Projet de Cryptologie

Master Mathématique MIC

Second semestre

Algorithme de Schoof

Etudiants :
Berthet Pierre-Augustin
Renard Marc

Professeur :
Brasca Riccardo

Juin 2021

Table des matières

I Introduction

Nous commencerons par présenter l'ensemble des notions importantes sur les courbes elliptiques nécessaires à comprendre l'algorithme de Schoof, suivies d'une présentation de l'algorithme lui même, d'une discussion sur l'implémentation et de résultats sur des exemples, contrôlés à l'aide de fonctions déjà existantes dans **pari/gp** comme **ellcard(E)**, et d'une conclusion sur l'amélioration de Elkies et de l'utilité de l'algorithme dans le cadre de la cryptologie.

I.1 Notations

On utilisera tout au long de ce rapport les notations suivantes :

— $E(C, \mathbb{K})$: la courbe elliptique E d'équation C sur le corps $\mathbb{K}$ de cardinal q,

— $P(x, y)$, un point de la courbe E,

— (x_k, y_k) , les coordonnées de kP

— $\pi(P)$, le morphisme de Fröbenius, de trace t, appliqué à P,

— $\bar{E}[l]$, le groupe des points de l-torsion de la courbe E,

— ψ_l , le l-ième polynôme de division de la courbe E,

— O , le point à l'infini de la courbe E,

— $\bar{t_l}$, l'estimation de t modulo l,

— $\bar{q}$, la valeur de q modulo l.

II Fondement théorique

Lorsque l'on manipule des courbes elliptiques, notamment pour faire des calculs d'addition de points, on tend à utiliser le corps K sur lequel se trouveront les coordonnées de la courbe. Dans notre cas, nous souhaitons pouvoir manipuler un point général noté P afin d'effectuer l'algorithme de Schoof. A cette fin, nous introduisons donc un anneau noté R sur lequel se trouveront les coordonnées de ce point P.

II.1 Anneau R et lois

II.1.1 Structure de l'anneau R

Définition II.1. On appelle R l'anneau suivant :

$$\mathbb{F}_q[x,y]/(y^2 = x^3 + A * x + B)$$

Pour des raisons d'implémentation, on travaillera dans **pari/gp** sur un autre anneau, isomorphe à celui ci :

$$\mathbb{F}_q[x]^2$$

II.1.2 Loi d'addition

Définition II.2. *Loi d'addition sur R.*
Soient $P_1 = a_1(x) + y * a_2(x)$ et $P_2 = b_1(x) + y * b_2(x)$ deux éléments de R. On a :

$$P_1 + P_2 = (a_1(x) + b_1(x)) + y * (a_2(x) + b_2(x))$$

On voit bien ici le parallèle avec $\mathbb{F}_q[x]^2$.

II.1.3 Loi de multiplication

Définition II.3. *Loi de multiplication sur R.*
On reprend les points P_1 et P_2 de la loi d'addition. On a :

$$P_1 * P_2 = (a_1 * b_1 + a_2 * b_2 * (x^3 + A * x + B)) + y * (a_1 * b_2 + a_2 * b_1)$$

Rappel : les a_i , b_i sont des polynômes en x

II.2 Courbe elliptique : lois sur la courbe

II.2.1 Loi d'addition de points

Définition II.4. Soient $P(x_p, y_p)$ et $Q(x_q, y_q)$ deux points d'une courbe elliptique $E(x^3 + A * x + B, \mathbb{K})$. On définit l'addition de P et Q comme suit :

- *Elément neutre* : $P + O = P$,
- *Négation* : $P + -P = O$,
- *Addition* : $P \neq Q$, $P + Q = R$, on a :

$$\nu = (y_q - y_p)/(x_q - x_p),$$

$$x_r = \nu^2 - x_p - x_q,$$

$$y_r = \nu * (x_p - x_r) - y_p.$$

- *Double* : $P = Q$, $2 * P = R$, on a :

$$\nu = (3 * x_p + A)/(2 * y_p),$$

$$x_r = \nu^2 - 2 * x_p,$$

$$y_r = \nu * (x_p - x_r) - y_p.$$

Remarque : Il est très aisé de calculer l'opposé du point P(x,y) : il s'agit du point (x,-y).
On en déduit la propriété suivante :

Propriété II.5. Trois points $P1$, $P2$ et $P3$ d'une courbe elliptique E sont alignés si et seulement si ces trois points sont colinéaires.

Remarque : cela va nous permettre de simplifier considérablement nos tests en évitant de faire des additions elliptiques supplémentaires pour vérifier si $P1 + P2 + P3 = O$.

II.3 Courbes elliptique : propriétés du cardinal de la courbe : la borne de Hasse

II.3.1 Morphisme de Fröbenius

Définition II.6. On appelle morphisme de Fröbenius sur E, noté π, l'endomorphisme de courbe elliptique (ou encore isogénie) suivant :

$$\pi : E \to E$$

$$\pi(x, y) \mapsto Q = (x^q, y^q)$$

$$\pi(O) \mapsto O$$

Cette isogénie a pour équation caractéristique le polynôme suivant :

$$X^2 - tX + q = 0,$$

avec t la trace de l'endomorphisme et q son déterminant, qui se trouve être aussi le cardinal de $\mathbb{K}$.

Remarque : Notez l'équation caractéristique de ce morphisme : elle est particulièrement importante au sein de l'algorithme de Schoof car elle va servir de base à l'estimation de t qui est directement lié à $|E(\mathbb{F}_q)|$.

II.3.2 Théorême

Théorème II.7. *Borne de Hasse. Soit E une courbe elliptique sur $\mathbb{F}_q$. Alors on a l'égalité suivante :*

$$|E(\mathbb{F}_q)| = q + 1 - t, \text{ avec } |t| < 2\sqrt{q},$$

avec q le cardinal de $\mathbb{K}$, et t la trace du morphisme de Fröbenius sur E.

Remarque : Si l'on combine ce que l'on sait du morphisme de Fröbenius et cette borne de Hasse, on peut alors faire une estimation de t sur des modulos de petites tailles et les combiner à l'aide du lemme des restes chinois. La borne de Hasse donnant alors un encadrement de t, on peut en connaître la valeur exacte, sous couvert d'avoir pris suffisamment de modulos pour éviter d'avoir plusieurs réponses possibles au sein de l'intervalle de Hasse.

II.4 Polynomes de division et groupes de l-torsion d'une courbe elliptique

II.4.1 Définitions

Définition II.8. On appelle groupe de l-torsion d'une courbe E, noté $\bar{E}[l]$, l'ensemble des points de cette courbe tels que la multiplication de ces points par le scalaire l donne le point à l'infini O. On définira la multiplication par un scalaire dans la section suivante.

Définition II.9. Le *l-ème* polynôme de division d'une courbe E, noté ψ_l, est définit comme étant le polynôme dont les racines sont les abscisses des points du groupe $\bar{E}[l]$.

Propriété II.10. Les polynômes de division peuvent se calculer récursivement à partir de ψ_1, ψ_2, ψ_3 et ψ_4 selon les formules suivantes :

- $\psi_0 = 0$,
- $\psi_1 = 1$,
- $\psi_2 = 2y$,
- $\psi_3 = 3x^4 + 6Ax^2 + 12Bx - A^2$,
- $\psi_4 = 4y(x^6 + 5Ax^4 + 20Bx^3 - 5A^2x^2 - 4ABx - 8B^2 - A^3)$,
- $\psi_{2m+1} = \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3$ avec $m \geq 2$,
- $\psi_{2m} = \left(\frac{\psi_m}{2y}\right) \cdot (\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2)$ avec $m \geq 3$.

Remarque : cette définition récursive des polynômes de division nous permet de précalculer les polynômes qui nous seront nécessaires dans l'algorithme de Schoof, diminuant ainsi considérablement les temps de calcul au prix d'une place mémoire plus importante (On parle d'ailleurs de compromis temps - mémoire).

II.4.2 Points spéciaux

Définition II.11. On appelle point spécial tout point P de la courbe E tel que $P = (x_0, 0)$.

Propriété II.12. Les points spéciaux sont les points de $\bar{E}[2]$.

Remarque : cette propriété nous permet de gérer le cas $l = 2$ dans l'algorithme de Schoof.

II.4.3 Loi de multiplication par un scalaire

A l'instar de l'exponentiation rapide, il existe un algorithme basé sur l'écriture en base 2 du scalaire. Dans le cadre de notre projet, nous employons une variante basée sur les polynômes de division, que nous admettrons.

Propriété II.13. Soit $P(x_p, y_p)$ un point de la courbe E et k un scalaire positif. Alors le point $R(x_r, y_r) = k * P$ est donné par :

$$x_r = x_p - (\psi_{k-1} * \psi_{k+1})/(\psi_k^2),$$
$$y_r = (\psi_{2k})/(2 * \psi_k^4).$$

Remarque : on peut facilement effectuer une multiplication par un scalaire négatif. Il suffit de faire la multiplication par sa valeur absolue puis de prendre l'opposé du résultat selon la loi d'addition définie précédemment.

III Algorithme

III.1 Algorithme de Schoof

III.1.1 Stratégie générale

On va employer ce que l'on a vu sur le morphisme de Fröbenius et la borne de Hasse. La stratégie ici s'apparente à un algorithme diviser pour régner : plutôt que de calculer t directement avec l'équation caractéristique du morphisme de Fröbenius, on va en faire une estimation sur plusieurs modulos premiers de petites tailles et les combiner à l'aide du lemme des restes chinois.
Le problème est alors le suivant : comment s'assurer que l'équation caractéristique reste juste lorsque l'on travaille modulo l ?
La réponse se trouve dans les points de l-torsion. En effet, si l'on veut calculer $\bar{t_l}$, il faut s'assurer que $t(x, y)$ et $\bar{t_l}(x, y)$ aient le même rôle dans leurs équations respectives. Autrement dit, que $t(x, y)$ et $(\bar{t_l} + k * l)(x, y)$ aient le même rôle. Donc que $k * l(x, y)$ soit O pour tout k. Or, les points $P(x, y)$ qui vérifient cette propriété sont les points de l-torsion de E. Il suffit donc de travailler modulo ψ_l pour estimer $\bar{t_l}$.

III.1.2 Détaillé de l'algorithme

1. On commence par générer la liste S des nombres premiers l sur lesquels on va travailler. Cette liste doit obéir aux critères suivants :

$$p \notin S, \prod_{l \in S} l > 4\sqrt{q}$$

2. On traite tout d'abord le cas $l = 2$. On fait le calcul booléen suivant :

$$\bar{t_l} = (pgcd(x^q - x, x^3 + Ax + B) == 1)$$

3. Pour tous les autres éléments de S, on a les étapes suivantes :
   (a) On calcule l'unique valeur $\bar{q_l} = q$ modulo l, $|\bar{q_l}| < l/2$.
   (b) On calcule $F(x^{q^2}, y^{q^2})$ et $Q(x_{\bar{q_l}}, y_{\bar{q_l}})$
   (c) Si $x^{q^2} \neq x_{\bar{q_l}}$:
      i. Pour t allant de 1 à $\frac{l-1}{2}$
         A. Calculer $T(x^q_{\bar{t_l}}, y^q_{\bar{t_l}})$
         B. Si F, Q et T sont alignés alors $t_l = -t$ et on passe au l suivant. on emploie ici la propriété II.5 dans la rubrique II.2.1.
         C. Si F, Q et $-T$ sont alignés alors $t_l = t$ et on passe au l suivant
         D. Si nous ne sommes dans aucun des deux cas précédents, passer à l'instruction suivante
   (d) Trouver, si elle existe, une racine ω de q modulo l. Si q n'est pas un carré modulo l alors $t_l = 0$ et on peut passer au l suivant. Sinon on passe à l'étape e.
   (e) Calculer (x^q_ω, y^q_ω)
   (f) Si $(x^q_\omega, y^q_\omega) = F$ alors $t_l = 2\omega$
   (g) Si $(x^q_\omega, y^q_\omega) = -F$ alors $t_l = -2\omega$
   (h) Sinon $t_l = 0$
4. Appliquer le lemme des restes chinois sur les t_l pour obtenir $t_N = t$ modulo N où $N = \prod_{l \in S} l$.
5. Tant que $c := 1 + q - t < 1 + q - 2\sqrt{q}$ faire $c \longleftarrow c + N$

III.1.3 Explications supplémentaires

Dans l'étape 1, on choisit pour borne $4\sqrt{q}$. Cela garantit que l'on trouvera une unique valeur de t dans l'intervalle de Hasse avec le lemme des restes chinois. En effet t est encadré par $-2\sqrt{q}$ et $2\sqrt{q}$, ce qui fait bien une étendue de $4\sqrt{q}$.

Dans l'étape 2, on justifie le recours au pgcd comme suit : Dans le cas $l = 2$, on a $y = 0$ car on travaille sur des points de 2-torsion, comme spécifié à la propriété II.12 (rubrique II.4.2). En conséquence, cela revient à vérifier si $x^3 + Ax + B$ admet des racines dans $\mathbb{F}_q$, d'où l'usage du pgcd.

Dans l'étape 3, $x^{q^2} \neq x_{\bar{q}_l} \iff (x^{q^2}, y^{q^2}) \neq \pm\bar{q}(x,y)$
Si F, Q et T sont alignés, cela signifie que $(x^{q^2}, y^{q^2}) + t*(x^q, y^q) + \bar{q}_l(x,y) = O$. Ainsi par comparaison à l'équation caractéristique du Fröbenius on déduit que $t_l = -t$.
Si F, Q et $-T$ sont alignés, cela signifie que $(x^{q^2}, y^{q^2}) - t*(x^q, y^q) + \bar{q}_l(x,y) = O$. Ainsi par comparaison à l'équation caractéristique du Fröbenius, on déduit que $t_l = t$.

Arrivés à l'étape d, nous avons $(x^{q^2}, y^{q^2}) = \pm\bar{q}(x,y)$.
Supposons d'abord que $(x^{q^2}, y^{q^2}) = \bar{q}(x,y)$.On ne peut pas avoir $\bar{q}(x,y) = -\bar{q}(x,y)$, sinon nous aurions $2\bar{q}(x,y) = O$,or $|2\bar{q}| \leq 2*\frac{l-1}{2} = l-1$ et comme nous travaillons sur les points de l-torsion avec l premier, $(l-1)(x,y) \neq O$. Ainsi $(x^{q^2}, y^{q^2}) + \bar{q}(x,y) \neq O$ et donc $t_l \neq 0$. L'équation caractéristique devient alors :
$(x^{q^2}, y^{q^2}) - t_l * (x^q, y^q) + \bar{q}(x,y) = 0 \iff 2\bar{q}(x,y) = t_l * (x^q, y^q) \iff 2\bar{q}P = t_l\pi(P)$.
Or par hypothèse $\pi^2(P) = \bar{q}P$, donc $t_l^2\bar{q}P = t_l^2\pi^2(P) = t_l\pi(t_l\pi(P)) = (2q)^2P$.
On en déduit ainsi que q est un carré modulo l sous cette hypothèse. Si c'est réellement le cas, on écrit $q = \omega^2 \ mod\ l$.
On alors $t_l^2\bar{q}P = (2q)^2P \iff t_l^2P = 4\bar{q}P = (2\omega)^2P$ donc $t_l = \pm 2\omega$.
Si $\omega(P) = \bar{q}P$ alors $\pi^2(P) - t_l\pi(P) + \bar{q}P = 2\omega\pi^2(P) - t_l\pi(P) = O$ ce qui revient pour un point sur la courbe elliptique à $2\omega P = t_l P$ et ainsi $t_l = 2\omega$ et sinon si$\omega(P) = -\bar{q}P$ $t_l = -2\omega$ par le même raisonnement.

Si q n'était un carré, notre hypothèse précédente ($(x^{q^2}, y^{q^2}) = \bar{q}(x,y)$) était fausse et donc $(x^{q^2}, y^{q^2}) = -\bar{q}(x,y)$ et on en déduit directement que $t_l = 0$.

L'étape 4 consiste simplement à appliquer le théorème des reste chinois pour obtenir t_N modulo N où $N = \prod_{l \in S} l$.

Enfin, par construction de S, il existe un unique t dans l'intervalle de Hasse tel que $t = t_N$ modulo N. Une fois ce t trouvé, nous avons $|E(\mathbb{F}_q)| = 1 + q - t$.

IV Implémentation

IV.1 Calcul de nP à l'aide des polynômes de division

Soit $P(x,y)$ un point de la courbe elliptique d'équation $y^2 = x^3 + Ax + B$ sur $\mathbb{F}_q$.

Une première méthode pour calculer les coordonnées de nP est d'utiliser la méthode *double and add* en utilisant l'addition sur la courbe elliptique. Cette méthode est l'équivalent de l'exponentiation rapide, mais appliquée à la multiplication d'un point par un scalaire. Cependant, cette méthode est un procédé itératif qui à chaque étape nécessitera l'exécution d'additions sur la courbe elliptique.

Pour éviter cela, nous pouvons utiliser la propriété II.13 (rubrique II.4.3) qui permet de calculer nP à l'aide des polynômes de division $\psi_{n-1}, \psi_n, \psi_{n+1}$ et ψ_{2n}. Cette méthode nous permet de calculer les coordonnées de nP sans passer par les additions elliptiques. Cette méthode est de plus compatible avec la représentation de l'anneau R adaptée à **pari gp** car il est facile d'extraire les polynômes en x des résultats de ce calcul.

IV.2 Calcul des polynômes de division au préalable

Durant l'exécution de l'algorithme, les polynômes de division sont utilisés à différentes fins :

1. Pour chaque l de la liste des nombre premiers, nous allons travailler modulo ψ_l,
2. lors de la multiplication par $q_l = q \ mod \ l$ avec $|q_l| < \frac{l}{2}$,
3. lors de la multiplication par $t_l \in [\![1; \frac{l-1}{2}]\!]$.

Si on note L le plus grand des nombres premiers dans la liste, alors :

1. Le polynôme de division de plus haut degré utilisé sera ψ_L,
2. le polynôme de division de plus haut degré potentiellement utilisé sera ψ_{L-1}, c'est le cas où $|q_L| = \frac{L-1}{2}$,
3. le polynôme de division de plus haut degré potentiellement utilisé sera ψ_{L-1}, c'est le cas où $t_L = \frac{L-1}{2}$.

Nous pouvons donc en déduire que le polynôme de division de plus haut degré qui sera utilisé sera ψ_L.
Ainsi, juste après la génération de la liste des nombres premiers, nous pouvons générer une map M (**M = Map()**) qui contiendra les couples (l, ψ_l)). Il suffira par la suite d'accéder à cette map pour récupérer le polynôme de division désiré (**psi = mapget(M,l)**).

L'avantage de cette map est qu'elle évite de calculer plusieurs fois un même polynôme. Par exemple si 13 est dans la liste des nombres premiers, les premiers tests pour cette valeur de l vont potentiellement nécessiter le calcul de $n * P, \ \forall n \in [\![1; 6]\!]$. Pour calculer $3P$ il faudra utiliser ψ_2, ψ_3, ψ_4 et ψ_6, et pour calculer $4P$ il faudra utiliser ψ_3, ψ_4, ψ_5 et ψ_8. Ceci met en évidence qu'un même polynôme de divison va être utilisé plusieurs fois au cours de l'éxecution de l'algorithme. Le fait de les calculer dès

le départ et de les stocker dans une map permet d'éviter tous les "re-calculs" des polynômes de division qui ont déjà été calculés et utilisés auparavant dans l'algorithme.

Cette méthode pourrait encore être améliorée, car il est possible que cette map contienne des polynômes qui ne seront finalement pas utilisés. Une manière d'éviter cela serait la suivante :
À chaque fois que l'algorithme a besoin d'un polynôme de division, il regarde s'il se trouve déjà dans la map, si c'est le cas il le récupère, et sinon il le calcule et l'ajoute à la map. De cette manière, seuls les polynômes de division qui seront réellement utilisés seront calculés, et aucun ne sera calculé plusieurs fois.

IV.3 Contexte de travail du point de vue de pari/gp

pari/gp est très efficace en ce qui concerne le calcul sur les polynômes à une variable. Cependant, nous avons précédemment identifié les points de la courbe à l'ensemble

$$R^2 = (\mathbb{F}_q[x, y]/(y^2 = x^3 + A * x + B))^2$$

De part la construction de l'anneau R, on remarque que tout élément de R peut s'écrire sous la forme $a(x) + y * b(x)$. On considère alors l'isomorphisme suivant :

$$R \longrightarrow (\mathbb{F}_q[X])^2$$

$$a(x) + y * b(x) \mapsto [a(x), b(x)]$$

Cet isomorphiqme nous permettra de représenter tout élément de R comme un couple de polynômes de $\mathbb{F}_q[X]$ dans **pari/gp**.
Ainsi le point $P(x, y) = (x+0*y, 0+1y) \in R^2$ sera représenté dans notre algorithme par :

$$P([x, 0], [0, 1])$$

V Résultats

On présente ici tout d'abord tout un ensemble de résultats sur les courbes elliptiques d'équation $y^2 = x^3 + Ax + B$ sur $\mathbb{F}_{37}$ [4].

(A mod 37, B mod 37)	(5,0)	(0,9)	(0,6)	(1,12)	(2,2)
$\#(E(\mathbb{F}_{37}))$	26	27	28	29	30
(A mod 37, B mod 37)	(2,8)	(3,6)	(1,13)	(1,18)	(1,8)
$\#(E(\mathbb{F}_{37}))$	31	32	33	34	35
(A mod 37, B mod 37)	(1,0)	(0,5)	(1,5)	(0,3)	(1,2)
$\#(E(\mathbb{F}_{37}))$	36	37	38	39	40
(A mod 37, B mod 37)	(1,16)	(1,9)	(2,9)	(1,7)	(2,14)
$\#(E(\mathbb{F}_{37}))$	41	42	43	44	45
(A mod 37, B mod 37)	(1,11)	(3,15)	(0,1)	(0,2)	(2,0)
$\#(E(\mathbb{F}_{37}))$	46	47	48	49	50

Voici 7 exemples sur des corps finis de type $\mathbb{F}_p$

(A,B)	$\#(E(\mathbb{F}_p))$
(Mod(3,1031),Mod(2,1031))	1018
(Mod(3,32771),Mod(2,32771))	33117
(Mod(1,65537),Mod(5,65537))	65460
(Mod(1,100003),Mod(5,100003))	99707
(Mod(1,131101),Mod(5,131101))	131132
(Mod(2,190027),Mod(6,190027))	190183
(Mod(1,333517),Mod(2,333517))	332640

Et enfin, voici 7 exemples sur des corps finis de type $\mathbb{F}_q$
Dans les exemples ci-dessous, g est un générateur de $\mathbb{F}_q$

q	(A,B)	$\#(E(\mathbb{F}_q))$
25	(g^2,g^6)	27
49	(g^2,g^6)	55
125	(g^{10},g^{15})	108
625	(g^{10},g^{15})	577
$1331 = 11^3$	(g^{10},g^{15})	1274
$28561 = 13^4$	(g^{10},g^{15})	28800
$130321 = 19^4$	(g^{10},g^{15})	130969

VI Améliorations et applications à la Cryptologie

VI.1 Amélioration de Elkies

La liste S des nombres premiers que nous utilisons est composée de nombres premiers consécutifs, différents de la caractéristique de $\mathbb{K}$. Une amélioration, proposée par Elkies, consiste à sélectionner une liste bien spécifique de nombres premiers afin de gagner en temps de calcul. On a la définition suivante :

Définition VI.1. On dit que l est un nombre premier de Elkies si :
$\pi^2 - t * \pi + q = (\pi - \lambda) * (\pi - \overline{\lambda})$ modulo l, avec $\lambda * \overline{\lambda} = q$.

De fait, on a :

$$t_l = \lambda + \overline{\lambda} \text{ modulo } l.$$

et on peut résoudre l'équation caractéristique en la remplaçant par la suivante, moins coûteuse :

$$x' - x_\lambda^q = 0 \text{ modulo } F_l, \text{ où } F_l = \prod_{P \in C, P \neq O}(x - P_X).$$

Remarque : P_X est l'abscisse de P.

VI.2 De l'importance de connaitre le cardinal d'une courbe elliptique

Des algorithmes comme ECDSA ou ECDH reposent sur la difficulté à faire une "division" de points (par opposition à la multiplication de point par un scalaire). En conséquence, on peut utiliser le scalaire k comme clé secrète pour générer kP, avec P un point connu d'une courbe E connue de tous. Il est très difficile de deviner k à partir de kP. Il faudrait alors tenter de calculer $2P$, $3P$ etc... jusqu'à tomber sur kP. Or, les courbes elliptiques ont des structures de groupe cyclique ou de produit de deux groupes cycliques, d'après le Théorême de Cassel 1966 [5]. D'où l'importance de connaître l'ordre de ces groupes, à savoir donc le cardinal de la courbe. En effet, si ce cardinal est très inférieur à k par exemple, on aura un scalaire $u << k$ tel que $uP = kP$, ce qui réduit donc la complexité de l'attaque *brute-force* décrite ici conséquemment (à noter que, si l'attaquant ici n'obtient pas forcément k, il obtient un moyen d'usurper l'identité de l'attaqué).

Références

[1] Gregg Musiker. *Schoof's Algorithm for Counting Points on $E(\mathbb{F}_q)$*. 7 Décembre 2005.

[2] René Schoof. *Counting points on elliptic curves over finite fields*. Journal de Théorie des Nombres de Bordeaux 7 , 219-254 , 1995.

[3] Luca Defeo. *Algorithmique et Programmation en C, 2014-2019*. Master Algèbre Appliquée à la Cryptologie et au Calcul Formel, UVSQ.

[4] Hankerson, Menezes, Vanstone. *Guide to Elliptic Curves Cryptography*. Springer.

[5] Mohammed Krir. *Introduction aux Courbes Elliptiques*. Master Algèbre Appliquée à la Cryptologie et au Calcul Formel, UVSQ.