviser par $x_P - x_Q$, il faut que ce nombre soit inversible dans $\mathbb{Z}/n\mathbb{Z}$, et donc calculer $pgcd(x_P - x_Q, n)$.
Si l'inversion est impossible, ce pgcd sera un diviseur non trivial de n. L'algorithme consiste donc à additionner un même point sur une courbe elliptique aléatoire, en espérant que l'ordre de ce point dans un des corps $E(\mathbb{F}_{p_i})$ soit assez petit pour pouvoir obtenir une coordonnée nulle dans

$E(\mathbb{F}_{p_1}) \times E(\mathbb{F}_{p_2}) \times ... \times E(\mathbb{F}_{p_r})$. Ainsi l'addition ne sera pas possible dans $E(\mathbb{Z}/n\mathbb{Z})$, et on pourra trouver un facteur de n grace à un calcul de pgcd.

Voici l'algorithme sur une simple courbe. Comme pour l'algorithme (p-1), on doit choisir une borne B de friabilité. Si l'algorithme échoue on peut augmenter cette borne ou passer à une autre courbe ou les deux.

Algorithm 2 Algorithme ECM pour une courbe elliptique

$x_0 \leftarrow Rand(0, n-1)$ {On initialise notre courbe elliptique}
$y_0 \leftarrow Rand(0, n-1)$
$a \leftarrow Rand(0, n-1)$
$b \leftarrow y_0^2 - x_0^3 - ax_0$
$g \leftarrow pgcd(4a^3 + 27b^2, n)$
if $1 < g < n$ **then**
  return g
else if $g == n$ **then**
  return 0
end if
$E \leftarrow E_{a,b}(\mathbb{Z}/n\mathbb{Z})$
$A \leftarrow (x_0, y_0)$
for all $k \leq B$, k premier **do**
  $a_i \leftarrow$ le plus grand entier tel que $k^{a_i} < B$
  if Calculer $k^{a_i}A$ n'échoue pas **then**
    $A \leftarrow k^{a_i}A$
  else
    return Facteur non trivial de n {Trouvé lors de l'essai de la multiplication}
  end if
end for
return échec de l'algorithme

Dans la méthode p-1 de Pollard, on espère que p-1 soit friable pour réussir a factoriser le nombre. Pour ECM, c'est $\#E(\mathbb{F}_p)$ qui doit être friable. La méthode ECM apporte cependant un gros avantage, qui est que si $\#E(\mathbb{F}_p)$ n'est pas friable, on peut réessayer d'appliquer la méthode sur une autre courbe, pour avoir de nouveaux ensembles $E(\mathbb{F}_{p_i})$, dont le cardinal serait peut être friable, ce qui ne peut pas être fait dans la méthode (p-1).
L'algorithme présenté ci-dessus prend une courbe et additionne un point dessus un certain nombre de fois jusqu'à ce que l'addition échoue. Pour que l'algorithme soit vraiment meilleur que la méthode (p-1) il faut en cas d'échec, retourner au début et recalculer une courbe pour effectuer l'algorithme à nouveau. Nous discuterons des chances de tirer une courbe susceptible de marcher plus tard dans le document.
Le calcul de $k * P$ peut être fait d'une autre manière, on pourrait aussi prendre $B!$ ou $ppcm(1, ..., B)$.

Exemple 4.1.1. *Si l'on prend*

$$n = 455839,\ E : y^2 = x^3 + 5x - 5,\ et\ P = (1, 1) \tag{17}$$

On choisit $B = 10$ *et on calcule* $10! * P$.
$2!P = 2P$, *on doit calculer*

$$\lambda = (3x_p^2 + a)/(2y_p) = (3 * 1^2 + 5) * 2^{-}1 = 8/2 = 4 \tag{18}$$

Le calcul est possible car $pgcd(2, n) = 1$ *et on en déduit d'après les regles de l'addition que* $2P = (14, -53)$.
En remarquant que $3 * 2P = 4P + 2P$, *on calcule* $4P$ *de la même manière. Cette fois-ci* $\lambda = 106$ *qui est inversible modulo n, et on peut calculer* $4P$. *Similairement on peut calculer* $5!P$, $6!P$, $7!P$, *mais calculer* $8!P$ *demande de diviser par 599 i.e d'inverser 599. Or il n'est pas inversible car il divise n. Ainsi on trouve lors du calcul de* $8!P$ *un diviseur de n.*

4.2 Probabilités de succès et complexité

Dans cette section nous allons nous intéresser à la complexité et à l'efficacité d'ECM. En effet puisque l'algorithme repose sur la B-friabilité de $\#E(\mathbb{F}_p)$, il est naturel de s'interroger sur la quantité de courbes qui fournissent des ensembles qui ont en effet un cardinal B-friable. D'après le Théorème de Hasse (3.2.1), tous nos cardinaux sont dans l'intervalle

$$[p+1-2\sqrt{p}; p+1+2\sqrt{p}] \tag{19}$$

Theorème 4.2.1. *Pour tout $m \in [p+1-2\sqrt{p}; p+1+2\sqrt{p}]$, il existe une courbe elliptique tel que*

$$\#E_{a,b}(\mathbb{F}_p) = m \tag{20}$$

Ce théorème est assez rassurant. En effet il nous dit que tous les entiers dans l'intervalle $[p+1-2\sqrt{p}; p+1+2\sqrt{p}]$ sont des cardinaux de courbes elliptiques sur $\mathbb{F}_p$, et donc que pour une borne de friabilité bien choisie on peut toujours trouver un cardinal friable.

Propriété 4.2.2. *(cf 2, prop 2.7)*
Il existe une constante c tel que soit $n, B \in \mathbb{Z}_{>1}$ tel que n a au moins deux diviseurs premiers supérieurs à 3, et soit

$$u = \#\{s \in \mathbb{Z}, |s-(p+1)| < \sqrt{p}, \forall p | s, p < B\} \tag{21}$$

alors le nombre N de triplets $(a,x,y) \in (\mathbb{Z}/n\mathbb{Z})^3$ pour lesquels l'algorithme ECM énoncé précedemment trouve un diviseur non trivial de n satisfait :

$$\frac{N}{n^3} > \frac{c}{log(p)} * \frac{u-2}{2[\sqrt{p}]+1} \tag{22}$$

Ici u représente le nombre d'entiers dans l'intervalle $[p+1-\sqrt{p}, p+1+\sqrt{p}]$ qui sont B-friables. La propriété énonce que la probabilité d'avoir une courbe $y^2 = x^3+ax+b$ qui permettrait de trouver un diviseur de n est à comparable à la probabilité qu'un entier dans $[p+1-\sqrt{p}, p+1+\sqrt{p}$ soit B-friable. On appelera $f(w)$ cette dernière probabilité, ainsi

$$f(w) = \frac{u}{2[\sqrt{p}]+1} \tag{23}$$

De plus on conservera pour la suite la définition de u.

Corollaire 4.2.3. *(cf 2, prop 2.8)*
Soit $n, v \in \mathbb{Z}_{>1}$ tel que n a au moins deux diviseurs premiers supérieurs à 3, et tel p le plus petit nombre premier divisant n soit inférieur à v et soit h le nombre de courbes elliptiques que l'on est prêt à utiliser pour trouver un facteur de n. Alors la probabilité de trouver un facteur de n est d'au moins

$$1 - c^{-hf(w)/log(v)} \tag{24}$$

On voit bien que plus on fait d'essais, plus h augmente et plus la probabilité de réussite augmente. Nous allons maintenant nous intéresser à la complexité de l'algorithme pour une courbe. Pour cela nous avons besoin du cout d'une addition sur la courbe, et du cout d'une multiplication.

Propriété 4.2.4. *Le cout d'une addition sur la courbe est en $O(log^2 n)$*

En effet la seule opération non négligeable en temps est l'inversion modulo n, qui peut s'effectuer a l'aide de l'algorithme d'euclide étendu, qui a une complexité en $O(log^2 n)$.
Pour l'addition, comme vu dans l'exemple, Pour calculer $6P$, on n'additionne pas 6 fois P a lui même. Une manière simple et efficace de calculer un produit a l'aide d'addition est d'utiliser sa représentation binaire. Par exemple, $27P = 1P+2P+8P+16P$, et $16P = 8P+8P$, $8P = 4P+4P$, $4P = 2P+2P$. De plus même si il y a moins d'additions, si la multiplication doit échouer, elle échouera quel que soit la chaine d'addition utilisée.

Propriété 4.2.5. *Le cout d'une multiplication d'un point P par $k \in \mathbb{Z}$ est en $O(log(k))$*

La preuve vient du fait que la taille de la représentation binaire d'un entier est en $O(log(k))$.
Pour estimer la complexité de l'algorithme ECM nous considérons que

$$k = \prod_{p \in \pi(B)} p^{a_i}, \tag{25}$$

où $\pi(B)$ est l'ensemble des nombres premiers inférieurs à B, et a_i est la plus grande puissance tel que $p^{a_i} \leq B$.
Pour un tel k, le calcul de kP s'effectue en

$$O(\frac{B}{log(B)} log(B)) \tag{26}$$

opérations, donc $O(B)$ opérations.En effet on a à peu pres $B/log(B)$ nombres premiers dans le produit, et le deuxième $log(B)$ est là pour l'exponentiation. On peut maintenant estimer la complexité de l'algorithme en fonction du nombre de courbes que l'on veut utiliser. On déduit des propriétés précédentes que la complexité de l'algorithme est en

$$O(hBlog(B)log^2(n)) \tag{27}$$

Des calculs plus compliqués permettent de trouver h et une valeur optimale pour la borne B afin de trouver précisément la bonne complexité. Il est conjecturé que l'algorithme a une complexité en

$$O(L(p)^{\sqrt{2}+o(1)} M(log(n))) \tag{28}$$

, ou $L(x) = e^{\sqrt{log(x)log(log(x))}}$ et $M(log(n))$ est le coût d'une addition sur une courbe.
La fonction $L(x)$ ne sort pas de nulle part. En effet d'après un théorème de Canfield, Erdös et Pomerance, la probabilité qu'un nombre aléatoire inférieur à x ait tous ses facteurs inférieurs à $L(x)^\alpha$ est

$$L(x)^{-1+2\alpha+o(1)}, \tag{29}$$

pour $x \to \infty$.

5 Améliorations

Nous allons évoquer les diverses améliorations que l'on peut apporter a l'algorithme ECM.

5.1 Deuxième phase

Soit n un nombre entier et p le plus petit facteur premier de n. Supposons que $\#E_{a,b}(\mathbb{F}_p)$ n'est pas B-friable pour le B que l'on a choisi et que l'algorithme de base a donc échoué à trouvé un facteur. Alors peut être que

$$\#E_{a,b}(\mathbb{F}_p) = q * \prod_{p_i \leq B} p_i^{a_i}, \tag{30}$$

c'est à dire qu'il ne manque qu'un seul facteur premier supérieur à B. Bien que l'on pourrait multiplier le point actuel par tous les nombres premiers entre B et q, il y a une méthode plus efficace.
En effet on peut utiliser le point

$$Q = [\prod_{p_i \leq B} p_i^{a_i}]P \tag{31}$$

qui est le point qui a "survécu" à l'algorithme classique, et regarder les points

$$[q_0]Q, [q_0 + \delta_0]Q, [q_0 + \delta_0 + \delta_1]Q, [q_0 + \delta_0 + \delta_1 + \delta_2]Q..., \tag{32}$$

où q_0 est le nombre premier le plus proche de B, et δ_i sont les différences entre deux nombres premiers consecutifs après q_0.L'idée est qu'on peut stocker les points $R_i = \delta_i Q$ et calculer rapidement les

multiples. Le gain en complexité est conséquent car multiplier un point par q requiert $O(ln(q))$ opérations tandis-que additionner un R_i précalculé est une opération.
Cette méthode induit une deuxième phase "B_2", pour laquelle, comme pour la première phase "B_1" on doit définir une borne. La borne B que l'on va appeler B_1 etait une borne de friabilité, tandis-que la borne B_2 indique une borne supérieure au nombre premier q restant. En pratique on prend $B_2 = 100B_1$
On peut noter qu'une phase B_2 existe aussi pour la méthode (p-1) de Pollard, mais ne sera pas abordée ici.
Beaucoup d'autres améliorations existent. Par exemple, d'autres formes de représentation de courbes elliptiques (comme la forme de Montgomery) permettent de représenter un point (x, y, z) d'une courbe et l'additionner sans jamais calculer la coordonnée y. D'autres construisent leurs courbes de manière à ce que le cardinal $\#E_{\overline{a},\overline{b}}(\mathbb{F}_p)$ soit d'office divisible par 12 ou 6, et d'autres utilisent des méthodes d'addition qui évitent le calcul d'un inverse modulo n pour réduire le temps d'exécution.

Dans la pratique les resultats de l'algorithme ne sont pas décevants du tout et il est assez simple à programmer, d'autant plus qu'il est parallélisable, car on peut regarder plusieurs courbes indépendamment les unes des autres. Nous avons programmé l'algorithme en C et une partie en Rust, en implémentant la phase $B1$ et la phase $B2$. L'algorithme factorise des entiers produits de deux nombres premiers chacun sur 50-60 bits en quelques secondes avec 8 coeurs et un bon processeur. Sur une machine bien plus puissante (128 coeurs) les produits d'entiers premiers de 80bits se factorisent en quelques secondes. L'algorithme a été programmé sans les améliorations évoquées précedemment à part la phase $B2$.

6 Références bibliographiques

1 primality using elliptic curves *W. Bosma*

2 Factoring integers with elliptic curves *H.W.Lenstra,Jr.*

3 Prime numbers a computational perspective *Richard Crandall, Carl Pomerance*

4 `https://en.wikipedia.org/wiki/Lenstra_elliptic-curve_factorization`

5 Some Integer Factorization Algorithms using Elliptic Curves *Richard P. Brent*

6 20 Years of ECM *Paul Zimmermann, Bruce Dodson*

7 Group Law for elliptic Curves(Based on Cassels' Lectures on Elliptic curves, Chapter 7)