Compter les points sur une courbe elliptique

Jérémie Coulaud

29 janvier 2019

Table des matières

1 Introduction aux courbes elliptiques

La cryptographie sur courbe elliptique introduite en 1985 à révolutionné la cryptographie à clé publique permettant une alternative efficace aux classique RSA et Diffie Hellman. Les courbes elliptiques sont énormément utilisé pour les signatures (ECDSA), notamment pour le Bitcoin, et supporté dans la majorité des applications TLS, SSH. On se contentera de donner une explication succincte de ce qu'est une courbe elliptique ainsi que différents propriétés associées.

Définition 1. *On définit une courbe elliptique sur un corps K dans un plan par une équation de Weierstrass de la forme :*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Les coefficient $a_{1 \leq i \leq 6}$ sont des éléments du corps K.

Dans le cas ou le corps K a une caractéristique différent de 2 ou 3 on peut via des changements de variable se ramener à l'équation courte suivante :

$$y^2 = x^3 + ax + b$$

Définition 2. *Soit $E : y^2 = x^3 + ax + b$ une courbe sur K on pose :*

$$\Delta = -16 * (4a^3 + 27b^2) \quad et \quad j(E) = \frac{(-48a)^3}{\Delta}$$

Δ est appellé le Discriminant de E, et $j(E)$ son j-invariant. La courbe E est une courbe elliptique si et seulement si $\Delta \neq 0$.

Définition 3. *L'ensemble des points de la courbe E est noté $E(K)$ avec :*

$$E(K) = \left\{(x, y) \in \mathbb{K} \,|\, y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 = 0_E\right\} \cup O$$

Où O est le point à l'infini

On peut définir une loi de groupe abélien $\oplus$ sur $E(K)$ de neutre le point à l'infini O. On a pour tout $P = (x, y)$ dans $E(K)$ donnée sous forme réduite :

(1) $O \oplus (x, y) = (x, y) \oplus O = (x, y)$
(2) $\ominus(x, y) = (x, -y)$
(3) Soit $P, Q, R \in E(K)$, si ces trois points sont alignés alors $P \oplus Q \oplus R = 0$

Maintenant qu'on a donnée une structure de groupe abélien à $E(K)$ on peut donner des formules explicites d'additions de points sur la courbe.

Proposition 1. *Soit $P = (x_1, y_1), Q = (x_2, y_2)$.*

$$P \oplus Q = (x_3, y_3) = (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3 - y_1))$$

En posant :

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & si \quad P \neq Q \\ \frac{3x^2 + a}{2y_1} & si \quad P = Q \end{cases} \tag{1}$$

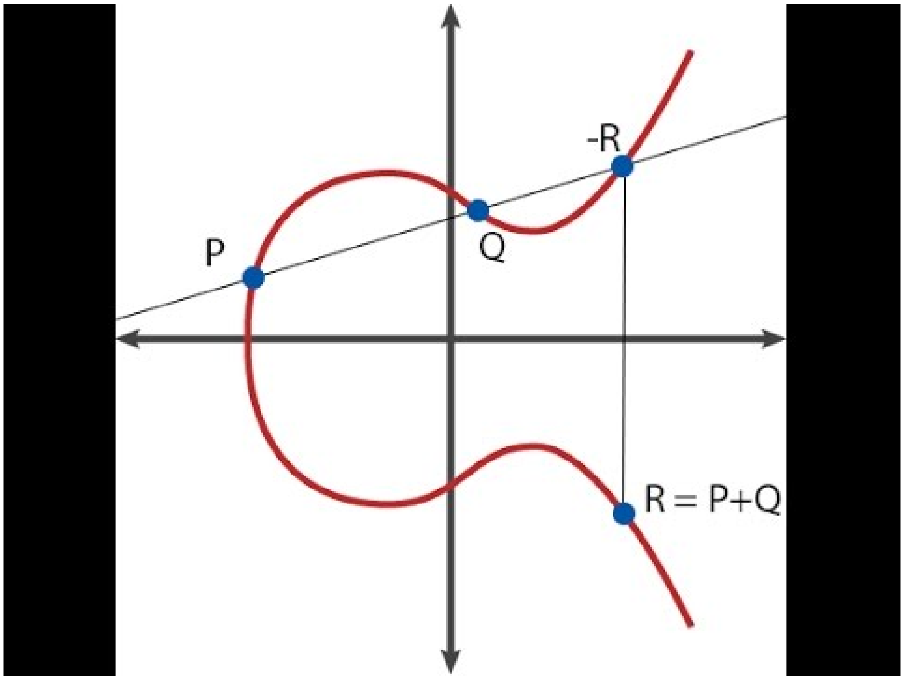

FIGURE 1 – vision graphique de l'addition de deux points sur la courbe

Maintenant que l'on dispose de formule d'additions pour deux points sur une courbe elliptique on peut donner un sens à la multiplication scalaire d'un point comme $kP = \underbrace{P + \ldots + P}_{k \text{ fois}}$. On va noter cette multiplication scalaire par :

$$\begin{array}{cccc} [l]_E & : & E(\mathbb{K}) & \to & E(\mathbb{K}) \\ & & P & \mapsto & lP \end{array}$$

Ce qui nous permet de définir l'ensemble des points de l-torsion comme le noyau de $[l]$. On note $E[l]$ cet ensemble.

$$E[l] = \left\{P \in E(\overline{\mathbb{K}}) \,|\, [l]P = 0_E\right\}$$

2 Compter les points sur une courbe

On va considérer dans la suite que la caractéristique du corps utilisée pour définir nos courbes elliptiques est plus grande que 3. On peut donc écrire notre courbe elliptique sur $\mathbb{F}_p$ sous leur forme réduite $y^2 = x^3 + ax + b$

2.1 Algorithme naif

On note $E : y^2 = f(x)$, compter les points de E revient donc pour chaque valeur de $x \in \mathbb{F}_p$ à regarder si $f(x)$ est un carré modulo p. On calcule donc le symbole de Legendre de $f(x)$, on a les cas suivant :

— $\left(\dfrac{f(x)}{p}\right) = -1$, $f(x)$ n'est pas un carré modulo p, on ne trouve aucun point appartenant à la courbe.

— $\left(\dfrac{f(x)}{p}\right) = 0$, $f(x)$ est divisible par p, on trouve 1 point sur la courbe.

— $\left(\dfrac{f(x)}{p}\right) = 1$, $f(x)$ est un carré modulo p, on trouve 2 points sur la courbe.

Au final en considérant le point à l'infini on peut calculer le nombre de points de E :

$$\#E(\mathbb{F}_p) = 1 + \sum_{x \in \mathbb{F}_p} \left(\left(\frac{f(x)}{p}\right) + 1\right)$$

Soit :

$$\#E(\mathbb{F}_p) = 1 + p + \sum_{x \in \mathbb{F}_p} \left(\frac{f(x)}{p} \right) \tag{2}$$

La complexité est en la taille de p. Cette méthode est pratique quand p est petit mais devient impraticable s'il est trop grand.

2.2 Shanks

Il s'agit d'un algorithme Baby steps-giant steps de complexité exponentielle.

2.3 Schoof

Soit E une courbe elliptique défini sur $\mathbb{F}_p$ avec p premier > 3 sous sa forme réduite

$$E : y^2 = x^3 + ax + b$$

On rappelle le théorème de Hasse-Weil :

Théorème 1. *$\#E(\mathbb{F}_p) = p + 1 - t$ avec $|t| \leq 2\sqrt{p}$ trace de l'endomorphisme de Frobenius de E.*

Pour trouver le nombre de points de E il faut donc déterminer t. L'idée de Schoof est de calculer t modulo de petits nombres premiers puis d'utiliser le théorème des restes chinois.

Avant de développer l'algorithme il est nécessaire de donner d'autres définitions.

Définition 4 (Frobenius)**.** *Soit E une courbe elliptique défini sur $\mathbb{F}_p$, l'endomorphisme de Frobenius est défini par*

$$\begin{array}{cccc} \phi_p & : & E(\mathbb{F}_p) & \rightarrow & E(\mathbb{F}_p) \\ & & (x, y) & \mapsto & (x^p, y^p) \end{array}$$

On peut définir le polynôme caractéristique de cet endomorphisme par $\phi_p^2 - t\phi_p + p = 0$, cette relation reste vrai sur les points de l-torsion. Ainsi nous avons :

$$\phi_p^2(P) + [p_l]P = [t_l]\phi_p(P) \quad \forall P \in E[l] \tag{3}$$

Avec $t_l \equiv t \pmod{l}$, $p_l \equiv p \pmod{l}$ et $0 \leq t_l, p_l \leq l$. Il faut aussi introduire les polynôme de division d'une courbe elliptiques E. On appelle $f_n(X)$ le n-ième polynôme de divisons de E.

Définition 5. *Soit une courbe elliptique $E : y^2 = x^3 + ax + b$ défini sur $\mathbb{K}$. On définit $f_n(X)$ sur $\mathbb{Z}[x]$ de manière récursive :*

$$\begin{aligned} f_0(X) &= 0 \\ f_1(X) &= 1 \\ f_2(X) &= 1 \\ f_3(X) &= 3X^4 + 6aX^2 + 12bX - a^2 \\ f_4(X) &= 2X^6 + 10aX^4 + 40bX^3 - 10a^2X^2 - 8abX - 2(a^3 + 8b^2) \end{aligned}$$

On pose $F(X) = 4X^3 + 4aX + 4b$, et on a :

$$\left\{ \begin{array}{ll} f_{2n} & = f_n(f_{n+2}f_{n-1}^2 - f_{n-2}f_{n+1}^2) \\ f_{2n+1} & = \left\{ \begin{array}{ll} F^2 f_{n+2}f_n^3 - f_{n-1}f_{n+1}^3 & si \quad n \text{ est pair} \\ f_{n+2}f_n^3 - f_{n-1}f_{n+1}^3 F^2 & si \quad n \text{ est impair} \end{array} \right. \end{array} \right. \tag{4}$$

Ces polynômes sont de degrés au plus $\frac{(n^2-1)}{2}$ si n est pair, ou bien au plus $\frac{(n^2-2)}{2}$ si n est impair.

Démonstration. Preuve du degré de f_n ? Avec n premier on a $E[n] \simeq \frac{\mathbb{Z}}{n\mathbb{Z}} \times \frac{\mathbb{Z}}{n\mathbb{Z}}$, soit $n^2 - 1$ points de n-torsion ? on doit les compter 2fois donc j'imagine. Sinon la démo doit découler toute seule en utilisant les formules de recurrence mais un peu plus pénible a écrire $\square$

On peut utiliser les polynôme de division pour calculer la multiplication scalaire d'un point de la courbe E. On a les formules suivantes :

Théorème 2. *Soit E une courbe elliptique défini sur $\mathbb{K}$, un point P sur cette courbe et $m \in \mathbb{N}^*$.*

$$[m]P = \begin{cases} O_E & si \quad P \in E[m] \\ \left(\frac{\phi_m(X,Y)}{\psi_m^2(X,Y)}, \frac{\omega_m(X,Y)}{\psi_m^3(X,Y)} \right) & sinon \end{cases} \tag{5}$$

En posant :

$$\psi_m = \begin{cases} 2Yf_m & si\ m\ est\ pair \\ f_m & sinon \end{cases}$$

et

$$\begin{cases} \phi_m & = X\psi_m^2 - \psi_{m-1}\psi_{m+1} \\ \psi_m\omega_m & = \psi_{2m} \end{cases}$$

On peut aussi réécrire $[m]P$ sous cette forme :

$$[m]P = \begin{cases} O_E & si \quad P \in E[m] \\ \left(X - \frac{\psi_{m-1}(X,Y)\psi_{m+1}(X,Y)}{\psi_m^2(X,Y)}, \frac{\psi_{2m}(X,Y)}{\psi_m^4(X,Y)} \right) & sinon \end{cases} \tag{6}$$

Démonstration. On veut démontrer 6.
On note $[m]P = (x_1, y_1)$ On a alors :

$$x_1 = \frac{\phi_m}{\psi_m^2} = \frac{X\psi_m^2 - \psi_{m-1}\psi_{m+1}}{\psi_m^2} = X - \frac{\psi_{m-1}\psi_{m+1}}{\psi_m^2}$$

et

$$y_1 = \frac{\omega_m}{\psi_m^3} = \frac{\psi_{2m}}{\psi_m}\frac{1}{\psi_m^3} = \frac{\psi_{2m}}{\psi_m^4}$$

□

On peut ainsi exprimer $[m]P$ comme un polynôme en X, Y. Mais ce n'est pas la seule particularité de ces polynôme utile pour notre algorithme. En effet $P = (x_1, y_1)$ est un point de l-torsion si et seulement si x_1 est une racine du l-ième polynôme de division f_l. De plus P est sur la courbe E. Les points de l-torsion sont donc solution du système d'équation :

$$E(X,Y) = Y^2 - X^3 - aX - b = 0, \quad f_l(X) = 0 \tag{7}$$

L'équation 3 peut donc se réécrire en utilisant les points de l-torsion. On va maintenant faire des calcul dans l'anneau $\mathbb{W} = \frac{\mathbb{F}_l[X,Y]}{(f_l(X),E(X,Y))}$.L'idée de l'algorithme de Schoof est donc de tester pour des valeurs $\tau_l \in \{0, \ldots, l-1\}$ si l'équation suivante est vrai :

$$(X^{p^2}, Y^{p^2}) + [p_l](X,Y) = [\tau_l](X^p, Y^p) \tag{8}$$

L'unique solution que l'on trouve est t_l.On répète l'opération pour d'autres l premiers jusqu'à avoir assez de t_l pour appliquer le théorème des restes chinois et retrouver la valeur de t.
On va maintenant détailler l'algorithme étape par étape.

2.3.1 Choix de l'ensemble de premiers

L'idée de Schoof est de calculer la trace de l'endomorphisme de Frobenius modulo de petits premiers. Il faut choisir un ensemble $S = (l_1, \ldots l_n)$ tel que $\prod l_i > 4\sqrt{p}$. En effet on rappelle que $|t| \leq 2\sqrt{p}$, on veut s'assurer de bien pouvoir reconstruire notre solution quand on va utiliser les restes chinois. Pour construire S on va juste ajouter des nombres premiers à une liste en utlisant la fonction `next_prime` de SAGE tant que le produit des éléments de S est plus petit que $4\sqrt{p}$.

2.3.2 Calcul des polynome de division

Il existe dans SAGE une fonction permettant de calculer les polynôme de division, `polynomial_polynomial(n)` renvoyant le n-ième polynome de division. Mais comme au cours de l'algorithme on doit souvent utiliser différents polynôme de division et de plus ils sont calculé via des formules de récurrence. Il ne me paraissait pas forcement très judicieux d'utiliser la fonction de sage qui allait de toute façon très certainement à chaque appel devoir recalculer via les formules de récurrence le polynôme souhaité. L'idée étant de stocker tous ces polynômes dans un dictionnaire, une fois la phase de pré calcul faite on peut donc accéder à tous les polynômes de division que l'on souhaite en temps constant au cours de l'exécution de l'algorithme de Schoof. Et en utilisant la fonction de SAGE j'ai tout de même vérifié via une fonction de test que je trouvais bien les même polynômes qu'avec ma fonction.

2.3.3 Calcul de $(X^{p^2}, Y^{p^2}) + [p_l](X, Y)$

Comme vu en introduction aux courbes elliptiques la somme de deux points appartenant à la courbe dépend de plusieurs cas. Est ce que nos points sont distincts où ont la même coordonnée en x.En effet les formules à appliquer seront différentes dépendamment que l'on soit dans un cas ou l'autre et nous ne disposons pas de méthode pour vérifier rapidement dans quel cas on se situe. Mais il est plus probable d'avoir des coordonnéees différentes, on va donc travailler dans ce cas la. Si l'algorithme trouve un t_l vérifiant l'équation on est dans le bon cas, s'il n'en trouve aucun cela veut dire que l'on se trouve dans l'autre cas.

Cas : $(X^{p^2}, Y^{p^2}) \neq \pm[p_l](X, Y)$ On peut utiliser notre formule usuelle d'addition sur courbe elliptique pour calculer $(x_1, y_1) = (X^{p^2}, Y^{p^2}) + (X_{\bar{p_l}}, Y_{\bar{p_l}})$ en notant $(X_{\bar{p_l}}, Y_{\bar{p_l}}) = [p_l](X, Y)$.
Soit $\lambda = \frac{Y_{\bar{p_l}} - Y^{p^2}}{X_{\bar{p_l} - X^{p^2}}}$, on a alors :

$$(x_1, y_1) = (\lambda^2 - X^{p^2} - X_{\bar{p_l}},\ \lambda(X^{p^2} - x_1) - Y^{p^2}) \tag{9}$$

On va aussi noter $(X_\tau, Y_\tau) = [\tau](X, Y)$. On doit maintenant tester en faisant varier la valeur de τ entre 0 et $l-1$ si $x_1 = X_\tau$. Si c'est le cas on se retrouve avec deux possibilités comme valeur de y_1, soit on a le point soit on a son opposé. On regarde donc si $y_1 = Y_\tau$, on a alors $t_l = \tau$, sinon on a $t_l = -\tau$

Cas : $(X^{p^2}, Y^{p^2}) = \pm[p_l](X, Y)$ La encore deux cas possible. On va d'abord regarder lorsque $(X^{p^2}, Y^{p^2}) = [p_l](X, Y)$.
Soit $P = (X, Y)$, on rappelle l'équation caractéristique de l'endomorphisme Frobenius restreinte aux points de l-torsion :

$$\phi_p^2(P) + [p_l]P = [t_l]\phi_p(P) \pmod{l} \tag{10}$$

On a alors dans notre cas $2[p_l]P = [t_l]\phi_p(P) \pmod{l}$. De plus on a $\phi_p^2 = [p_l]P$. En combinant ces deux égalités on trouve :

$$[t_l^2 p_l]P = [t_l^2]\phi_p^2 = [t_l]\phi([t_l]\phi_p) = [(2p_l)^2]P \pmod{l} \tag{11}$$

Ainsi p_l est un carré modulo l. On pose $p_l = \omega^2 \pmod{l}$. On va maintenant calculer $\omega\phi P$, si $[p_l]P = \omega\phi P$ alors $t_l = 2\omega$ sinon $t_l = -2\omega$.
Si p_l n'est pas un carré modulo l on est dans le cas $(X^{p^2}, Y^{p^2}) = -[p_l](X, Y)$ et on on a alors $t_l = 0$ d'après l'équation caractéristique.

Algorithme On donne une version complète en langage naturel de l'algorithme de schoof :

2.3.4 Complexité

La complexité de l'algorithme de Schoof est en $O(\log q^8)$.

Algorithm 1 Schoof

Require: E une courbe elliptique de la forme $y^2 = x^3 + ax = b$ sur $\mathbb{F}_p$
Ensure: Le nombre de points de E
  Choisir un ensemble de premier S tel que $\prod_{l \in S} l \leq 4\sqrt{p}$
  if $\gcd(x^q - x, x^3 + ax + b)$ **then**
    $t_2 = 0$
  else
    $t_2 = 1$
  end if
  for all $l \in S$ **do**
    Calculer le polynôme de division f_l, on fera les calculs dans l'anneau $\mathbb{F} = \frac{\mathbb{F}_l[X,Y]}{(f_l(X), y^2 - x^3 - ax - b}$)
    On pose $p_l = p \mod (l)$
    Calculer (x^p, y^p), (x^{p^2}, y^{p^2}), $[p_l](x, y) = (x_{p_l}, y_{p_l})$
    if $x^{p^2} \neq x_{p_l}$ **then**
      Calculer $(X, Y) = (x^{p^2}, y^{p^2}) + (x_{p_l}, y_{p_l})$
      for all $1 \leq \tau \leq l - 1$ **do**
        if $X = x_\tau^p$ **then**
          if $Y = y_{p_l}^q$ **then**
            $t_l = \tau$
          else
            $t_l = -\tau$
          end if
        end if
      end for
    else
      if p est un carré modulo l **then**
        blabla
      end if
    end if
  end for