PROGRAMMATION PARI/GP: L'ALGORITHME RHO DE POLLARD

BILL ALLOMBERT

1. L'algorithme rho de Pollard sur les corps finis

Il s'agit d'implanter une version simple de l'algorithme rho de Pollard pour le logarithme discret dans les corps finis. Soit K un corps fini, a et b deux éléments, il s'agit de trouver e tel que $a = b^e$ (en supposant qu'il existe). L'idée est de trouver un point qui peut s'écrire de deux façon comme produits de puissance de a et b:

$$X = a^{u_1} b^{v_1} \tag{1}$$

$$X = a^{u_2} b^{v_2} \tag{2}$$

(3)

et de résoudre ce système modulo l'ordre de Q.

Pour trouver cela, nous cherchons une collision avec l'algorithme de Floyd.

Nous avons besoin de 4 ingrédients

1.1. **Une fonction "rnd".** Nous avons besoin d'une fonction `rnd` qui prends en entrée un élément de K et retourne soit 0, 1 or 2, qui se comporte de façon pseudo-aléatoire. (Use `a.pol` pour obtenir une représentant de l'élément a de $\mathbb{F}_q$ dans $\mathbb{Z}[X]$).

1.2. **La fonction ρ.** Nous avons besoin d'une fonction `rho` qui prend en entrée un triplet $[X, u, v]$ tel que $X = a^u b^v$, calcule `h=rnd(X)` et retourne:

- si $h = 0$, retourne $[aX, u + 1, v]$
- si $h = 1$, retourne $[bX, u, v + 1]$
- si $h = 2$, retourne $[X^2, 2u, 2v]$

(Le nouveau triplet satisfait encore la condition $X = a^u b^v$.

1.3. **L'algorithme de Floyd.** L'idée est de calculer deux suites de points X_n et $Y_n = X_{2n}$ par récurrence, tel que $X_0 = P$ and $X_{n+1} = \rho(X_n)$ jusqu'à trouver n tel que $X_n = Y_n$, et les coéfficients u_n et v_n tel que $X_n = a^{u_n} b^{v_n}$ correspondants.

1.4. **Le logarithme discret.** En supposant que $u_n \neq u_{2n}$, nous pouvons résoudre l'équation

$$a^{u_n} b^{v_n} == a^{u_{2n}} b^{v_{2n}} \tag{4}$$

pour trouver e.

2. L'algorithme rho de Pollard

sur les courbes elliptique Il s'agit d'implanter une version simple de l'algorithme rho de Pollard pour le logarithme discret sur les courbes elliptiques. Soit E une courbe elliptique, P et Q deux points, il s'agit de trouver e tel que $P = eQ$ (en supposant qu'il existe). L'idée est de trouver un point qui peut s'écrire de deux façon comme combinaison linéaire de P et Q:

$$X = a_1 P + b_1 Q \tag{5}$$

$$X = a_2 P + b_2 Q \tag{6}$$

(7)

et de résoudre ce système modulo l'ordre de Q.

Pour trouver cela, nous cherchons une collision avec l'algorithme de Floyd.

Nous avons besoin de 4 ingrédients

2.1. **Une fonction "rnd".** Nous avons besoin d'une fonction `rnd` qui prends en entrée un point de E eet retourne soit 0, 1 or 2, qui se comporte de façon pseudo-aléatoire. (Use `a.pol` pour obtenir une représentant de l'élément a de $\mathbb{F}_q$ dans $\mathbb{Z}[X]$).

2.2. **La fonction ρ.** Nous avons besoin d'une fonction `rho` qui prend en entrée un triplet $[X, a, b]$ tel que $X = aP + bQ$, calcule `h=rnd(X)` et retourne:

- si $h = 0$, retourne $[X + P, a + 1, b]$
- si $h = 1$, retourne $[X + Q, a, b + 1]$
- si $h = 2$, retourne $[2X, 2a, 2b]$

(Le nouveau triplet satisfait encore la condition $X = aP + bQ$)

2.3. **L'algorithme de Floyd.** L'idée est de calculer deux suites de points X_n et $Y_n = X_{2n}$ par récurrence, tel que $X_0 = P$ and $X_{n+1} = \rho(X_n)$ jusqu'à trouver n tel que $X_n = Y_n$, et les coéfficients a_n et b_n tel que $X_n = a_n P + b_n Q$ correspondants.

2.4. **Le logarithme discret.** En supposant que $a_n \neq a_{2n}$, nous pouvons résoudre l'équation

$$a_n P + b_n Q = a_{2n} P + b_{2n} Q \tag{8}$$

pour trouver e

2.5. **Améliorations.** Nous pouvons aussi nous arrêter si $X_n = -X_{2n}$ et résoudre une équation similaire.