N^0 d'ordre : 14/2004 – M / MT

République algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté de Mathématiques

Mémoire présenté

Pour l'obtention du Diplôme de Magister en :

Mathématiques

Spécialité : Algèbre et Théorie des Nombres

Par : MOUHOUB FATIHA

Thème

Application des courbes elliptiques

Soutenu le 15 / 07 / 2004 ; devant le jury composé de :

Mr- M.ZITOUNI , Professeur, USTHB.	**Président**
Mr- K.BETINA , Professeur, USTHB.	**Directeur de thèse**
Mr- A.KESSI , Professeur, USTHB.	**Examinateur**
Mr-M.S.REZAOUI, Chargé de cours, USTHB	**Examinateur**

Sommaire :

Introduction :

La théorie des courbes elliptiques est un domaine riche et a donné de nombreux résultats.

Ces courbes sont de la forme :

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

Où les coefficients appartiennent à un corps commutatif K.

L'une des plus importantes propriétés des courbes elliptiques sera étudiée dans le premier chapitre, nous verrons qu'on peut les munir d'une structure de groupe commutatif. On montrera aussi comment calculer le nombre des points d'une courbe elliptique sur le corps fini F_p.

Dans le deuxième chapitre, on propose l'étude de deux applications liées aux courbes elliptiques : tester la primalité d'un nombre et factoriser un nombre positif, composés.

La méthode de factorisation par courbes elliptiques a été proposée par H.W.Lenstra en 1985, elle est basée sur le théorème de Hasse qui donne une approximation de la cardinalité d'une courbe elliptique.

Finalement, dans le chapitre 4 quelques cryptosystèmes classiques seront adaptés pour pouvoir être utilisés sur les courbes elliptiques.

Pour les usages de ces courbes en cryptographie, on utilise la forme réduite : $y^2 = x^3+ax+b$.

Ce chapitre est aussi une application de la théorie des diviseurs qu'on donnera dans le troisième chapitre.

Chapitre 1

Les courbes elliptiques, définition, propriétés

1.Définition des courbes elliptiques :

Définition 1.1: (plan projectif).

Soit K un corps, le plan projectif $P^2(K)$ sur le corps K est le quotient de l'ensemble des points $(a, b, c)\in K-\{(0, 0, 0)\}$ par la relation d'équivalence $\Re$ définie par : $\forall((a, b, c),(a', b', c'))\in(K^3-\{(0, 0, 0)\})^2$

$(a, b, c)\Re(a', b', c') \Leftrightarrow (\exists t\in K-\{0\} ; (a, b, c)=t(a', b', c'))$.

Définition 1.2 : (polynômes homogènes).

Soit K un corps, et $P\in K[X_1,\ldots,X_n]$.

le polynôme P est homogène de degré d si tous ses termes sont de degré d.

Définition 1.3 :

Soit K un corps, une cubique C du plan projectif $P^2(K)$ est l'ensemble des points vérifiant une équation $F(x, y, z)=0$ où $F\in K[x, y, z]$ homogène de degré 3.

Définition 1.4 :

Soit une courbe C dans le plan $P^2(K)$ définie par un polynôme homogène $F(x,y,z)$ et un point $P=[a,b,c]$sur C.

P est un point singulier de la courbe C si :$(\partial F/\partial x(P),\partial F/\partial y(P),\partial F/\partial z(P))=(0,0,0)$.

Si aucun des points de C n'est singulier, on dit que C est une courbe non singulière.

Autrement elle est singulière.

Définition 1.5 : (courbes elliptiques).

Soit K un corps. On appelle courbe elliptique sur K, notée E(K),une courbe cubique, non singulière dans le plan projectif $P^2(K)$ représentée

par une équation spéciale dite de Weirstrass :

$$Y^2Z+a_1XYZ+a_3YZ^2 = X^3+a_2X^2Z+a_4XZ^2+a_6Z^3. \qquad (1)$$

et munie du point à l'infini noté θ, où les coefficients a_i sont dans le corps K.

Proposition1.6 :

Soit E une courbe elliptique. Si la caractéristique de K est différente de 2et 3, alors on peut se ramener à une équation de E, dite forme courte de Weirstrass :

$$E : Y^2Z = X^3+aXZ^2+bZ^3 . \qquad (2)$$

On peut écrire cette équation en coordonnées non homogènes(x=X/Z et y=Y/Z) :

$$E : y^2 = x^3+ax+b. \qquad (3)$$

Plus le point θ=(0,1,0) qui est le seul point à l'infini.

Preuve :

il suffit de considérer les changements de variables :

$Y=(Y-\frac{1}{2}(a_1X+a_3))$ et $X=(X-(a_1^2+4a_2)Z/12)$ pour obtenir le résultat de la proposition 1.6. □

Remarque :

Le point à l'infini $\theta=(0,1,0)$ n'est pas singulier ; en effet :

Regardons E comme une courbe de $P^2(K)$, avec K un corps de

caractéristique$\neq 2,3$,donnée par son équation :$F(X, Y, Z)=Y^2Z-X^3-aXZ^2-bZ^3=0$.

On a :

$(\partial F/\partial X(\theta),\partial F/\partial Y(\theta),\partial F/\partial Z(\theta)) = (0,0,1)\neq(0,0,0)$.

Théorème 1.7 :

La courbe elliptique E est non singulière si et seulement si l'expression

$4a^3+27b^2$ est non nulle.

Preuve :

Considérons la définition de la courbe E par son équation de Weirstrass réduite :

E : $f(x, y, z) =y^2-x^3-ax-b=0$.

E est singulière en un point $(\alpha,\beta)\in E$ si et seulement si : $\begin{cases} 3\alpha^2+a = 0 & \text{(i)} \\ 2\beta = 0 & \text{(ii)} \end{cases}$

Le point (α, β) est aussi un point de la courbe E, donc : $\beta^2 =0= \alpha^3+a\alpha+b$ (iii)

On remplace les valeurs : $\alpha=\pm\sqrt{-\frac{a}{3}}$ dans (iii).

Il s'ensuit que : $4a^3+27b^2 =0$.

Finalement : E est non singulière si et seulement si $4a^3+27b^2 \neq 0$. □

2. Structure de groupe abélien :

Dans cette partie nous allons montrer qu'on peut munir une courbe elliptique d'une opération de groupe commutatif.

Proposition 1.8 :

Soit E une courbe elliptique et D une droite définie sur un corps K. si E a au moins deux points d'intersection(comptés avec leurs multiplicités) avec la droite D alors E a exactement trois points d'intersections(comptés avec leurs multiplicités) avec la droite D.

Preuve :

Soit la courbe elliptique E définie par l'équation $f(x, y)=y^2-x^3-ax-b=0$ et munie du point à l'infini $\theta=(0, 1, 0)$.

Soit D une droite d'équation $y =\alpha x+\beta$.

Les points d'intersection de E et D sont tels que : $f(x, \alpha x+\beta)=0$.

Ceci implique : $x^3-\alpha^2x^2+(a-2\alpha\beta)x+(b-\beta^2) = 0$.

Notons par P(x) le polynôme $x^3-\alpha^2x^2+(a-2\alpha\beta)x+(b-\beta^2)$.

Considérons deux points (x_1,y_1)et (x_2,y_2) d'intersection de E et D différents de θ.

$x_1$et x_2 sont deux racines réelles du polynôme P(x) qui est de degré 3, il existe donc une troisième racine x_3 (x_3 n'est pas forcement différente de x_1 et x_2).

Il en résulte le troisième point d'intersection de E et D : $(x_3, \alpha x_3+\beta)$.

Dans le cas où x est constante la droite D est parallèle à l'axe (oy) et le troisième point d'intersection de E et D est le point à l'infini θ. □

2.1Loi de la sécante tangente :

Soit E une courbe elliptique définie sur $P^2(K)$ par :

E : $f(x, y)=y^2-x^3-ax-b=0$ munie du point θ.

On définit sur E une loi de composition * dite loi de composition de la sécante tangente comme suit :

Si P et Q sont deux points distincts de la courbe elliptique E alors P*Q est le troisième point d'intersection de la droite D passant par P et Q avec E .

Proposition 1.9 :

Soit $P_1=(x_1, y_1)$, $P_2=(x_2, y_2)$ et $P_3=(x_3, y_3)$ trois points de $E-\{\theta\}$.

1-Si $x_1 \neq x_2$ et si $P_3=P_1 * P_2$ alors :

$$\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=\lambda(x_3-x_1)+y_1 \end{cases} \quad \text{avec } \lambda=(y_2-y_1) / (x_2-x_1)$$

2-Si $P_3=P_1*P_1$ alors : $x_3=$

$$\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=\lambda(x_3-x_1)+y_1 \end{cases} \quad \text{avec } \lambda=(3x_1^2+a) / 2y_1 \ .$$

Preuve :

Si $x_1 \neq x_2$, la droite passant par P_1et P_2 a pour équation : $y=\alpha x+\gamma$

Avec $\alpha=(y_2-y_1) / (x_2-x_1)$.

On a $f(x, \alpha x+\gamma) =(\alpha x+\gamma)^2-x^3-ax-b$

$$=-x^3+\alpha^2x^2+(2\alpha\gamma-a)x+(\gamma^2-b).$$

Les abscisses des points P_1 , P_2 et P_3 sont les zéros de l'équation $f(x,\alpha x+\gamma)=0$.

D'où : $f(x, \alpha x+\gamma) = -(x-x_1)(x-x_2)(x-x_3)$.

Par identification on obtient : $x_3=\alpha^2-x_1-x_2$ et $y_3=\alpha(x_3-x_1)+y_1$.

Si $x_1=x_2$ et $y_1=y_2$ la tangente à la courbe E en P_1 a pour équation : $y=\alpha x+\gamma$

Avec $\alpha=(3x_1^2+a) / 2y_1$. □

Remarque :

Dans le cas où $x_1=x_2$ et $y_1=-y_2$ on pose $P_1 * P_2=\theta$.

Corollaire 1.10:

Soit E une courbe elliptique définie sur un corps K.

On munit E d'une opération de groupe commutatif + définie par :

$\forall P_1 ,P_2 \in E$; $P_1+P_2=\theta*(P_1*P_2)$.

Le point à l'infini θ est l'élément neutre de la loi du groupe, et on a :

1-Si $x_1=x_2$ et $y_1=-y_2$ alors $P_1+P_2=\theta$.

2- $P_1+P_2=P_3=(x_3,y_3)$ avec : $\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=-y_1+\lambda(x_1-x_3) \end{cases}$

Et

$\lambda = 3x_1^2+a / 2y_1$ si $P_1 = P_2$

ou

$\lambda = (y_2-y_1/x_2-x_1)$ si $P_1 \neq P_2$

Preuve :

La commutativité de la loi + est vérifiée par la symétrie de la définition(coïncidence des droites P_1P_2 et P_2P_1).

Le symétrique –P est le troisième point d'intersection de la courbe E avec la droite passant par P et θ.

Soit le point $P=(x_p, y_p)$ et $-P=(x_{-p}, y_{-p})$.

Les coordonnées du point –P inverse du point P sont obtenues par intersection de la courbe E et la droite $x=x_p$.

Le point –P a donc pour abscisse x_p et ordonnée solution de l'équation :

$y^2-x_p^3-ax_p-b=0$.

on a $f(x_p, y) = y^2-x_p^3-ax_p-b$

$=(y-y_p)(y-y_{-p})$.

par identification on obtient : $y_p+y_{-p}=0$ et donc $y_p=y_{-p}$.

Nous avons ainsi : $-p=(x_p, -y_p)$.

L'associativité de l'addition + est vérifiée par le calcul des points $(P_1+P_2)+P_3$ et $P_1+(P_2+P_3)$. □

3.Courbe elliptique sur le corps F_P :

Soit p un nombre premier, F_p le corps fini de cardinal p.

Théorème 1.11 (Hasse-1934) :

Soit la courbe elliptique $E(F_p)$donnée par l'équation : $y^2=x^3+ax+b$ sur le corps F_p alors $\#E(F_p)=p+1-t$ avec $t\in Z$ et $|t| \leq 2\sqrt{p}$.

Ce qui veut dire que le cardinal de $E(F_p)$ vérifie l'inégalité :

$$p-2\sqrt{p}+1 \leq \#E(F_p) \leq p+2\sqrt{p}+1.$$

Définition 1.12 :(Symbole de Legendre).

Soit p un nombre premier impair, le symbole de Legendre $(\frac{a}{p})$ est définie par :

$$(\frac{a}{p})= \begin{cases} 1 \text{ si a est un résidu quadratique modulo p ie } \exists x\in F_p \text{ tel que } x^2\equiv a \bmod p. \\ -1 \text{ si a est un non résidu quadratique modulo p ie } \forall x\in F_p ; x^2\equiv a \bmod p. \\ 0 \text{ si p divise a ie } a\equiv 0 \bmod p. \end{cases}$$

Définition 1.13 :

Soit E une courbe elliptique définie sur un corps fini F_q de caractéristique p.

L'automorphisme de Frobenius est défini par :

$$\phi_E : E(F_q^{alg}) \rightarrow E(F_q^{alg}) \text{ tel que } \phi_E(P) = \begin{cases} \theta & \text{si } P = \theta. \\ (x^q, y^q) & \text{si } P = (x, y) \end{cases}$$

Théorème 1.14 :

Le polynôme caractéristique de ϕ_E est $\phi_E{}^2 - t\,\phi_E + q = 0$ ($t \in Z$).

Où t est défini par : $\# E(F_q) = q+1-t$.

t est appelé la trace du Frobenius ϕ_E.

Théorème 1.15 : (Algorithme de Schoof).

Le nombre de points de E dans F_p est : $1+p+\sum_x \left[\frac{x3+ax+b}{p} \right]$.

Preuve :

Considérons la courbe elliptique E (F_P) d'équation : $y^2=x^3+ax+b$.

Pour x fixé dans F_p, si x^3+ax+b est un résidu quadratique modulo p ; x^3+ax+b possède deux racines carrés modulo p.

On obtient ainsi tout les points fini de E :

$$1+\left[\frac{x3+ax+b}{p}\right]=\begin{cases}2 \text{ si } \forall x\ ;\ x^3+ax+b \text{ est un résidu quadratique mod p}\\ 0 \text{ si } \forall x\ ;\ x^3+ax+b \text{ est un non résidu quadratique mod p.}\\ 1 \text{ si } \forall x\ ;\ x^3+ax+b\equiv 0 \text{ mod p.}\end{cases}$$

Il en résulte le nombre de points finis de E est :

$$\sum_x 1+\left[\frac{x3+ax+b}{p}\right] = p+\sum_x\left[\frac{x3+ax+b}{p}\right].$$

En ajoutant le point à l'infini on obtient le résultat du théorème 2. □

Lemme 1.16:

Le nombre des courbes elliptiques de la forme $y^2 = x^2 + ax + b \bmod p$;

avec p premier différent de 2 et 3 est $p^2 - p$.

preuve :

Le nombre des courbes elliptiques sur le corps F_P est égale aux nombres de

Paires (a , b) vérifiant $4a^3 + 27b^2 \neq 0$.

Le nombre de paire (a , b) est p^2 , reste à éliminé les paires (a ,b) avec

$4a^3 + 27b^2 = 0$

Notons que $4a^3 + 27b^2 = 0$ si et seulement si $a=-3\,c^2$ et $b=2c^2$ pour un c dans F_p .

L'élément c est déterminé uniquement par $c=-(3/2)\,(\,b/a\,)$ (si $a\neq 0$).

Ainsi : $4a^3 + 27b^2 = 0$ pour exactement p paires (a , b) il en résulte :

$$\#\{ E ; E \text{ courbe elliptique sur } F_p \} = p^2 - p . \quad \square$$

Remarque :

Le logiciel MAGMA contient une implémentation de l'algorithme de Schoof.

La complexité totale de l'algorithme est $O(\ln^8 p)$.

Exemple 1:

```
Magma V2.10-17    Mon Jun 07 2004 11:37:40    [Seed = 2130315608]
Type ? for help.  Type <Ctrl>-D to quit.

> k:=GF(NextPrime(123698547));

> k;
Finite field of size 123698551

> E:=EllipticCurve([k|38523647,76902501]);

> E;
Elliptic Curve defined by y^2 = x^3 + 38523647*x + 76902501 over GF(123698551)

> time Order(E);
123695154
Time: 0.050

> FactoredOrder(E);
[ <2, 1>, <3, 3>, <11, 3>, <1721, 1> ]

> P:=Random(E);
> P;
(87037600 : 52119703 : 1)

> IsOrder(P,Order(E));
false

> FactoredOrder(P);
[ <3, 3>, <11, 3>, <1721, 1> ]

> TraceOfFrobenius(E);
3398

> time TraceOfFrobenius(E,25);
1822584300956518364913165890263386253683072591863227510096530745125168804041 24\
```

122851995540686484495002
Time: 0.000

Exemple 2 :

Magma V2.10-17 Tue Jun 08 2004 10:14:58 [Seed = 716120042]
Type ? for help. Type <Ctrl>-D to quit.

> p:=NextPrime(5^246);

> k:=GF(p);

> E:=EllipticCurve([Random(k),Random(k)]);

> time E;
Elliptic Curve defined by y^2 = x^3 + 737981648876303647187534893654382329895413302825659443448945320389\

188287896690870312715108565908942721627599829364963769\

402078344785095086687782823110905588886573271765461*x +
280232892359549420140988734721408263018221418056410064830876021437298388758902\

940257790061797865999600833383037104348532373208120011384260698831084458996735\

84756738800032770 over GF(884343660041671129639562430750760916082278437318275213\

826427235345263800643832530736282368355562314399800042900085725552332298132155\

91341430067586770746856927871704101592 3)
Time: 0.000

> time Order(E);
884343660041671129639562430750760916082278437318275213826427235345263800643832\

530736282368355562314399800042900085725552332298132155591341430067586770746856\

92787170410159233073628294855362025272181406870893955553872243778915247394353\

16998807794783903381515258956221520
Time: 2554.273

> aInvariants(E);
[0, 0, 0, 737981648876303647187534893654382329895413302825659443448945320389\

188287896690870312715108565908942721627599829364963769402078344785095086687782\

823110905588886573271765461, 280232892359549420140988734721408263018221418056\

410064830876021437298388758902940257790061797865999600833383037104348532373208\

120011384260698831084458996735847567388000327703]

> bInvariants(E);
[0, 147596329775260729437506978730876465979082660565131888689789064077837657\

57933817406254302171318178854432551996587299275388041566895701901733755656462\

2181

1177773146543530922, 2365879093965265509243925081348721359906072349073650454970\

76850403929754391779230294877878835901684003533489248331668577160534347889945 70\

13652567510652400864623978478997185, 2258872504196143643453335317967472175982 40\

33057787545723380753364300823239642360859888366062472227833587751910165887980 60\

15330513214930592237629283189149364927260531515797 0]

> cInvariants(E);
[879406385602098141697644664212769396913408333028210740577198638858215221315 04\

64761803793206614486469351877353194768016685315146058095002910489803829213874 57\

9838068160337487, 188943852767190519425839226411751758800986624469112574564179 9\

62776473488717973039010755536067850490425168810565331669368597872520796026308 04\

84618026114590259386236915531574]

> jInvariant(E);
381041333336109995603053695353869049755955956701083110248082210198761134260481 4\

84926897645272148784550222871601257892682606290170113984423154971784098937006 22\

04439678783987

> Discriminant(E);
704378765539010204467762484249143344488287004949650790459440612775153455019366 2\

34123390418776856231470380379369358178048965887810529367123289716182068957498 98\

81131494156770

> TraceOfFrobenius(E);
124870064629577816068279938698283199114807830918030117399204911770921768872808\

854606228

> TraceOfFrobenius(E,3);
136580089100690825268793089866456103379952641103567306939976998629674364686849 3\

17597015383152366307567122002166208708056614307818964636247113083076567664007 12\

99520816978753249795284287139648373116659636913059354960293618376670449042337 86\

6222786175638778140980

> P:=Random(E);

> P;
(539179590517232932248818565502677455146980245932466297969131741726496373041563\

78466251581062844097245175630114895151910920836646387049782260395512197663490 19\

304216252773314 : 64655810421345010054403990635788744501702190922030047112687 50\

4803986214216331720329774465121357362880819418226672659985765913206816495278351\

019196487135924755284897423741 48 : 1)

> 100*P;
(3878087853732290000448120440215589846306734964258726440417155017481830078567 11\

342435779525659701776202210981302558722920910795837673566656959021750023259722 2\

907307923934982 : 7237902811175147529124254836104063815608469715835236844924942\

630390970125797067399810421388183505846569346713997610587333225336872294082839 0\

096477693175650874893573720394 61 : 1)

> Q:=Random(E);

> Q;
(5584184362325009109247534195871402277966786139213686525432454543761379545316 69\

083566461488653983266582157430529664237406115021589784481867164390878251263336 8\

793156970366998 : 1985047393351233965938521132609775689505365501403321637386769\

681442112617105814077998424645203511848150944168315278731618821120932294427375 9\

894060540109771036530871294781 08 : 1)

> P+Q;
(4221086672804255883062839670906920581268415847003013421493036734206048682653 91\

001825685642791684909104082115731335470623567156498539138221338930304720091732 5\

45344712378385 : 536890278829889373287529091865683322225848955646366042087504 03\

854492274262103020259132480981025544378960680089541125501863494695024932611027 1\

720563493439878458546424984358 4 : 1)

> -P;
(5391795905172329322488185655026774551469802459324662979691317417264964737 04156 3\

784662515810628440972451756301148951519109208366463870497822603955121976634901 9\

304216252773314 : 2377855558282210290955225243928734710652565280979747426995521\

873054016584805153274385377171419886855916058606333591256946731660639906385579 1\

987562189938760937502272986417 75 : 1)

> Q-P;
(2779561250452653690386683206236649147357935564886900416612497416624229662311 84\

850852737659748600489475797295381188234019080585447955602402820514190054858355 6\

667229878941267 : 826207776275131970609186347837045127009100266314051837085757\5

67984266541201816083177892972010711877403818616269031015265891010901493512558055\

113428210274301089479481801364921 : 1)

Chapitre 2

Factorisation sur une courbe elliptiques

Ce chapitre est consacré à la description de deux applications liées aux courbes elliptiques.

Nous allons voir comment il est possible de tester la primalité d'un nombre et s'il n'est pas premier comment le factoriser en utilisant les courbes elliptiques.

Nous utilisons une courbe elliptique E de la forme :

$y^2 = x^3+ax+b$; $4a^3+27b^2 \neq 0$.

Le point θ est le point à l'infini sur la courbe E.

1.Test de primalité :

1.1Le critère de Pocklington :

Théorème 2.1 :

Soit n un entier positif. Notons $n-1 = \prod_{j=1}^{n} p_j^{ej}$ la décomposition de n-1 en facteurs premiers.

Supposons qu'il existe des entiers a et i dans l'ensemble {1, …, n} tels que :

$a^{n-1} \equiv 1 \bmod n$ et $\text{pgcd}(a^{n-1/pi} - 1, n) = 1$.

Si d est un facteur de n , alors $d \equiv 1 \bmod p_i^{ei}$.

Corollaire 2.2 : (Critère de Pocklington)

Si q est un facteur premier de n-1 strictement supérieur à $\sqrt{n}$ -1 et s'il existe un entier a vérifiant $a^{n-1} \equiv 1 \bmod n$ et pgcd $(a^{n-1/q}-1, n) = 1$, alors n est premier.

Le critère de Pocklington a été affiné par Lehmer de la manière suivante :

Corollaire 2.3 : (Critère de Pocklington – Lehmer)

S'il existe des entiers F,U tels que :

1. n-1 = FU.
2. pgcd (F, U) = 1.
3. $F > \sqrt{n} - 1$.

Et si pour tout facteur premier p_i de F il existe a_{pi} qui vérifie :

$a_{pi}^{n-1} \equiv 1 \bmod n$ et pgcd $(a_{pi}^{n-1/pi} - 1, n) = 1$, alors n est premier.

Remarque :

Le critère de Pocklington est le critère de Pocklington-Lehmer dans le cas particulier où F est un facteur premier de n-1.

Le test de primalité utilisant les courbes elliptiques est une variante du test de

Pocklington dans $(Z / nZ)^*$.

1.2Le critère de Goldwasser-Kilian :

Soit n un entier dont on veut tester la primalité.

Dans cette partie nous allons établir l'analogue du critère de Pocklington dans le groupe (E, +), où E est définie sur le corps (Z / nZ, +, ×).

Comparons les groupes $(Z / nZ)^*$ et (E, θ) :

	$(Z/nZ-\{0\}, \times)$	(E, +)
Eléments	{1, …, n-1}	$\{(x, y)\in(Z/nZ) \mid y^2 = x^3+ax+b\}\cup \theta$
Loi de composition	×	+
Elément neutre	1	θ
Cardinalité	n-1	#E avec $\mid \#E - (n-1)\mid \leq 2\sqrt{n}$

1)La condition $a^{n-1} \equiv 1 \bmod n$ devient # E . P = θ.

2)$pgcd(a^{n-1/q} - 1, n) = 1$ implique $a^{n-1/q} \equiv 1 \bmod n$.

Donc la condition $pgcd (a^{n-1/q} - 1, n) = 1$ devient $\frac{\#E}{q}$. P ≠ θ.

3)$q > \sqrt{n} - 1$ implique $\forall p < \sqrt{n}$, q > p.

dans (E, +) on écrit :

$\forall p < \sqrt{n}$, q > #E_p avec E_p est la courbe elliptique E modulo p.

Le théorème de Hasse implique #E $\leq (\sqrt{p}+1)^2 \leq (n^{1/4}+1)^2$.

Alors la condition q > $\sqrt{n}$ -1 devient q > $(n^{1/4}+1)^2$.

Proposition 2.4 : (Critère de Goldwasser-Kilian)

Soit n$\in$N un entier relativement premier à 6.

S'il existe un entier m et un point P de la courbe elliptique E : $y^2 = x^3+ax+b \cup \theta$

tels que :

1.Il existe un facteur q de m strictement supérieur à $(n^{1/4} + 1)^2$.

2.mP = θ.

3.$\frac{m}{q}$ P = (x, y, z) avec z $\in (Z/nZ)^*$.

Alors n est premier.

Preuve :

On fait un raisonnement par l'absurde.

Supposons que n a un facteur premier p.

On note E' la courbe elliptique E modulo p, m' le cardinal de E' et P' le point de E' correspondant au point P sur E.

Par hypothèse, mP = θ et $\frac{m}{q}$ P $\neq \theta$ sur E. Donc mP' = θ et $\frac{m}{q}$ P' $\neq \theta$ sur E'.

Ceci implique que q divise l'ordre de P' comme point de E' et par suite q divise m' (théorème de Lagrange) .

D'après le théorème de Hasse il en résulte que :

$q \leq m' \leq (\sqrt{p}+1)^2 \leq (n^{1/4}+1)^2$.

Contradiction avec l'hypothèse. □

Remarque :

On fait les calculs en suivant les étapes de la proposition 2.4 comme si n était premier.

Lors du calcul des coordonnées des points mP et $\frac{m}{q}$P, si le dénominateur intervenant dans le calcul de λ (corollaire 1.10) ne soit pas inversible, cela signifie que Z/nZ n'est pas un corps et donc que n est composé.

Exemple 3 : Test de primalité avec les courbes elliptiques (ECPP method).

ECPP = Elliptic Curve Primality Proving

On va utiliser la méthode ECPP implémentée par François Morain.

```
>p:=NextPrime(954685120321145782520102132125454879021358478521057896208752012365847801236980240123587978410236587403214589021546548789210215487963210584780123588741023698547503214897410326023569784520123600019874520125487951021548785201269874520321459872032159874520123658749851203697858785489521305879509785236 57);
Time: 350.391

> p;
954685120321145782520102132125454879021358478521057896208752012365847801236980240123587978410236587403214589021546548789210215487963210584780123588741023698547503214897410326023569784520123600019874520125487951021548785201269874520321459872032159 87\
```

4520123658749851203697858785489521305879509785246 93

> time IsPrime(p);
true
Time: 342.625

2.Factorisation :

2.1Calcul de kP :

Soit K un entier naturel.

Le calcul du point kP se fait par la méthode du doublement de l'addition :

Si $k=k_0+k_1 2^1+k_2 2^2+\ldots+k_n 2^n$ est l'écriture de k en base deux alors :

$kP=\sum_{i=0}^{n} k_i(2^iP)$.

Cette méthode permet de calculer kP avec O(lnk) opérations.

Si $k=p_1^{t_1}.p_2^{t_2}\ldots p_s^{t_s}$ avec $p_1< p_2 <\ldots< p_s$.

Le calcul de kP se fait de la manière suivante :

On calcul p_1P, $p_1(p_1P)$, …, $p_1^{t_1}P$ par la méthode du doublement de

l'addition ;ensuite

$P_2(p_1^{t_1}P)$, $p_2(p_2p_1^{t_1})$, …, $p_2^{t_2}(p_1^{t_1}P)$. et par suite : $p_s^{t_s}(\ldots(p_1^{t_1}P)\ldots)$.

Finalement : $kP=\prod_{i=1}^{s} p_i^{t_i}P$.

Et dans ce cas kP est calculé avec $O(t_1 lnp_1+\ldots+t_s lnp_s)$ opérations.

Un entier d est appelé diviseur non trivial d'un entier n si d est un diviseur de n

et si :$1< d < n$.

2.2Méthode (p-1) de pollard :

Cette méthode est une tentative pour trouver un diviseur non trivial d'un entier n, n> 1 de la manière suivante :

1-On choisit un entier B $\in$ N et un autre entier k $\in$ N tels que :

k= ppcm (1,…,B).

2-On choisit un nombre a quelconque de l'ensemble {2,..,n-2} puis on calcul

a^k mod n et d=pgcd(a^k-1, n) avec l'algorithme Euclidien.
Si d est différent de 1 et de n ,on a trouvé un facteur non trivial de n et l'algorithme est terminé.
Si d est égal à 1ou n on fait un autre choix de a dans {2,…,n-2} et / ou un autre choix de k.(si d=1, nous augmentons la valeur de B).

Discussion :

1-On suppose que n admet un facteur premier p tel que p-1 est une puissance de petit nombre premier (inférieurs à la borne B).
alors on a : p-1 divise k , d'après le petit théorème de FERMAT
$a^k \equiv 1$ mod p. donc p divise $a^k - 1$, ainsi p divise pgcd (a^k-1, n).

2- L'algorithme de Pollard a peu de chance de fonctionner , car il n'est pas fréquent que p-1 soit puissance de petit nombres premiers . d'autant que , dans les cryptosystèmes comme le RSA on fabrique l'entier n et on peut s'arranger pour que n ne vérifie pas cette condition .

3-Dans le groupe multiplicatif (Z /pZ)*, si l'ordre de ces groupes pour p|n est divisible seulement par des nombres premiers grands ; cet algorithme ne fonctionne pas .

Exemple 4:

Soit l'entier n=2546907(n est supposé non premier).

On prend B=16 et a=5.

k=ppcm(1,2,…,16)=$2^4\times3^2\times5\times7\times11\times13$=720720.

5^{720720}mod 2546907=1678051.

On calcul la pgcd(1678050,2546907) par l'algorithme Euclidien .

Alors pgcd(1678050,2546907) =3729.

Ainsi nous avons : n=3729×683.

683 étant un nombre premier, on fait la factorisation de 3729.

En prenant B=7 et a=5 on obtient par la même méthode 3729 = 33×113

D'où la factorisation de n : n = 3×11×113×683.

2.3Courbes elliptiques modulo n :

Soit n un entier positif , considérons la courbe cubique E défini sur Z /n Z par l'équation : $y^2 = x^3+ ax +b$.

l'ensemble E(Z /n Z) des points de E sur Z / n Z est défini par :

E(Z /n Z)= {[x,y,z] $\in$ P^2 (Z /n Z) : $y^2 z= x^3 +ax z^2+bz^3$.}

Si $6(4a^3+27b^2) \in (Z/nZ)^*$, E est appelée courbe elliptique sur Z/nZ.

On note le point [0,1,0] de $P^2(Z/nZ)$ par θ.

Théorème 2.5 :

Soient deux points : $P=[x_1, y_1, 1]$ et $Q=[x_2, y_2,1]$ sur la courbe E.

On définit la somme P+Q par :

$$P+Q = \begin{cases} \theta & \text{si } x_1=x_2 \text{ et } y_1=-y_2 \\ [x_3, y_3, 1] & \text{sinon.} \end{cases}$$

Avec : $x_3= (3x_1^2+a / 2y_1)^2-2x_1$; $y_3= -y_1+(3x_1^2+a / 2y_1)(x_1-x_3)$

Si $[x_1, y_1, 1] =[x_2, y_2, 1]$

Et $x_3=(y_2-y_1/x_2-x_1)^2-x_1-x_2$; $y_3=-y_1+(y_2-y_1/x_2-x_1)(x_1-x_3)$

Si $[x_1, y_1, 1] \neq [x_2, y_2, 1]$.

2.4Factorisation par courbes elliptiques (algorithme de Lenstra) :

L'idée de cet algorithme est due à Lenstra.

On suppose que l'on doit factoriser un entier n donné.

On note p un diviseur de n (que l'on veut trouver).

Définition 2.6:

Soit $n \in N$ et soit $n=\prod_{i=1}^{m} p_i^{\alpha i}$ la décomposition de n en facteurs premiers.

n est B-lisse si et seulement si : $p_i \leq B$, $\forall i \in \{1,\ldots,m\}$.

n est B-superlisse si et seulement si : $p_i^{\alpha i} \leq B$, $\forall i \in \{1,\ldots,m\}$.

2.4.1Algorithme de Lenstra :

Considérons un entier n, impair, composé.

Nous voulons trouver un diviseur non trivial d de n.

1-On choisit une courbe elliptique E : $y^2=x^3+ax+b\cup\{\theta\}$ avec a et b dans Z, et un point P sur E.

2-On vérifie que pgcd($4a^3+27b^2$, n)=1(pour que E(Z/nZ) soit non singulière).

3-On choisit un entier B dans N et un autre entier $k\in N$ tels que k=ppcm(1,…,B)

5-On calcul kP.

Discussion :

1-Dans le choix de la courbe elliptique, on choisit a et b tels que pgcd($4a^3+27b^2$,n)=1.

Si nous les prenons aléatoirement et que ce n'est pas le cas, ce pgcd est un diviseur de n.

2-Le calcul des coordonnées du point kP s'effectue modulo n : comme le facteur p est inconnu, le calcul de kP se fait dans E(Z/nZ) et non dans E(Z/pZ). Ces calculs font intervenir des divisions et ceci n'est pas toujours possible modulo n.

En effet: si pgcd(x_2-x_1, n)≠1 et pgcd($2y_1$,n)≠1on ne peut pas calculer les inverses de x_2-x_1 et $y_1$dans la formule d'addition donnée dans le théorème 2.5.

Dans ce cas un diviseur de n est d = pgcd(x_2-x_1, n) ou d = pgcd($2y_1$, n).

Si on a pu mener les calculs jusqu'au bout, on recommence à l'étape 1 en changeant de courbe elliptique.

3-La méthode des courbes elliptiques est obtenue à partir de la méthode (p-1) de Pollard, en remplaçant le groupe $(Z/pZ)^*$ par le groupe E(Z/pZ).

Ainsi pour que l'algorithme de Lenstra fonctionne il suffit que l'ordre du groupe E(Z/pZ) soit B-superlisse.

Remarques :

1-On prend l'entier n tel qu'il ne soit pas divisible par 2 ou 3, pour nous assurer que Z/pZ sera de caractéristique $\neq 2,3$.

2-Pour le choix de P sur la courbe elliptique E, on peut imposer que b=-a et on considère les courbes paramétrées par $a \in Z$:

E_a : $y^2=x^3+ax-a$.Ces courbes contiennent le point (1, 1).

2.4.2Choix de la borne de lissité :

On peut suivre le raisonnement suivant pour choisir la borne B :

Tout nombre composé a un diviseur premier inférieur à $\sqrt{n}$,si p est un facteur premier de n on a :p $< \sqrt{n}$.

De plus, d'après le théorème de Hasse, la cardinalité de $E(F_p)$est inférieure à $(\sqrt{p}+1)^2$.

Il nous suffit donc de prendre $B \geq [(n^{1/4}+1)^2]$.

Remarques :

*L'algorithme peut être répété jusqu'à ce que la factorisation complète de n soit obtenue, nous choisissons à chaque fois une nouvelle courbe et un nouveau point donnés par le triplet (a, x, y)$\in Z_n^3$.

*Les courbes elliptiques permettent de factoriser en un temps polynomial.

Cependant, elles permettent seulement de trouver des facteurs premier ayant

moins de 70 chiffres décimaux (la méthode de factorisation par crible quadratique est la meilleure méthode disponible pour factoriser des nombres entiers de plus qu'environ 40 chiffres décimaux)

*Il a été montré que le temps moyen d'aboutissement de l'algorithme est $O(e^{(1+\varepsilon)\ln n \ln\ln n})$.

Exemple 5 :

```
> n:=27830575208037651856006080702l;
> SetVerbose("Factorization",1);
> time Factorisation(n);
Integer main factorization (primality of factors will be proved)
Seed: 1
   Number: 27830575208037651856006080702l

Trial Division
   Number: 27830575208037651856006080702l
   Minimum: 2
   Maximum: 10000
   No factors found
   Time: 0.000

Pollard Rho
   Trials: 8191
   Number: 27830575208037651856006080702l
   (30 digits)
   No factor found
Time: 0.015
```

```
1 composite number remaining

ECM
x: 27830575208037651856006080702 1
   (30 digits)
   Initial smoothness: 500, steps: 2, step size: 100
   Step 1/2; smoothness: 500, digits: 30, elapsed time: 0.000
   Step 2/2; smoothness: 600, digits: 30, elapsed time: 0.078
   No factor found
   Time: 0.171

MPQS
   x: 27830575208037651856006080702 1



(30 digits)
   Factor 1: 302159874512047 (15 digits)
   Factor 2: 921054632187043 (15 digits)
   Total MPQS Time: 0.078

Total time: 0.281

[ <302159874512047, 1>, <921054632187043, 1> ]
Time: 0.281

> m:=304533566097581210092619532619966326338351568200064358301522 9;
> time Factorisation(m);
Integer main factorization (primality of factors will be proved)
Seed: 1
   Number: 304533566097581210092619532619966326338351568200064358301522 9

Trial Division
   Number: 304533566097581210092619532619966326338351568200064358301522 9
   Minimum: 2
   Maximum: 10000
   No factors found
   Time: 0.000

Pollard Rho
   Trials: 8191
   Number: 304533566097581210092619532619966326338351568200064358301522 9
   (61 digits)
   No factor found
   Time: 0.031

1 composite number remaining
ECM
   x: 304533566097581210092619532619966326338351568200064358301522 9
   (61 digits)
   Initial smoothness: 500, steps: 24, step size: 200
   Step 1/24; smoothness: 500, digits: 61, elapsed time: 0.000
   Step 2/24; smoothness: 700, digits: 61, elapsed time: 0.171
   Step 3/24; smoothness: 900, digits: 61, elapsed time: 0.390
   Step 4/24; smoothness: 1100, digits: 61, elapsed time: 0.625
```

```
   Step 5/24; smoothness: 1300, digits: 61, elapsed time: 0.906
   Step 6/24; smoothness: 1500, digits: 61, elapsed time: 1.218
   Step 7/24; smoothness: 1700, digits: 61, elapsed time: 1.593
   Step 8/24; smoothness: 1900, digits: 61, elapsed time: 1.984
   Step 9/24; smoothness: 2100, digits: 61, elapsed time: 2.406
   Step 10/24; smoothness: 2300, digits: 61, elapsed time: 2.859
   Step 11/24; smoothness: 2500, digits: 61, elapsed time: 3.375
   Step 12/24; smoothness: 2700, digits: 61, elapsed time: 3.906
   Step 13/24; smoothness: 2900, digits: 61, elapsed time: 4.484
   Step 14/24; smoothness: 3100, digits: 61, elapsed time: 5.109
   Step 15/24; smoothness: 3300, digits: 61, elapsed time: 5.781
   Step 16/24; smoothness: 3500, digits: 61, elapsed time: 6.468
   Step 17/24; smoothness: 3700, digits: 61, elapsed time: 7.187
   Step 18/24; smoothness: 3900, digits: 61, elapsed time: 7.984
   Step 19/24; smoothness: 4100, digits: 61, elapsed time: 8.812
   Step 20/24; smoothness: 4300, digits: 61, elapsed time: 9.656
   Step 21/24; smoothness: 4500, digits: 61, elapsed time: 10.531
   Step 22/24; smoothness: 4700, digits: 61, elapsed time: 11.484
   Step 23/24; smoothness: 4900, digits: 61, elapsed time: 12.468
   Step 24/24; smoothness: 5100, digits: 61, elapsed time: 13.500
   No factor found
   Time: 14.703

MPQS
   x: 304533566097581210092619532619966326338351568200064358301522 9
   (61 digits)
   Factor 1: 95120223252555655555554784410319 (31 digits)
   Factor 2: 320156487952102354871023459891 (30 digits)
   Total MPQS Time: 72.765

Total time: 87.500

[ <320156487952102354871023459891, 1>, <95120223252555655555554784410319, 1> ]
Time: 87.500
```

Exemple 6 : Factorisation avec les courbes elliptiques

```
> t:=291843501046320161162477811153442159623 13;
> time ECM(t);
[]
[ 291843501046320161162477811153442159623 13 ]
10
Time: 1.891
> SetVerbose("Factorization",1);
> time ECM(t);

ECM
   x: 291843501046320161162477811153442159623 13
   (41 digits)
   Initial smoothness: 500, phase 2 multiplier: 10
   Steps: 10, step ratio: 1.200
```

```
   Step 1/10; smoothness: 500, digits: 41, elapsed time: 0.000
   Step 2/10; smoothness: 600, digits: 41, elapsed time: 0.109
   Step 3/10; smoothness: 720, digits: 41, elapsed time: 0.218
   Step 4/10; smoothness: 864, digits: 41, elapsed time: 0.359
   Step 5/10; smoothness: 1036, digits: 41, elapsed time: 0.515
   Step 6/10; smoothness: 1243, digits: 41, elapsed time: 0.687
   Step 7/10; smoothness: 1491, digits: 41, elapsed time: 0.875
   Step 8/10; smoothness: 1789, digits: 41, elapsed time: 1.093
   Step 9/10; smoothness: 2146, digits: 41, elapsed time: 1.359
   Step 10/10; smoothness: 2575, digits: 41, elapsed time: 1.640
   No factor found
   Time: 1.968

[]
[ 291843501046320161162477811153442159623 13 ]
10
Time: 1.969
> time ECM(t,5000,1.2,50);

ECM
   x: 291843501046320161162477811153442159623 13


  (41 digits)
   Initial smoothness: 5000, phase 2 multiplier: 10
   Steps: 50, step ratio: 1.200
   Step 1/50; smoothness: 5000, digits: 41, elapsed time: 0.000
   Step 2/50; smoothness: 6000, digits: 41, elapsed time: 0.640
   Step 3/50; smoothness: 7200, digits: 41, elapsed time: 1.359
   Step 4/50; smoothness: 8640, digits: 41, elapsed time: 2.234
   Step 5/50; smoothness: 10368, digits: 41, elapsed time: 3.234
   Step 6/50; smoothness: 12441, digits: 41, elapsed time: 4.421
   Step 7/50; smoothness: 14929, digits: 41, elapsed time: 6.015
   Step 8/50; smoothness: 17914, digits: 41, elapsed time: 7.796
   Step 9/50; smoothness: 21496, digits: 41, elapsed time: 9.859
   Step 10/50; smoothness: 25795, digits: 41, elapsed time: 12.250
   Step 11/50; smoothness: 30954, digits: 41, elapsed time: 15.046
   Step 12/50; smoothness: 37144, digits: 41, elapsed time: 18.343
   Step 13/50; smoothness: 44572, digits: 41, elapsed time: 22.218
   Step 14/50; smoothness: 53486, digits: 41, elapsed time: 26.812
   Step 15/50; smoothness: 64183, digits: 41, elapsed time: 33.093
   Step 16/50; smoothness: 77019, digits: 41, elapsed time: 40.390
   Step 17/50; smoothness: 92422, digits: 41, elapsed time: 48.890
   Step 18/50; smoothness: 110906, digits: 41, elapsed time: 58.734
   Step 19/50; smoothness: 133087, digits: 41, elapsed time: 70.359
   Step 20/50; smoothness: 159704, digits: 41, elapsed time: 83.953
   Step 21/50; smoothness: 191644, digits: 41, elapsed time: 100.015
   Factor: 965874501233587452137 (21 digits)
   Cofactor: 30215468021320154849 (20 digits)
   Time: 123.281

[ <30215468021320154849, 1>, <965874501233587452137, 1> ]
[]

21
Time: 123.297
```

Exemple 7 : Factorisation avec MPQS method (crible quadradique)

```
Magma V2.10-17    Wed Jun 09 2004 11:33:30    [Seed = 1709440078]
Type ? for help.  Type <Ctrl>-D to quit.
> p:=NextPrime(3210156489702875102256537);
> q:=NextPrime(9658741202125562320332252);
> n:=p*q;
> n;
3100607075236392318550202595059738671961358954853 7
> SetVerbose("Factorization",1);
> time MPQS(n);

MPQS
   x: 3100607075236392318550202595059738671961358954853 7
   (50 digits)
   Factor 1: 965874120212556232033228 3 (25 digits)
   Factor 2: 321015648970287510225653 9 (25 digits)
Total MPQS time: 5.984

[ <321015648970287510225653 9, 1>, <965874120212556232033228 3, 1> ]
[]
Time: 5.984



> m:=3056500446283354906835929754225442183285789743194516941325 69;
> time MPQS(m);

MPQS
   x: 3056500446283354906835929754225442183285789743194516941325 69
   (60 digits)
   Factor 1: 952134482256537210232121525563 (30 digits)
   Factor 2: 321015623658489702875102256763 (30 digits)
Total MPQS time: 35.859

[ <321015623658489702875102256763, 1>, <952134482256537210232121525563, 1> ]
[]
Time: 35.859
>
```

Chapitre 3

Diviseurs sur une courbe elliptique

Soit K un corps algébriquement clos, E une courbe elliptique définie sur K.

K(E) désigne l'ensemble des fonctions rationnelles sur E.

Définition 3.1 :

1-Un diviseur D sur E est la somme formelle : $D=\sum_{PdansE} n_p(P)$; où $n_p \in Z$ et les n_p sont presque tous nuls.

2-Le degré du diviseur D est défini par : $degD=\sum_{P} n_p$.

Remarques :

1-Le groupe des diviseurs sur E, noté Div(E), est le groupe abélien libre engendré

par les points de E.

$$Div(E)=\{ \sum_{P} n_p(P) \text{ ; } n_p \in Z \text{ et les } n_p \text{ presque tous nuls}\}.$$

2-Si D_1 , $D_2 \in Div(E)$ alors : $deg(D_1+D_2) = deg\ D_1+deg\ D_2$.

3-Si $\forall P \in E$; $n_p=0$ on écrit D=0 (D est un diviseur de degré 0).

Le groupe des diviseurs de degré 0 est un sous-groupe de Div(E), noté Div°(E).

Définition 3.2 :

Un diviseur $D = \sum_{P} n_p(P)$ sur E est positif si $n_p \geq 0$, $\forall P \in E$.

On écrit $D \geq 0$.

Si $D_1, D_2 \in Div(E)$ alors :

$D_1 \geq D_2$ si et seulement si $D_1 - D_2$ est un diviseur positif.

Définition 3.3 :

Un diviseur D sur E est principal s'il s'écrit $D = div(f)$ pour $f \in K(E)$.

Deux diviseurs D_1et D_2 sont linéairement équivalent si $D_1 - D_2$ est un diviseur principal. On écrit $D_1 \sim D_2$ ou $D_1 \equiv D_2$.

Remarques :

1-Si $f, g \in K(E)$ alors :

$div(fg) = div(f) + div(g)$ et $div(f / g) = div(f) - div(g)$.

2-Un diviseur principal est un diviseur de degré 0, il s'ensuit que :

Si D_1 et D_2 sont deux diviseurs sur E linéairement équivalent alors

$degD_1 = degD_2$.

L'espace L(D) :

Définition 3.4 :

Soit $D = \sum_P n_p(P)$ un diviseur sur E, on pose :

$L(D) = \{f \in K(E) : div(f) + D \geq 0\} \cup \{f \equiv 0\}$.

L'ensemble L(D) est un espace vectoriel sur K de dimension finie.

On note : $\dim_K L(D) = l(D)$.

Lemme 3.5 :

Soient D et D' deux diviseurs sur E.

Si $D \sim D'$ alors $l(D) = l(D')$.

Preuve :

$D \sim D'$ implique $D = D'+\mathrm{div}(g)$, $g \in K(E)$.

Considérons la fonction linéaire injective $\phi : L(D) \to L(D')$ définie par

$\phi(f) = f.g$.

Soit $f \in L(D)$ alors :

$\mathrm{Div}(fg)+D' = \mathrm{div}(f)+\mathrm{div}(g)+D' = \mathrm{div}(f)+D \geq 0$ donc $fg \in L(D')$.

Ce qui implique que ϕest bien une application de L(D) dans L(D').

Soit $h \in L(D')$ et $h' = h / g \in K(E)$ alors :

$\mathrm{div}(h')+D = \mathrm{div}(h / g)+D = \mathrm{div}(h)-\mathrm{div}(g)+D = \mathrm{div}(h)+D' \geq 0$ donc $h' \in L(D)$,

il en résulte que l'application ϕ est surjective.

Alors ϕ est un isomorphisme de L(D) dans L(D').

D'où le résultat : $l(D) = l(D')$. □

Proposition 3.6 :

Soit $D \in \mathrm{Div}(E)$: Si $D < 0$ alors $l(D) = 0$.

Preuve :

Soit $f \in K(E)$ non identiquement nulle sur E et soit $f \in L(D)$ alors :

$div(f)+D \geq 0$.

div(f) est un diviseur principal, il est donc de degré 0, d'où

$deg(D) = deg(div(f)+D)$.

Et comme $div(f)+D \geq 0$, $deg(div(f)+D) \geq 0$.

Ainsi $degD \geq 0$. Contradiction avec l'hypothèse $D < 0$.

Donc il y a seulement la fonction 0 dans L(D). □

Théorème 3.7 :(théorème de Riemann-Roch).

Il existe un entier g tel que pour tout diviseur D sur la courbe E on a :

$$l(D) \geq deg(D)+1-g.$$

avec égalité si $degD > 2g-2$.

L'entier g est appelé le genre de la courbe E.

Corollaire 3.8 :

Soit E une courbe elliptique et D un diviseur sur E, avec $degD \geq 1$ alors :

$l(D) = degD$.

Regardons les cas suivants :

1-Soit $P \in E$, alors $l((P)) = 1$.

L((P)) contient les fonctions constantes, ce qui montre qu'il n'y a aucune

fonction sur E ayant un seul pole en P.

2-Considérons le point $\theta \in E$, alors $l(2(\theta)) = 2$.

La fonction x a un pole double en θ donc $x \in L(2(\theta))$.

Nous avons que $\{1,x\}$ est une base de $L(2(\theta))$.

De même $\{1, x, y\}$ est une base de $L(3(\theta))$.

3-Les fonctions $1, x, y, x^2, xy, x^3, y^2$ sont dans $L(6(\theta))$ et $l(6(\theta)) = 6$.

Ceci veut dire que ces fonctions sont lineairement dépendantes sur K ; en effet l'équation de la courbe elliptique E nous donne la relation entre ces fonctions.

Lemme 3.9 :

Soit E une courbe elliptique, et $P, Q \in E$ alors :

$(P) \sim (Q)$ si et seulement si $P = Q$.

Preuve :

Si $P = Q$ alors $(P) \sim (Q)$.

Supposons maintenant que $(P) \sim (Q)$, d'après la définition 3.3 il existe $f \in K(E)$ telle que $(P)-(Q) = div(f)$.

Nous avons $f \in L((Q))$ et $l((Q)) = 1$, mais $L((Q))$ contient les fonctions constantes.

Ainsi f est constante et $P = Q$. □

Lemme 3.10 :

Soit D un diviseur sur E, de degré 0.

Alors : $D = (P)-(\theta)+div(f)$; pour un unique point de E et une fonction $f \in K(E)$.

Preuve :

Le diviseur D étant de degré 0, $l(D) = 0$ (cor 3.8).

Donc $l(D+(\theta)) = 1$.

Soit $f \in K(E)$ un générateur de $L(D+(\theta))$. Puisque $div(f)+D+(\theta) \geq 0$ on a :

$div(f) \geq -D-(\theta)$.

Or par hypothèse $degD = 0$ et $deg(div(f)) = 0$, alors : $div(f) = -D-(\theta)+(P)$

pour un $P \in E$.

ainsi $D \sim (P)-(\theta)$.

Unicité :

Soit $P' \in E$, P' a les mêmes propriétés que le point P.

Alors : $(P) \sim D+(\theta) \sim (P')$, donc $(P) \sim (P')$.

Le lemme 3.9 implique P = P', d'où l'unicité de P. □

Théorème 3.11 :

Soit E une courbe elliptique et $D = \sum_P n_p(P)$ un diviseur sur E, alors :

D est principal $\Leftrightarrow \sum_P n_p = 0$ dans Z et $\sum_P n_p (P) = \theta$ sur E.

Preuve :

Tout diviseur principal est de degré 0. Supposons que D est un diviseur principal.

Ceci implique : $D \sim 0$.

Considérons l'application surjective $\sigma : Div^\circ \to E$ qui vérifie :

Pour $D_1, D_2 \in Div^\circ(E)$; $\sigma(D_1) = \sigma(D_2) \Leftrightarrow D_1 \sim D_2$.

Et pour tout $P \in E$: $\sigma((P)-(\theta)) = P$.

$D = (P)-(\theta)+div(f)$ pour un certain P dans E, d'après le corollaire 3.10.

Ainsi $D \sim (P)-(\theta)$.

Donc : $D \sim 0 \Leftrightarrow \sigma(D) = \theta \Leftrightarrow \sum_P n_p \sigma((P)-(\theta)) = \theta \Leftrightarrow \sum_P n_p (P) = \theta$. □

Chapitre 4

Application à la cryptographie

Le domaine d'application des courbes elliptiques est très vaste, nous avons vu dans le chapitre 2 comment il est possible de tester la primalité d'un nombre et s'il n'est pas premier comment le factoriser.

Dans cette partie nous allons voir que les courbes elliptiques sont également utilisées en cryptographie.

Les schémas de Diffie-Hellman et d'El Gamel seront adaptés pour pouvoir être utilisés sur les courbes elliptiques.

1.Le protocole de Diffie et Hellman :

Supposons qu'Alice(A) et Bob(B) veulent s'échanger une clé k, Diffie et Hellman suggèrent l'échange suivant :

1-A et B choisissent, ensemble et publiquement un nombre premier p, et un entier $1 < a < p$.

2-A choisit secrètement $x_1 \in F_p^*$, et B choisit $x_2 \in F_p^*$.

A envoie à B a^{x1} et B calcule $k = (a^{x1})^{x2} = a^{x1x2} \bmod p$.

B envoie à A a^{x2} et A calcule $k = (a^{x2})^{x1} = a^{x1x2} \bmod p$.

k est leur clé secrète.

Si quelqu'un a espionné leur conversation, il connaît p, a, a^{x1} et a^{x2}. pour obtenir k, il doit pouvoir calculer x_1 en connaissant a, p et a^{x1}. On appelle ceci résoudre le logarithme discret.

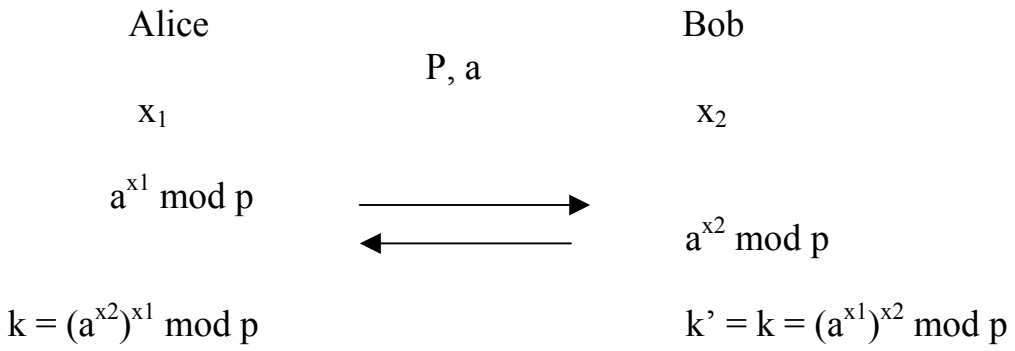


Figure 1 : Schéma de Diffie et Hellman classique.

Sur les courbes elliptiques le schéma est identique si ce n'est qu'Alice et Bob rendent publique une courbe elliptique E définie sur F_q et un point P sur $E(F_q)$.

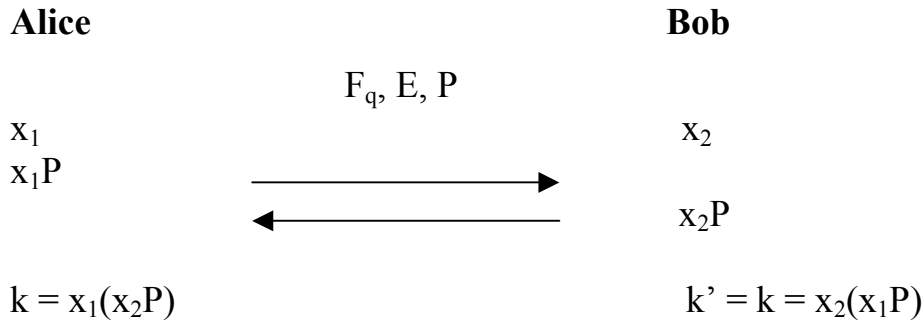


Figure 2 : Schéma de Diffie et Hellman elliptique.

2.Codage d'El Gamel :

L'échange dans le cryptosystème d'El-Gamel se fait de la façon suivante :

Le destinataire Bob choisit un nombre premier p et un générateur a du groupe multiplicatif $(Z / pZ)^*$. Il choisit secrètement un entier $1 \leq x_1 \leq p-2$ et calcule

$k_B = a^{x1}$.

Le destinateur Alice veut envoyer un message $M \in Z / pZ$ à Bob, elle choisit un entier x_2 et calcule $k_A \equiv a^{x2} \bmod p$ et $k_A' \equiv Mk_B^{x2} \bmod p$.

Alice envoie le couple (k_A , K_A') à Bob.

Bob retrouve le message en calculant $k_A' k_A^{-x1} \bmod p$.

La sécurité du système d'El Gamel repose sur la difficulté de résoudre le logarithme discret. Quand les valeurs de p et a sont très grande, il s'agit d'un problème difficile.

Alice		Bob
	P, a	
M, x_2		x_1
$k_A \equiv a^{x2} \bmod p$		
$k'_A \equiv Mk_B^{x2} \bmod p$		
$c = (k_A, k_A')$	⟶	$M \equiv k_A' k_A^{-x1} \bmod p$

Figure 3 : Codage d'El Gamel classique.

Avec les même notations qu'auparavant, l'analogue sur les courbes elliptiques est illustré par le schéma suivant :

Alice		**Bob**
	F_q, E, P	
M, x_2		x_1
$k_A = x_2P$		
$k_A' = M+x_2x_1P$		
$c = (k_A, k_A')$	⟶	$M' = M = k_A' - x_1x_2P$.

Figure 4 : codage d'El Gamel elliptique.

3.Le cryptosystème RSA :

La méthode de cryptographie RSA a été inventée en 1977 par Ron Rivest, Adi Shamir et Lan Adelman.

Si Bob souhaite recevoir des messages en utilisant le RSA il procède de la façon suivante :

Il choisit deux grands nombres premiers distincts p et q, un entier e premier avec le produit (p-1)(q-1) et un autre entier d tel que $ed \equiv 1 \bmod (p-1)(q-1)$.

Le couple (n, e) constitue la clé publique de Bob et le couple (n, d) constitue sa clé privée, avec n = pq.

Si Alice veut envoyer un message codé à Bob elle le représente sous la forme d'un ou plusieurs entiers M compris entre 0 et n-1.

Alice possède la clé publique (n, e) de Bob, elle calcule $C \equiv M^e \bmod n$ qu'elle transmet.

Bob reçoit C et calcule grâce à sa clé privée $D \equiv C^d \bmod n \equiv M^{ed} \bmod n \equiv M \bmod n$.

Les attaques du RSA se font essentiellement en factorisant l'entier n de la clé publique. La sécurité du système repose donc sur la difficulté de factoriser de grands nombres, c'est pour cette raison qu'il faut choisir des clés très grandes.

Remarque :

Le cryptosystème RSA n'a pas d'analogue sur les courbes elliptiques.

4.Le cryptosystème utilisant les courbes elliptiques :

Le principe de fonctionnement de ce cryptosystème est le suivant :

4.1Création et échange des clés :

Alice et Bob se mettent d'accord et publiquement sur une courbe elliptique $E(a, b, p) : y^2 = x^3+ax+b \bmod p$ et un point P sur la courbe.

Secrètement, Alice choisit un entier k_A et Bob un entier k_B.

Alice envoie à Bob le point k_AP et Bob envoie à Alice le point k_B. Chacun de leur coté, ils sont capables de calculer $k_A(k_BP) = k_B(k_AP) = (k_Ak_B)P$. Ce point de la courbe est leur clé secrète commune.

4.2Envoie du message :

Alice veut envoyer un message à Bob.

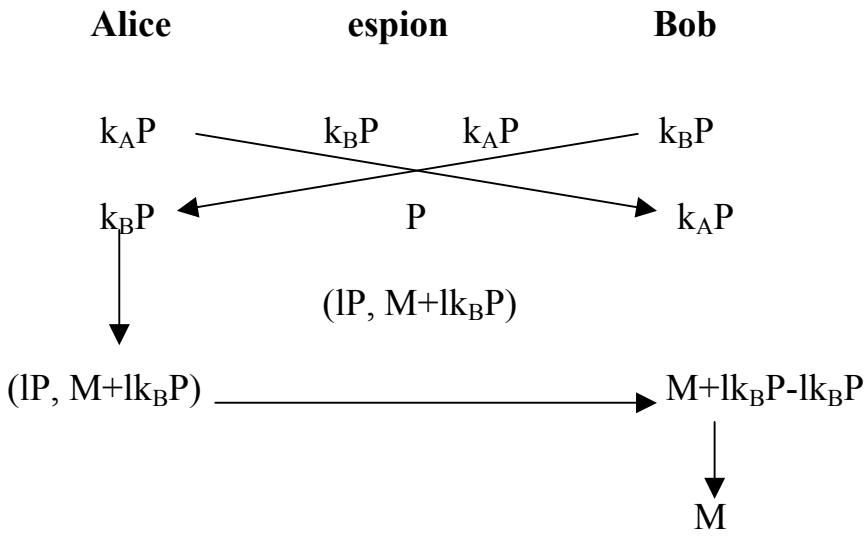
Elle commence par traduire son message en une suite de points (ils se sont mis d'accord sur la façon de transformer un texte en une suite de points de la courbe elliptique). On appelle M un de ces points.

Alice choisit de façon secrète un entier l et envoie à Bob le couple $(lP, M+lk_BP)$.

4.3Réception du message codé :

Bob reçoit le couple $(lP, M+lk_BP)$, il multiplie lP par sa clé secrète k_B et soustrait le résultat à $M+lk_BP$ il obtient donc M.

Schéma récapitulatif :



4.4Le problème du logarithme discret :

Si quelqu'un a espionné l'échange d'Alice et Bob il connaît E(a, b, p), P, k_AP et k_BP. Pour pouvoir calculer k_Ak_BP, il faut pouvoir calculer k_A connaissant P et k_AP.

Autrement dit, il doit pouvoir résoudre l'équation : kP = Q avec

P, Q $\in E(F_p)$ et k$\in$Z.

C'est ce que l'on appelle résoudre le logarithme discret sur la courbe elliptique.

k est appelé logarithme discret de Q en base P.

Exemple 8:

Calculer le logarithme discret de Q = (21, 13) en base P = (10, 12) sur la courbe

E(5, 11, 27).

On commence par calculer le point 2P = P+P = (x_{2p}, y_{2p}) mod 27.

$x_{2p} \equiv (3.10^2+5 / 2.12)^2 - 2.10 \bmod 27.$
$\equiv 15 \bmod 27.$

$y_{2p} \equiv -12 + (3.10^2+5 / 2.12)(10-15) \bmod 27.$
$\equiv 2 \bmod 7.$

2P = (15, 2)

3P = 2P+P = (x_{3p}, y_{3p}).

$x_{3p} \equiv (12-2 / 10-5) -15-10 \bmod 27.$
$\equiv 21 \bmod 27.$

$y_{3p} \equiv -2+(12-2 / 10-5)(15-21) \bmod 27.$
$\equiv 13 \bmod 27.$

3P = (21, 13).

Donc le logarithme discret de Q en base P est 3.

Exemple 9 :

Dans cet exemple, on va créer une courbe elliptique sur un corps fini. Notre point de base Q est choisi aléatoirement sur la courbe, l'autre point P est sélectionné comme un multiple aléatoire de Q. En utilisant la fonction Log on trouve le multiplicateur m et ensuite on vérifie que la solution est correcte.

```
>k :=GF(NextPrime(3265325874125)) ;

>k ;
Finite field of size 3265325874133
>E :=EllipticCurve([Random(k),Random(k)]) ;
```

```
>E ;
Elliptic curve defined by y^2=x^3+1420387103330*x+579949386593 over
GF(3265325874133)

>Q :=Random(E) ;
Q ;
(1555066603586 :36897426276 :1)

>FactoredOrder(Q) ;
[<2,3>, <1453,1>, <280912153,1>]

>P :=Random(Order(Q))*Q ;

>P ;
(2232623733837 :868822072610 :1)

>m :=Log(Q,P) ;

>m ;
1447157775959

>m*Q-P ;
(0 :1 :0)
```

4.5Sécurité :

La sécurité du cryptosystème utilisant les courbes elliptiques repose sur le fait que chaque courbe détermine un groupe différent dont on ne sait même pas trouver l'ordre par un algorithme rapide, et encore moins la structure contrairement au groupe Z / pZ.

Ce qui rend la résolution du logarithme discret sur les courbes elliptiques plus difficile que celle sur le groupe $(Z / pZ)^*$.

Dans la partie suivante nous verrons comment le problème du logarithme discret sur une courbe elliptique $E(F_q)$ où $q = p^n$, peut être réduit à un problème du logarithme discret sur le corps fini F_q et sur une extension F_q^k de F_q.

5.Accouplement de Weil :

Soit E(K) une courbe elliptique sur le corps fini $K = F_q$, $q = p^n$ avec p un nombre premier.

$E[m] = \{P \in E(K^{alg}) : mP = \theta \text{ sur } E(K^{alg})\}$ où K^{alg} désigne la clôture algébrique de K.

Soit $D = \sum_{p \in E} n_p(P)$ un diviseur sur E(K). le support du diviseur D est l'ensemble des points $P \in E$ tel que $n_p \neq 0$.

Soit $f \in K^{alg}(E)$ une fonction rationnelle telle que D et div(f) ont des supports disjoints.

Ceci nous permet de définir $f(D) = \prod_{p \in supp\ D} f(D)^{np}$, puisque f(p) est définis pour $p \in$ supp(D).

On fixe un entier m premier avec p, et on note $\mu_m \subset (K^{alg})^*$ le groupe des racines $m^{ième}$ de l'unité.

Soit $P, Q \in E[m]$ et A, B deux diviseurs de degré 0 à supports disjoints tels que :

$A \sim (P)-(\theta)$ et $B \sim (Q)-(\theta)$.

Le théorème 3.11 implique l'existence de deux fonctions rationnelles

$f_A, f_B \in K^{alg}(E)$ telles que : $div(f_A) = mA = m(P)-m(\theta)$ et

$div(f_B) = mB = m(Q)-m(\theta)$.

Puisque A et B ont deux supports disjoints alors $div(f_A)$et B (respectivement :

div(f_B) et A) ont des supports disjoints.

Alors on définit l'application :

$$e_m : E[m]\times E[m] \rightarrow (K^{alg})^* \text{ avec } e_m(P, Q) = f_A(B) / f_B(A).$$

Théorème 4.1 :

L'application e_m est bien définie et $e_m(P, Q) \in \mu_m$.

Pour la preuve du théorème 4.1, on applique la loi de réciprocité de Weil :

Si $f, g \in K^{alg}(E)^*$ vérifient $supp(div(f)) \cap supp(div(g)) = \phi$; alors

$f(div(g)) = g(div(f))$.

Proposition 4.2 :

L'accouplement e_m de Weil vérifie les propriétés suivantes :

1-$\forall P \in E[m]$; $e_m(P, P) = 1$.

2-e_m est antisymétrique :

$\forall P,Q \in E[m]$; $e_m(P, Q) = e_m(Q, P)^{-1}$.

3-$e_m$est bilinéaire :

$\forall P,Q$ et $R \in E[m]$; $e_m(P+Q, R) = e_m(P, R)\, e_m(Q, R)$

et $e_m(P, Q+R) = e_m(P, Q)\, e_m(P, R)$.

4-e_m est non dégénérée :

Soit $P \in E[m]$, alors $e_m(P, \theta) = 1$.

Si $e_m(P, Q) = 1$ pour tous $Q \in E[m]$, alors $P = \theta$.

5-Si $E[m] \subseteq E(K)$ alors $e_m(P, Q) \in K$, $\forall P, Q \in E[m]$ et $\mu_m \subset K^*$.

6.Calcul de l'accouplement e_m de Weil :

Soit l'entier m tel que pgcd(m, p) = 1, et $P, Q \in E[m]$.

On veut calculer $e_m(P, Q)$.

Considérons les points $T, U \in E$ tels que $P+T \neq U, Q+U$ et $T \neq U, Q+U$.

Posons A = (P+T)-(T).

D'après le théorème 3.11 on a $A \sim (P)-(\theta)$ puisque deg A = 0 et $A-P+\theta = \theta$ sur la courbe.

On pose aussi B = (Q+U)-(U), alors $B \sim (Q)-(\theta)$

Soient $f_A, f_B \in K^{alg}(E)$ avec $\begin{cases} div(f_A) = m(P+T)-m(T) \\ \text{et} \\ div(f_B) = m(Q+U)-m(U). \end{cases}$

Les fonctions f_A et f_B peuvent être calculées par l'algorithme [8,p56].

On a : $e_m = f_A(B) / f_B(A)$.

$= f_A((Q+U)-(U)) / f_B((P+T)-(T))$.

$= f_A(Q+U)\, f_B(T) / f_A(U)\, f_B(P+T)$.

Exemple 10 :

Dans cet exemple, on choisit une courbe elliptique aléatoire sur un corps fini choisi aléatoirement. On calcul l'ensemble des points de 12 torsion sur la courbe elliptique, et on calcul l'accouplement de Weil de deux point de cet ensemble.

```
> p:=NextPrime(215473);
> p;
215483

> k:=GF(p);
> k;
Finite field of size 215483

> E:=EllipticCurve([Random(k),Random(k)]);
> E;
Elliptic Curve defined by y^2 = x^3 + 28174*x + 176691 over GF(215483)

> Order(E);
214912

> A:=DivisionPoints(Id(E),12);
> A;
[ (0 : 1 : 0), (34908 : 0 : 1), (39134 : 0 : 1), (117394 : 17610 : 1), (117394 :

197873 : 1), (141441 : 0 : 1), (167905 : 3542 : 1), (167905 : 211941 : 1) ]

> Q:=Random(A);
> Q;
(34908 : 0 : 1)

> R:=Random(A);
> R;
(141441 : 0 : 1)

> 12*Q;
(0 : 1 : 0)

> 12*R;
(0 : 1 : 0)

> e:=WeilPairing(Q,R,12);
>e;
1
> time;
Time: 0.000
```

Avant de passer à la réduction nous avons besoins de quelques résultats utiles :

Soit E(K) une courbe elliptique sur le corps fini $K = F_q$, $q = p^n$ avec p un nombre premier.

1-$E(F_q) \cong Z_{n1} \times Z_{n2}$, où $n_2 \mid n_1$ et $n_2 \mid q-1$.

2-Soit $m \in Z-\{0\}$, $E(F_q)[m]$ le sous groupe des points de n torsion de $E(F_q)$.

Si (m, q) = 1, alors $E[m] \cong Z_m \times Z_m$.

3-Pour toute courbe $E(F_q)$ il existe un k tel que $E[n] \subset E(F_q^k)$.

Nous avons aussi besoins des lemmes suivants :

Soit < P > le sous groupe de $E(F_q)$ engendré par le point P.

Lemme 4.3 :

Soit $E(F_q)$ une courbe elliptique telle que $E[n] \subseteq E(F_q)$; avec n vérifiant :

pgcd(n, q) = 1.

Soit $P \in E[n]$ un point d'ordre n. Alors < P > est un sous groupe de E[n].

$\forall P_1 , P_2 \in E[n]$; P_1et P_2 sont dans le même sous groupe < P > dans E[n] si

et seulement si $e_n(P, P_1) = e_n(P, P_2)$.

Lemme 4.4 :

Soit $E(F_q)$ une courbe elliptique ayant une structure du groupe $Z_{n1} \times Z_{n2}$ avec

$n_2 \mid n_1$.

P un élément d'ordre maximum n_1, alors :

$\forall P_1, P_2 \in E(F_q)$, P_1 et P_2 sont dans le même sous groupe < P > si et

seulement si $e_n(P, P_1) = e_n(P, P_2)$.

Lemme 4.5 :

Soit G un groupe, $\alpha \in G$ et $n \in N$ avec $n = \prod_{i=1}^{k} p_i^{\beta i}$.

Alors α est d'ordre n si et seulement si : 1- $\alpha^n = 1$.

2- $\alpha^{n/pi} \neq 1$, $\forall i : 1 \leq i \leq k$.

7.Réduction :

Soit $P \in E(F_q)$ ($E(F_q) \cong Z_{n1} \times Z_{n2}$ avec $n_2 \mid n_1$) un point d'ordre n, où n divise n_1.

On suppose que n est connu.

Soit R un autre point de la courbe elliptique $E(F_q)$.

On veut résoudre le problème du logarithme discret de R en base P sur E .

Autrement dit qu'el est l'entier l, $0 \leq l \leq n-1$ qui vérifie $lP = R$.

Puisque $e_n(P, P) = 1$ on a : $R \in < P > \Leftrightarrow nR = \theta$ et $e_n(P, R) = 1$.

Supposons que $R \in <P>$.

7.1.Algorithme 1 :

Entrée : $P \in E(F_q)$ tel que P est d'ordre n_1 sur E, et $R = lP$.

Sortie : un entier l' tel que $l' \equiv l \bmod n'$ et $n' \mid n_2$.

Etape 1 : choisir un point $T \in E(F_q)$.

Etape 2 : calculer $\alpha = e_{n1}(P, T)$ et $\beta = e_{n1}(R, T)$.

Etape 3 : calculer l' le logarithme discret de β en base α dans F_q ;

c'est à dire : calculer l' tel que $\alpha = \beta^{l'}$ dans F_q.

Théorème 4.6 :

L'algorithme 1 calcule $l' \equiv l \bmod n'$ avec $n' \mid n_2$.

Preuve :

Soit $G \in E(F_q)$ un point d'ordre n_2 tel que $E(F_q)$ soit engendrée par (P, G).

Et soit $T = c_1P + c_2G$ avec $c_1, c_2 \in Z$.

Alors,

$$\alpha^{n_2} = e_{n1}(P, T)^{n_2} = e_{n1}(P, P)^{c1n2}\, e_{n1}(P, c_2n_2G) = e_{n1}(P, \theta) = 1.$$

Ainsi l'ordre de α, noté n', divise n_2.

Puisque $n_2 \mid q-1$, $x^{n2} - 1 = 0$ a toutes ces racines dans F_q. D'où $\alpha \in F_q$.

On a :

$$\beta = e_{n1}(R, T) = e_{n1}(lP, T) = e_{n1}(P, T)^l = \alpha^l = \alpha^{l'}, \text{ où } l' \equiv l \bmod n'.$$

On peut déterminer l' en calculant le logarithme discret de β en base α dans F_q.

□

Soit k le plus petit entier positif tel que $E[n] \subseteq E(F_q^k)$.

Théorème 4.7:

Soit P dans E un point d'ordre n.

Il existe $Q \in E[n]$ tel que $e_n(P, Q)$ est une racine primitive $n^{ième}$ de l'unité.

Théorème 4.8 :

Soit $Q \in E[n]$ tel que $e_n(P, Q)$ est une racine primitive $n^{ième}$ de l'unité.

L'application $f : < P > \rightarrow \mu_n$ définie par $f(R) = e_n(R, Q)$ est un isomorphisme de groupe.

7.2Algorithme 2 :

Entrée : un élément $P \in E(F_q)$ d'ordre n et $R \in < P >$.

Sortie : un entier l tel que $R = lP$.

Etape 1 : trouver le plus petit entier k tel que $E[n] \subseteq E(F_q^k)$.

Etape 2 : trouver $Q \in E[n]$ tel que $\alpha = e_n(P, Q)$ soit d'ordre n.

Etape 3 : calculer $\beta = e_n(R, Q)$.

Etape 4 : calculer l, le logarithme discret de β en base α dans F_q^k.

Remarque :

Le problème du logarithme discret sur les courbes elliptiques supersingulière ([4] p.1642) est efficacement réductible au problème du logarithme discret dans une extension quadratique du corps fondamental (c'est à dire k=2).

Parmi ces courbes on a :

a) $y^2+y = x^3+b$ sur F_2^m avec m impair.

b) $y^2 = x^3-ax$ sur F_p , où $p > 3$ est un nombre premier et $p \equiv 3 \bmod 4$.

c) $y^2 = x^3+b$ sur F_p , où $p > 3$ est un nombre premier et $p \equiv 2 \bmod 3$.

De nos jours, le problème du logarithme discret dans F_q est efficacement résoluble, pour un q premier et plus petit que 2^{200} et pour $q = 2^m$ avec $m < 600$. Donc les courbes ci-dessus sont peu sures pour des applications cryptographiques.

Exemple 11:

L'exemple suivant démontre la réduction de Menezes, Okamoto, et Vanstone

(MOV)[4] d'un logarithme discret sur une courbes elliptiques supersingulière en logarithme discret dans un corps fini.

```
> p:=NextPrime(3^124);

> n:=p+1;
> n;
1455578342930689280434675661902780082182495258305659396184 90

> Factorization(n);
[ <2, 1>, <3, 2>, <5, 1>, <180317, 1>, <262199473, 1>, <39504363995133913, 1>,
<865923475887669700104067517, 1> ]

> time;
Time: 0.000

> E0:=SupersingularEllipticCurve(GF(p));

> G<x>, f:=AbelianGroup(E0);
> G;
Abelian Group isomorphic to Z/145557834293068928043467566190278008218249525830\

65939618490
Defined on 1 generator
Relations:
   145557834293068928043467566190278008218249525830565939618490*x = 0

> n eq #G;
true

> P0:=f(x);

> E1:=BaseExtend(E0,GF(p^2));
> P1:=E1!P0;
```

```
> repeat

> Q1:=Random(E1);

> Z1:=WeilPairing(P1,Q1,n);

> until Order(Z1) eq n;

> IsOrder(Q1,n);
true

> r:=7654321;

> Z2:=WeilPairing(r*P1,Q1,n);

> Z1^r eq Z2;
true

> WeilPairing(P1,r*P1,n);
1

> time;
Time: 0.000
```

Bibliographie :

[1] **Cyril Banderier**. crible quadratique, fractions continuées et consorts, 1996/1997.
http://pauillac.inria.fr/algo/banderier/Facto/facto.html

[2] **Richard P. Brent**. Some integer factorization algorithms using elliptic curves. Australian computer science communications 8 (1986), 149-163.
http://web.comlab.ox.ac.uk/oucl/work/richard.brent/pub/pub102.html.

[3] **Richard P. Brent**. Factorization of the tenth Fermat number. Mathematics of computation 68, 225 (1999), 429-451.
http://web.comlab.ax.ac.uk/oucl/work/richard.brent/pd/rpb161pdf

[4] **W.Fulton**. Algebraic curves, Benjamin, New York, 1969.

[5] **Marc Joye**. Introduction élémentaire à la théorie des courbes elliptiques. Technical report CG, Juin 1995.
http://www.dice.ucl.ac.be/crypto/

[6] **H.W. Lenstra**. Factoring integers with elliptic curves, annals of mathematics, 126(1987), 649-673.

[7] **Alfred Menezes, Tatsuaki Okamoto, et Scott A. Vanstone**. Reducing

elliptic curve logarithms to logarithms in a finite field, IEEE Transactions On Information Theory, vol 39, No 5, 1993.

[8] **Samuel Mimram**. Courbes elliptiques et factorisation, 2001-2002.
http://perso.ens-lyon.fr/samuel.mimram/prepa/tipe-ecc.pdf.

[9] **René Schoof**. Elliptic curves over finite fields and the computation of square roots mod p, mathematics of computation, 44(1985),484-494.

[10] **Silverman**. The arithmetic of elliptic curves, springer- verlag 1986.

[11] **Annagret Weng**. Elliptic curves, Edinburgh, March 1998.

Glossaire :

$D \sim D'$ équivalence linéaire de diviseurs, 35

L(D) espace vectoriel de toutes les fonctions rationnelles f telles que

$div(f)+D \geq 0$, 35

l(D) dimension de L(D), 36

RSA Ron Rivest, Adi Shamir, et Lan Adelman, 45

K^{alg} clôture algébrique de K, 50

$E(K^{alg})$ courbe elliptique sur la clôture algébrique de K, 50

$E[m]$ ensemble des points de m torsion sur $E(K^{alg})$, 50

Supp(D) support du diviseur D, 50

μ_m groupe des racines $m^{ième}$ de l'unité, 50

$e_m(P,Q)$ accouplement de Weil de P et Q, 51

$< P >$ sous groupe de $E(F_q)$ engendré par P, 54

MOV Menezes, Okamoto, et Vanstone, 58