TUTORIEL: COURBES ELLIPTIQUES SUR LES CORPS FINIS EN PARI/GP

BILL ALLOMBERT

Ce document est une introduction au calcul sur les courbes elliptiques sur les corps finis en PARI/GP qui ne nécessite pas de prérequis.

Dans la suite, `?` designe l'invite GP et `\\` les commentaires GP. Une version texte de la liste des commandes à taper (ou copier-coller !) et disponible à `https://pari.math.u-bordeaux.fr/Events/PARI2017c/talks/ecc.txt`

Entrez

```
? \l ecc.log
```

pour activer l'enregistrement de vos entrées pour en garder trace.

1. Corps finis premier

Pour créer un nombre premier aléatoire :

```
? p=randomprime(2^100)
%1 = 792438309994299602682608069491
```

Pour créer un élément de $\mathbb{F}_p$:

```
? a=Mod(2,p)
? a^(p-1) \\ exponentiation
%3 = Mod(1,792438309994299602682608069491)
```

2. Corps finis généraux

Pour construire un polynôme irreductible de degré n sur $\mathbb{F}_p$, uitilisez `ffinit(p,n)`.

```
? P=ffinit(13,2)
%4 = Mod(1,13)*x^2+Mod(1,13)*x+Mod(12,13)
? polisirreducible(P)
%5 = 1
```

Pour construire un element de $\mathbb{F}_{p^n}$ (marche aussi pour $n = 1$) à partir de son polynôme minimal :

```
? a=ffgen(P,'a)
```

Ce qui peut être abrégé en `ffgen(p^n,'a)`

```
? a=ffgen(13^7,'a)
```

Opérations élémentaires

Date: 23 Novembre 2017.

```
? a^10458086 \\ puissance
? fforder(a) \\ ordre d'un élément
? minpoly(a) \\ polynôme minimal
? random(a)  \\ tire au hasard un élément de F_p^n
```

Pour obtenir un generateur de $\mathbb{F}_{p^n}^\times$:

```
? b = ffprimroot(a)
? fforder(a)
? fforder(b)
```

Pour calculer un logarithme discret :

```
? n=fflog(a,b)
? b^n
```

2.1. **Exercice.** Calculez un logarithme dans $\mathbb{F}_{2^{127}}$ et notez combien de temps cela prend. Utilizes `#` pour activer le chronomètre ou `##` pour afficher le temps de calcul de la dernière commande.

3. Courbes elliptiques sur les corps finis

Pour un courbe donnée par un modèle de Weierstrass court : $y^2 = x^3 + a_4x + a_6$:

```
? Es = ellinit([a^4,a^6],a);
```

Pour un modèle de Weierstrass long : $y^2+a_1xy+a_3y = x^3+a_2x^2+a_4 \times x + a_6$:

```
? E = ellinit([a,a^2,a^3,a^4,a^6],a);
```

Fonctions élémentaires :

```
? E.j \\ j-invariant
```

Structure du groupe $E(\mathbb{F}_q)$

```
? ellcard(E) \\ cardinal of E(F_q)
? ellgroup(E) \\ structure of E(F_q)
```

Ci-dessus $[d1, d2]$ signifie $\mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z}$, avec $d_2 \mid d_1$.

```
? G=ellgenerators(E) \\ generators of E(F_q)
```

donne un système minimal de generateurs.

```
? P = random(E) \\ point au hasard sur E(F_q)
? Q = random(E) \\ un autre point sur E(F_q)
? ellisoncurve(E, P) \\ vérifie que le point est sur la courbe
? elladd(E, P, Q)   \\ P+Q sur E
? ellmul(E, P, 100) \\ 100.P sur E
? oP = ellorder(E,P) \\ordre de P
? oQ = ellorder(E,Q) \\ordre de Q
? o = lcm(oP,oQ); \\ P et Q sont dans E(F_q)[o]
? w=ellweilpairing(E,P,Q,o) \\ couplage de Weil de P et Q d'ordre o
? fforder(w)
? nP = ellmul(E, P, random(o));
? n = elllog(E,nP,P)
```

```
? ellmul(E,P,n) == nP
```

4. Application : vitesse du logarithme discret

Nous souhaitons comparer la vitesse du logarithme discret sur $\mathbb{F}_p$ et $E(\mathbb{F}_p)$, pour p de 30, 40 et 45 bits. Nous devons prendre des groupes d'ordre quasi-premier.

```
? until(isprime((p-1)/2), p=randomprime(2^30));p
? g=ffprimroot(ffgen(p))^2; a=g^random(p);
? fflog(a,g)
? ##
? until(isprime(ellcard(E)),E=ellinit([1,random(g)],g));
? G=ellgenerators(E)[1]; P=ellmul(E,G,random(ellcard(E)));
? elllog(E,P,G)
? ##
```

5. Application : L'attaque MOV sur le logarithme discret sur les courbes

Nous construisons des courbes où le couplage de Weil nous permet de réduire le logarithme discret sur la courbe à un logarithme discret dans un corps fini. Il s'agit de l'idée derrière l'attaque MOV de Menezes, Okamoto, et Vanstone.

5.1. **Exemple sur $\mathbb{F}_p$.** We utilisons la courbe $E : y^2 = x^3 + x$ sur $\mathbb{F}_p$ avec $p = 4n^2 + 1$ pour un entier n, qui a la propriété que $E(\mathbb{F}_p)$ est isomorphe à $\mathbb{Z}/2n\mathbb{Z} \times \mathbb{Z}/2n\mathbb{Z}$. Nous choisissons n premier. Pour trouver un entier n acceptable de 50 bits :

```
? until(isprime(p),n=randomprime(2^50);p=1+4*n^2);p
? a=ffgen(p); E=ellinit([1,0],a);
? ellgroup(E)
? [P,Q] = ellgenerators(E); \\ name P and Q the generators.
```

Nous utilisons le couplage de Weil avec le second générateur pour résoudre le logarithme discret dans le sous-groupe engendré par le premier générateur.

```
? e = random(2*n)
? R = ellmul(E,P,e);
? wR = ellweilpairing(E,Q,R,2*n);
? wP = ellweilpairing(E,Q,P,2*n);
? default(parisize,"32M");
? fflog(wR,wP,2*n)
? ##
? elllog(E,R,P,2*n)
? ##
```

5.2. **Exemple sur $\mathbb{F}_{p^2}$.** Nous utilisons la courbe $E : y^2 = x^3 + x$ et un premier $p \equiv 3 \pmod 4$, de sorte que E soit supersingulière, d'ordre $p+1$. Pour calculer le logarithme discret dans $E(\mathbb{F}_p)$, nous le considérons come sous-groupe de $E(\mathbb{F}_{p^2})$ qui est isomorphe à $\mathbb{Z}/(p+1)\mathbb{Z}\times\mathbb{Z}/(p+1)\mathbb{Z}$, pour pouvoir utiliser le couplage de Weil dans $E(\mathbb{F}_{p^2})$. Nous choisissons $(p+1)/4$ premier.

```
? until(p%4==3 && isprime((p+1)/4),p=randomprime(2^52));
? a=ffgen(p^2,'a);
? E=ellinit([1,0],p); \\ E(F_p)
? ellgroup(E)
? [P] = ellgenerators(E)
? E2=ellinit([1,0],a); \\ E(F_p^2)
? [m,m]=ellgroup(E2)
? [P1,Q] = ellgenerators(E2)
? ellweilpairing(E2,P,Q,m);
? e = random(m)
? R = ellmul(E,P,e);
? wR = ellweilpairing(E2,Q,R,m);
? wP = ellweilpairing(E2,Q,P,m);
? fflog(wR,wP,m)
? ##
? elllog(E,R,P,m)
? ##
```

5.3. **Exercices.**

- Faire l'attaque MOV sur une courbe supersingulière sur $\mathbb{F}_{3^{41}}$.
- Écrire une fonction qui calcule la structure du groupe de $E(\mathbb{F}_q)$ el utilisant le couplage de Weil.
- Implanter l'algorithme de Pollard rho pour le calcul du logarithme discret dans $E(\mathbb{F}_q)$.