MA^2 - Master Algèbre Appliquée
Université Versailles-Saint-Quentin

MSMA310
19 Mars 2008

Méthode arithmético-géométrique pour le comptage de points d'une courbe elliptique définie sur $\mathbf{F}_{2^d}$

V. Vitse

Table des matières

1 Motivations

L'ensemble des points rationnels d'une courbe elliptique définie sur un corps fini est naturellement muni d'une loi de groupe dans lequel le problème du logarithme discret

s'avère difficile. On ne connait en effet que des algorithmes de complexité asymptotique exponentielle en $\log q$ sur $E(\mathbf{F}_q)$, contrairement aux groupes multiplicatifs de $\mathbf{F}_q$ pour lesquels il existe des algorithmes sous-exponentiels.

Il est donc tentant de transformer les systèmes cryptographiques fondés sur l'exponentielle modulaire en leurs analogues elliptiques, ce qui a été proposé d'abord par Miller en 1985 [Mil], puis par Koblitz deux ans plus tard [Kob]. Pour la mise en place de schémas tels que le protocole d'échange de clés de Diffie-Hellman, la signature d'El Gamal, etc. sur courbe elliptique, il est indispensable de pouvoir trouver un sous-groupe cyclique de $E(\mathbf{F}_q)$ dont l'ordre est divisible par un grand nombre premier et donc a fortiori être capable de calculer la cardinalité de $E(\mathbf{F}_q)$.

L'algorithme de comptage de points d'une courbe elliptique sur un corps fini publié par Shoof en 1985 [Shoof], fut le premier algorithme à atteindre une complexité polynomiale. De nombreuses améliorations ont été ensuite apportées par Atkin, Elkies, Couveignes... donnant naissance à l'algorithme SEA. A titre d'exemple, en 1995 le calcul de la cardinalité d'une courbe définie sur $\mathbf{F}_{2^{155}}$ est réalisable en une dizaine de minutes sur une station de travail classique (cf. [Ler] pour un panorama sur le sujet).

Puis en 2000, une nouvelle famille d'algorithmes utilisant le relèvement canonique sur les $\mathbf{Z}_q$-adiques d'une courbe elliptique apparaît, notamment l'algorithme de Satoh [Satoh] qui est le premier à fonctionner alors plus rapidemment que SEA avec un temps de calcul en $O(n^{3+\epsilon})$ et une consommation mémoire en $O(n^3)$.

L'algorithme qui est présenté dans ce mémoire, a été découvert par Mestre [Mes] à la même époque. Il est basé sur un calcul de suite arithmético-géométrique et permet d'obtenir le nombre de points rationnels d'une courbe définie en caractéristique 2 avec une complexité en temps similaire à celle de l'algorithme de Satoh, mais une consommation mémoire réduite à $O(n^2)$. Il a en plus l'avantage d'être particulièrement simple à formuler et à implémenter, même s'il utilise des outils mathématiques élaborés tels que le relèvement canonique d'une courbe elliptique dans une extension du corps des 2-adiques.

2 Approche standard du comptage de points sur courbes elliptiques

Soient $p \in \mathbf{N}$ un nombre premier, $d \in \mathbf{N}^*$, $q = p^d$ et E une courbe elliptique définie sur le corps fini $\mathbf{F}_q$ à q éléments.

On s'intéresse au calcul de la cardinalité de l'ensemble des points rationnels $E(\mathbf{F}_q)$ de la courbe.

2.1 Nombre de points rationnels

La courbe E étant définie sur $\mathbf{F}_q$, le q-ième morphisme de Frobenius définit un endomorphisme de E :

$$\Phi_q : E \to E$$
$$\Phi_q([X : Y : T]) = [X^q : Y^q : T^q]$$

Etant donné que les points fixes de l'application

$$\begin{array}{ccc} \overline{\mathbf{F}}_q & \to & \overline{\mathbf{F}}_q \\ x & \mapsto & x^q, \end{array}$$

sont exactement les éléments de $\mathbf{F}_q$, on peut caractériser les points rationnels de la courbe par la propriété suivante :

Propriété 2.1. *Soit $P \in E$ un point de la courbe, alors P est rationnel sur $\mathbf{F}_q$ si et seulement si $\Phi_q(P) = P$.*
En particulier,

$$E(\mathbf{F}_q) = \ker(1 - \Phi_q)$$

Or le morphisme $(1 - \Phi_q)$ étant séparable ([Sil] p.83), on a

$$\#\ker(1 - \Phi_q) = \deg_s(1 - \Phi_q) = \deg(1 - \Phi_q)$$

On peut donc ramener le problème du comptage de points rationnels d'une courbe elliptique au calcul du degré de $1 - \Phi_q$.

2.2 Module de Tate et conjecture de Weil pour les courbes elliptiques

Pour pouvoir calculer ce degré, on va introduire le module de Tate T_l (et donc l'anneau local $\mathbf{Z}_l$ des l-adiques) et considérer l'action du q-ième morphisme de Frobenius sur ce module.

Module de Tate

On rappelle la construction de la limite projective dans le cas d'une famille indexée par $\mathbf{N}$:

Définition 2.2. *On définit un système projectif $(G_i, \pi_i)_{i\in\mathbf{N}}$ indexé par $\mathbf{N}$ comme la donnée d'une famille de groupes et d'homomorphismes $\pi_i : G_{i+1} \to G_i$.*

Sa limite projective est le sous-groupe de $\prod G_i$ défini par

$$\varprojlim G_n = \{x = (x_1, x_2, \ldots, x_n, \ldots) : x_n \in G_n, \pi_n(x_{n+1}) = x_n\}$$

Remarque : on généralise cette notion de limite projective au cas où les G_i sont des anneaux, et en particulier on obtient une structure d'anneau sur la limite projective.

Exemple : *anneau $\mathbf{Z}_l$ des l-adiques*
Si l'on considère l'homomorphisme naturel d'anneaux $\pi_n : \mathbf{Z}/l^{n+1}\mathbf{Z} \to \mathbf{Z}/l^n\mathbf{Z}$ (réduction modulo l^n), la limite projective $\varprojlim \mathbf{Z}/l^n\mathbf{Z}$ est un anneau appelé anneau des l-adiques et noté $\mathbf{Z}_l$.
L'anneau $\mathbf{Z}_l$ est de caractéristique 0, et intègre si l est premier (cf. section 2.3).

De la même façon, on construit le **module de Tate** $T_l(E)$ d'une courbe elliptique E en considérant la limite projective des groupes de l^n-torsion reliés par l'homorphisme $[l] : E[l^{n+1}] \to E[l^n]$, avec l entier premier :

Définition 2.3.

$$T_l(E) = \varprojlim E[l^n]$$

Comme chaque $E[l^n]$ est naturellement muni d'une structure de $\mathbf{Z}/l^n\mathbf{Z}$-module, le module de Tate $T_l(E)$ a une structure de $\mathbf{Z}_l$-module :

en effet,
si $a_n \in \mathbf{Z}/l^n\mathbf{Z}$ et si $u_n \in E[l^n]$, alors $[a_n]u_n$ est bien défini,
et si $a = (a_1, \ldots, a_n, \ldots) \in \mathbf{Z}_l$ et $u = (u_1, \ldots, u_n, \ldots) \in T_l(E)$, alors
$[l][a_n]u_n = [a_n][l]u_n = [a_n]u_{n-1} = [a_{n-1}]u_{n-1}$. Ainsi on peut poser

$$a.u = ([a_1]u_1, \ldots, [a_n]u_n, \ldots)$$

La proposition suivante explicite la structure de $T_l(E)$:

Proposition 2.4. *Si $l \neq p$ premier, alors*

$$T_l(E) \simeq \mathbf{Z}_l \times \mathbf{Z}_l$$

En particulier,

$$End(T_l(E)) \simeq \mathcal{M}_2(\mathbf{Z}_l)$$

Démonstration. Cette structure est héritée directement de celle des modules de l^n-torsion lorsque l est premier à la caractéristique du corps :

$$E[l^n] \simeq \mathbf{Z}/l^n\mathbf{Z} \times \mathbf{Z}/l^n\mathbf{Z}$$

□

De part la simplicité de sa structure, le module de Tate va être utile pour l'étude des endomorphismes de E, en particulier pour le q-ième morphisme de Frobenius.

En effet, si $\varphi \in End(E)$, comme $\varphi \circ [m] = [m] \circ \varphi$ (pour m entier quelconque), φ induit un homomorphisme de $E[l^n]$ vers $E[l^n]$.

Et comme $[l] \circ \varphi = \varphi \circ [l]$, ceci induit également un morphisme $\varphi_l : T_l(E) \to T_l(E)$ qui est $\mathbf{Z}_l$-linéaire sur le module de Tate. En résumé :

Proposition 2.5. *Soit $l \neq p$ premier, alors il existe un morphisme naturel*

$$\begin{array}{ccc} End(E) & \to & End(T_l(E)) \simeq \mathcal{M}_2(\mathbf{Z}_l) \\ \varphi & \mapsto & \varphi_l \end{array}$$

Ainsi, en faisant agir Φ_q sur $T_l(E)$, on obtient le polynôme caractéristique du q-ième Frobenius

$$\chi(\Phi_{q,l})(X) = X^2 - \mathrm{Tr}(\Phi_{q,l})X + \det(\Phi_{q,l}) \in \mathbf{Z}_l[X]$$

Conjecture de Weil sur les courbes elliptiques

Le résultat essentiel de ce paragraphe consiste à montrer que le polynôme caractéristique du q-ième Frobenius est en fait à coefficients dans $\mathbf{Z}$ et indépendant de l. Ceci nous permettra de relier le calcul du nombre de points rationnels de la courbe à la trace du q-ième Frobenius.

Théorème 2.6. $\det(\Phi_{q,l}) = \deg(\Phi_q) = q$ *et* $\mathrm{Tr}(\Phi_{q,l}) = 1 + q - \deg([1] - \Phi_q)$.
En particulier, la trace et le déterminant de Φ_q *ne dépendent pas de* l*, et le nombre de points rationnels de la courbe s'obtient avec le calcul de la trace du* q*-ième morphisme de Frobenius.*

Démonstration. On commence par prolonger la notion de couplage de Weil définie sur les groupes de l^n-torsion pour tout $n \in \mathbf{N}^*$

$$e_{l^n} : E[l^n] \times E[l^n] \to \mu_{l^n}$$

à un couplage de Weil définie sur le module de Tate :

$$e : T_l(E) \times T_l(E) \to T_l(\mu)$$

où $T_l(\mu)$ est obtenu en prenant la limite projective des groupes des racines l^n-ièmes reliés par l'exponentiation par l : $\mu_{l^{n+1}} \to \mu_{l^n},\ \xi \mapsto \xi^l$.

L'essentiel de la preuve du théorème repose alors sur le résultat suivant ([Sil] p.99) :

Proposition 2.7. *Le couplage de Weil*

$$e : T_l(E) \times T_l(E) \to T_l(\mu)$$

est une forme bilinéaire alternée et non dégénérée, et si $\varphi \in End(E)$ *alors* φ *et son isogénie duale* $\widehat{\varphi}$ *sont adjointes pour le couplage.*

On utilise cette proposition pour démontrer le lemme suivant :

Lemme 2.8. *Soit* $\varphi \in End(E)$ *un endomorphisme quelconque. Alors,*

$$\det \varphi_l = \deg \varphi$$

Démonstration. Soient (v_1, v_2) une $\mathbf{Z}_l$-base de $T_l(E)$ et A la représentation matricielle de φ_l dans la base (v_1, v_2) :

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

On utilise la bilinéarité et l'antisymétrie de e ([Sil] p.135) :

$$\begin{aligned} e(v_1, v_2)^{\deg \varphi} &= e([\deg \varphi]_l v_1, v_2) \\ &= e((\widehat{\varphi}\varphi)_l v_1, v_2) \\ &= e(\widehat{\varphi}_l \varphi_l v_1, v_2) \\ &= e(\varphi_l v_1, \varphi_l v_2) \\ &= e(av_1 + cv_2, bv_1 + dv_2) \\ &= e(v_1, v_1)^{ab} e(v_1, v_2)^{ad} e(v_2, v_1)^{cb} e(v_2, v_2)^{cd} \\ &= e(v_1, v_2)^{ad-bc} \\ &= e(v_1, v_2)^{\det(A)} \end{aligned}$$

et par non dégénérescence de e, on obtient le résultat. □

En appliquant le lemme à Φ_q et $[n] - \Phi_q$, où $n \in \mathbf{Z}$, on obtient :

$$\det \Phi_q = \deg \Phi_q = q$$

et

$$\deg([n] - \Phi_q) = \begin{vmatrix} n-a & -b \\ -c & n-d \end{vmatrix} = n^2 - n \operatorname{Tr}(\Phi_q) + \det(\Phi_q)$$

où $\mathrm{M}(\Phi_q) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ est la représentation matricielle de Φ_q dans une $\mathbf{Z}_l$-base de $T_l(E)$.

En particulier pour $n = 1$,

$$\operatorname{Tr}(\Phi_q) = 1 + q - \deg([1] - \Phi_q)$$

□

Dans la preuve, on a en fait montré un résultat plus général pour le polynôme caractéristique du q-ième Frobenius :

Corollaire 2.9. *Pour tout $n \in \mathbf{Z}$,*

$$\deg([n] - \Phi_q) = \chi(\Phi_q)(n) = n^2 - n \operatorname{Tr}(\Phi_q) + \det(\Phi_q)$$

Ceci va nous permettre de retrouver le théorème de Hasse donnant une bonne approximation du nombre de points rationnels de la courbe elliptique E :

Théorème 2.10 (Hasse)**.**

$$|\operatorname{Tr}(\Phi_q)| \leq 2\sqrt{\det \Phi_q}$$

En particulier, on peut approximer le nombre de points rationnels de la courbe E :

$$1 + q - 2\sqrt{q} \leq \#E(\mathbf{F}_q) \leq 1 + q + 2\sqrt{q}$$

Démonstration. Avec le corollaire précédent, on constate que

$$\forall n \in \mathbf{Z}, \chi(\Phi_q)(n) = \deg([n] - \Phi_q) \geq 0$$

On va généraliser ce résultat en montrant que le polynôme caractéristique de Φ_q est en fait positif sur tout $\mathbf{Q}$:
soient $(m, n) \in \mathbf{Z} \times \mathbf{Z}^*$, d'après le lemme 2.8

$$\frac{1}{n^2} \deg([m] - [n]\Phi_q) = \frac{1}{n^2} \det(mI_2 - nM(\Phi_q)) = \det\left(\frac{m}{n} I_2 - M(\Phi_q)\right) = \chi(\Phi_q)\left(\frac{m}{n}\right)$$

ce qui montre bien que $\chi(\Phi_q)(x) \geq 0, \forall x \in \mathbf{Q}$.

Son discriminant $\Delta = (\operatorname{Tr}(M(\Phi_q)))^2 - 4q$ vérifie donc $\Delta \leq 0$, ce qui donne bien l'encadrement cherché. □

Le nombre de points rationnels d'une courbe elliptique E peut donc s'obtenir à partir du calcul à une certaine précision de la trace du q-ième Frobenius :
comme $|\operatorname{Tr}(\Phi_q)| \leq 2p^{\lceil \frac{d}{2} \rceil}$, il suffit de connaître $\operatorname{Tr}(\Phi_q)$ modulo $2^2 p^{\lceil \frac{d}{2} \rceil}$ pour en déduire sa valeur exacte.

2.3 L'idée de Satoh

On utilise le résultat suivant dû à Satoh, qui permet de calculer facilement la trace d'un endomorphisme en regardant son action sur l'invariant différentiel du relevé canonique de E.

Théorème 2.11 ([Satoh], [CoFr])**.** *Soit $\mathcal{E}$ une courbe elliptique définie sur un corps de caractéristique 0, et soit ω une forme différentielle holomorphe sur $\mathcal{E}$. Pour tout $f \in End(\mathcal{E})$, on définit $\lambda_f = \frac{f^*(\omega)}{\omega}$. Alors λ_f est une racine du polynôme caractéristique de f et en particulier,*

$$\mathrm{Tr}(f) = \lambda_f + \frac{\deg(f)}{\lambda_f}$$

Ce résultat n'étant vérifié que sur un corps de caractéristique 0, l'idée consiste à introduire le corps $\mathbf{Q}_q$ des q-adiques et le relèvement canonique $\mathcal{E}$ de la courbe E défini sur $\mathbf{F}_q$:

Définition 2.12. *Le relèvement canonique de la courbe elliptique ordinaire E (i.e., lorsque $p = 2$, dont le j-invariant est non nul) est une courbe elliptique $\mathcal{E}$ sur $\mathbf{Q}_q$ satisfaisant :*
- *la réduction de $\mathcal{E}$ modulo p est E,*
- *l'homomorphisme d'anneaux $End(\mathcal{E}) \to End(E)$ induit par la réduction modulo p est un isomorphisme.*

On détaille dans la suite la construction du corps $\mathbf{Q}_q$, ainsi que certaines de ses propriétés utiles pour les calculs algorithmiques qui vont suivre, puis on introduit la substitution de Frobenius $\Sigma \in End(\mathcal{E})$ qui permettra de relever le morphisme de Frobenius défini sur E.

Corps des p-adiques

On rappelle qu'un entier p-adique est une suite $x = (x_1, x_2, \ldots)$ où $x_n \in \mathbf{Z}/p^n\mathbf{Z}$ est tel que $x_{n+1} = x_n \bmod p^n$. L'ensemble des p-adiques est noté $\mathbf{Z}_p$, la somme et le produit étant définis coordonnées par coordonnées de manière naturelle.

Propriété 2.13. $\mathbf{Z}_p$ *est un anneau de valuation discrète de corps résiduel $\mathbf{F}_p$ et de caractéristique 0.*

Démonstration.
- On commence par voir que $\mathbf{Z}_p$ est un anneau local :
  $\mathbf{Z}_p^* = \{x \in \mathbf{Z}_p : x \bmod p \neq 0\}$, donc $\mathbf{Z}_p \setminus \mathbf{Z}_p^* = p\mathbf{Z}_p$ est un idéal de $\mathbf{Z}_p$, il est maximal et principal.
- $\mathbf{Z}_p$ est intègre :
  Par l'absurde, on suppose qu'il existe $x, y \in \mathbf{Z}_p$ non nuls tels que $x.y = 0$.
  On considère $n \in \mathbf{N}$ tel que $x_n \neq 0$ et $y_n \neq 0$ (un tel n existe toujours puisque $x_n \neq 0 \Rightarrow x_m \neq 0,\ \forall m \geq n$). Comme $p^n \nmid x$ et $p^n \nmid y$, on a que $\forall m \geq n,\ p^n \nmid x_m$ et $p^n \nmid y_m$. En particulier, p^{2n} ne peut diviser $x_{2n}y_{2n}$, autrement dit $x_{2n}y_{2n} \neq 0$ ce qui contredit l'hypothèse $x.y = 0$.
- p est une uniformisante (et donc $\mathbf{Z}_p$ est un anneau de valuation discrète) :
  si $x \in \mathbf{Z}_p$ et $j = \max\{i : x = 0 \bmod p^i\}$, alors $x = up^j$ et $p \nmid u$, donc $u \in \mathbf{Z}_p^*$.

– Le morphisme canonique $\varphi : \mathbf{Z} \to \mathbf{Z}_p,\ x \to (x \bmod p, x \bmod p^2, \ldots)$ est injectif : $(x = 0 \bmod p^n\ \forall n) \Rightarrow (x = 0)$. En particulier $\mathrm{char}(\mathbf{Z}_p) = 0$.

□

La proposition suivante résume les propriétés de la valuation discrète sur $\mathbf{Z}_p$:

Proposition 2.14. *Soit $x \in \mathbf{Z}_p \setminus \{0\}$. Alors x s'écrit sous la forme up^n où $u \in \mathbf{Z}_p^*$ et $n \in \mathbf{N}$.*
L'entier n est appelé **valuation p-adique** *de x et noté $\nu_p(x)$. Par convention, on pose $\nu_p(0) = +\infty$.*
$\nu_p : \mathbf{Z}_p \to \mathbf{N},\ x \to \nu_p(x)$ vérifie :
– $\nu_p(xy) = \nu_p(x) + \nu_p(y)$
– $\nu_p(x + y) \geq \inf\{\nu_p(x), \nu_p(y)\}$, *avec égalité si $\nu_p(x) \neq \nu_p(y)$.*
ν_p est donc une valuation discrète sur $\mathbf{Z}_p$ appelée **valuation p-adique**.

$\mathbf{Z}_p$ étant un anneau intègre, on peut considérer son corps de fractions :

Définition 2.15. *On appelle corps des nombres p-adiques le corps des fractions de l'anneau $\mathbf{Z}_p$, noté $\mathbf{Q}_p$.*
La valuation p-adique se prolonge sur $\mathbf{Q}_p$ en posant $\nu_p(\frac{1}{x}) = -\nu_p(x),\ \forall x \in \mathbf{Z}_p$. Elle induit une norme sur $\mathbf{Q}_p$

$$|.|_p : \mathbf{Q}_p \to \mathbf{R}_+,\ x \to |x|_p = p^{-\nu_p(x)}$$

appelée **norme p-adique**.

Une autre façon d'introduire $\mathbf{Q}_p$ est de considérer sur $\mathbf{Z}$ la valuation suivante (encore appelée valuation p-adique) :

$$\forall x \in \mathbf{Z},\ \nu_p(x) = n \text{ où } x = qp^n \text{ avec } p \wedge q = 1$$

Cette valuation se prolonge naturellement à $\mathbf{Q}$ et induit la norme p-adique sur $\mathbf{Q}$.

De la même façon que l'on peut définir $\mathbf{R}$ comme le complété de $\mathbf{Q}$ pour la norme archimédienne habituelle, on peut également voir $\mathbf{Q}_p$ comme le complété de $\mathbf{Q}$ pour la norme p-adique [Lang] :

Proposition 2.16. $\mathbf{Q}$ *est dense dans $\mathbf{Q}_p$ pour la norme $|.|_p$ et $(\mathbf{Q}_p, |.|_p)$ est complet.*

On retrouve alors $\mathbf{Z}_p$ comme l'anneau de valuation de $\mathbf{Q}_p$:

$$\mathbf{Z}_p = \{x \in \mathbf{Q}_p : |x|_p \leq 1\}$$

Corps des q-adiques

La construction du corps $\mathbf{Q}_q$ des q-adiques s'obtient comme une extension non ramifiée du corps des p-adiques [Lang] :

Définition 2.17. *Soit K une extension de $\mathbf{Q}_p$ de degré d, alors $K = \mathbf{Q}_p[X]/(P)$ avec $P \in \mathbf{Z}_p[X]$ (quitte à multiplier par un élément de $\mathbf{Z}_p$, on peut supposer que le contenu de P vaut 1), $\deg(P) = d$ et P irréductible dans $\mathbf{Z}_p[X]$.*
On dit que K est une **extension non ramifiée** *de degré d de $\mathbf{Q}_p$, si*

$$\deg(P) = \deg(P_N),\ \forall N \in \mathbf{N}$$

où les polynômes $P_N \in (\mathbf{Z}/p^N\mathbf{Z})[X]$ sont les polynômes obtenus par réduction de P modulo p^N.

Proposition 2.18. *Il existe une unique (à isomorphisme près) extension non ramifiée de degré d de $\mathbf{Q}_p$, notée $\mathbf{Q}_q$ où $q = p^d$. Cette extension est galoisienne, de groupe de Galois cyclique.*

En tant qu'extension de $\mathbf{Q}_p$, ν_p et $|.|_p$ se prolongent de façon unique à $\mathbf{Q}_q$. On note $\mathbf{Z}_q$ l'anneau de valuation de $\mathbf{Q}_q$:

$$\mathbf{Z}_q = \{x \in \mathbf{Q}_q : |x|_p \leq 1\}$$

Alternativement, il est possible de voir $\mathbf{Z}_q$ comme une extension de $\mathbf{Z}_p$:

$$\mathbf{Z}_q \simeq \mathbf{Z}_p[X]/(P)$$

et $\mathbf{Z}_q$ hérite de certaines propriétés de $\mathbf{Z}_p$

Propriété 2.19. *(i) $\mathbf{Z}_q$ est un anneau local, d'idéal maximal $p\mathbf{Z}_q$.*
(ii) Comme $\mathbf{Z}_q/p\mathbf{Z}_q \simeq (\mathbf{F}_p[X])/(P_1)$ où $P_1 = P$ mod p, le corps résiduel de $\mathbf{Z}_q$ est $\mathbf{F}_q$.
(iii) $\mathbf{Z}_q$ est la limite projective de ses réductions modulo p^N :

$$\mathbf{Z}_q = \varprojlim \mathbf{Z}_q/p^N\mathbf{Z}_q = \varprojlim(\mathbf{Z}/p^N\mathbf{Z})[X]/(P_N)$$

Relevé canonique et substitution de Frobenius

La structure du corps $\mathbf{Q}_q$ est étroitement liée à celle de $\mathbf{F}_q$ par le théorème suivant :

Théorème 2.20. *On a un isomorphisme, donné par la réduction modulo p, entre les groupes de Galois :*

$$Gal(\mathbf{F}_q/\mathbf{F}_p) \simeq Gal(\mathbf{Q}_q/\mathbf{Q}_p)$$

En particulier, comme $Gal(\mathbf{F}_q/\mathbf{F}_p)$ est engendré par le petit Frobenius $\sigma : x \mapsto x^p$, on peut lui faire correspondre un générateur Σ du groupe cyclique $Gal(\mathbf{Q}_q/\mathbf{Q}_p)$ appelé **substitution de Frobenius**.

Grâce à cette correspondance, on va pouvoir relever le q-ième Frobenius en un endomorphisme qui sera défini sur le relevé canonique $\mathcal{E}$ de la courbe ordinaire E.

On rappelle la définition du relevé canonique :

Définition 2.21. *Un relevé canonique de la courbe elliptique ordinaire E définie sur $\mathbf{F}_q$ est une courbe elliptique $\mathcal{E}$ définie sur $\mathbf{Q}_q$ satisfaisant :*
– la réduction de $\mathcal{E}$ modulo p est E,

– *l'homomorphisme d'anneaux* $End(\mathcal{E}) \to End(E)$ *induit par la réduction modulo* p *est un isomorphisme.*

En particulier, le q*-ième Frobenius* $\Phi_q : E \to E$ *se relève en un endomorphisme* $\mathcal{F}_q : \mathcal{E} \to \mathcal{E}$*, et on a* $\operatorname{Tr} \mathcal{F}_q = \operatorname{Tr} \Phi_q$*.*

On a le résultat suivant, dû à Deuring [Deu] :

Théorème 2.22. *Le relevé canonique* $\mathcal{E}$ *de* E *existe et est unique à isomorphisme près.*

Le petit Frobenius σ de $\mathbf{F}_q$ induit le p-ième morphisme Φ_p qui est une isogénie de E sur sa courbe conjuguée E^σ obtenue en appliquant σ aux coefficients de E. Cette isogénie peut être réitérée, ce qui donne une isogénie de E^σ dans E^{σ^2}, etc. E étant définie sur $\mathbf{F}_q \simeq \mathbf{F}_{p^d}$, en réitérant d fois le p-ième Frobenius, on trouve un cycle d'isogénies qui revient sur la courbe initiale E :

$$E \xrightarrow{\Phi_{p,0}} E^\sigma \xrightarrow{\Phi_{p,1}} E^{\sigma^2} \xrightarrow{\Phi_{p,2}} \dots \xrightarrow{\Phi_{p,d-1}} E^{\sigma^d} = E$$

Autrement dit en appliquant d fois le p-ième Frobenius à la courbe, on retrouve l'action du q-ième Frobenius.

De la même façon, le résultat suivant permet de voir que la substitution de Frobenius induit une isogénie $\mathcal{F}_p : \mathcal{E} \to \mathcal{E}^\Sigma$:

Théorème 2.23. *Soit* E *une courbe ordinaire définie sur* $\mathbf{F}_q$ *et* $\mathcal{E}$ *le relevé canonique de* E *défini sur* $\mathbf{Q}_q$*.*

Alors le p*-ième Frobenius* $\Phi_p : E \to E^\sigma$ *se relève en une isogénie* $\mathcal{F}_p : \mathcal{E} \to \mathcal{E}^\Sigma$*, où* Σ *est la substitution de Frobenius.*

En appliquant successivement ce résultat, on peut relever le cycle d'isogénies

$$E \xrightarrow{\Phi_{p,0}} E^\sigma \xrightarrow{\Phi_{p,1}} E^{\sigma^2} \xrightarrow{\Phi_{p,2}} \dots \xrightarrow{\Phi_{p,d-1}} E^{\sigma^d} = E$$

en un cycle d'isogénies

$$\mathcal{E} \xrightarrow{\mathcal{F}_{p,0}} \mathcal{E}^\Sigma \xrightarrow{\mathcal{F}_{p,1}} \mathcal{E}^{\Sigma^2} \xrightarrow{\mathcal{F}_{p,2}} \dots \xrightarrow{\mathcal{F}_{p,d-1}} \mathcal{E}^{\Sigma^d} = \mathcal{E}$$

et $\operatorname{Tr}(\mathcal{F}_q) = \operatorname{Tr}(\mathcal{F}_{2,d-1} \circ \dots \circ \mathcal{F}_{2,0}) = \operatorname{Tr}(\Phi_{2,d-1} \circ \dots \circ \Phi_{2,0}) = \operatorname{Tr}(\Phi_q)$.

On peut alors appliquer le théorème 2.11 pour calculer la trace de $\mathcal{F}_q$ en étudiant son action sur les différentielles et en déduire la cardinalité de $E(\mathbf{F}_q)$.

L'algorithme original dû à Satoh soulève alors deux difficultés :
– le calcul du relevé canonique à une précision suffisante (dans un sens que l'on explicitera par la suite)
– le calcul de la trace à l'aide du cycle d'isogénies établi sur ce relevé.

L'algorithme dû à Mestre qui s'appuie sur la suite arithmético-géométrique, a pour principal intérêt de fournir une méthode permettant de traiter élégamment les deux problèmes à la fois.

3 Algorithmique sur $\mathbf{Z}_q$

Dans toute la suite de ce rapport, on se place dans le cas particulier où $p = 2$ et $q = 2^d$.

On donne dans cette section les algorithmes utilisés et leurs complexités.

3.1 Notion de précision, choix d'implémentation

Il n'est bien sûr pas envisageable de considérer les éléments de $\mathbf{Z}_2$ (ou de $\mathbf{Z}_q$) au sens mathématique comme des suites infinies. On se contentera donc en pratique de travailler avec une certaine précision N, c'est-à-dire qu'un élément $a \in \mathbf{Z}_2$ sera approximé par sa réduction a_N modulo 2^N, $a_N \in \mathbf{Z}/2^N\mathbf{Z}$. Un tel élément requiert donc $O(N)$ bits mémoire.

De même, travailler dans $\mathbf{Z}_q$ à la précision N revient à travailler dans $(\mathbf{Z}/2^N\mathbf{Z})[X]/(P_N)$, où P_N est la réduction modulo 2^N du polynôme irréductible $P \in \mathbf{Z}_2[X]$ de degré d choisi pour définir l'extension. Un élément de $\mathbf{Z}_q$ défini à la précision N sera donc représenté par un polynôme de $(\mathbf{Z}/2^N\mathbf{Z})[X]$ de degré inférieur ou égal à $d-1$. Un tel élément requiert donc $O(dN)$ bits mémoire.

Il peut être pertinent d'évaluer la complexité des opérations d'addition et de multiplication dans $\mathbf{Z}_q$:
- L'addition de deux éléments de $\mathbf{Z}_q$ à la précision N se fait coefficients par coefficients et nécessite une simple réduction modulo 2^N, soit une complexité en $O(dN)$.
- La multiplication nécessite par contre une multiplication de deux polynômes de degré $d-1$, une réduction modulo 2^N et une réduction modulo P_N, soit $O(d^2N^2)$ si l'on utilise la multiplication naïve de polynômes.

La représentation binaire des entiers en machine est bien adaptée au travail en caractéristique 2, par conséquent la réduction modulo 2^N est particulièrement aisée.

Par contre, pour accélérer les calculs de réduction modulo P_N, on a tout intérêt à choisir P_N avec le plus possible de coefficients nuls (polynôme creux). En pratique, pour trouver P on part d'un polynôme irréductible P_1 sur $\mathbf{F}_2[X]$ de degré d et creux, dont l'existence est assurée par le résultat de Seroussi suivant :

Théorème 3.1 ([Sero])**.** *Pour tout entier $d \leq 10000$, il existe un polynôme de $\mathbf{F}_2[X]$ irréductible de degré d ayant seulement 3 ou 5 coefficients non nuls (polynômes trinomiaux ou pentanomiaux).*

On choisit ensuite pour P le relevé de P_1 dans $\mathbf{Z}_2[X]$ obtenu en relevant 0 et 1 (dans $\mathbf{F}_2$) par 0 et 1 (dans $\mathbf{Z}_2$).

Pour l'implémentation en C++, on a choisi d'utiliser la bibliothèque NTL [Shoup] en conjonction avec la bibliothèque GMP [GMP]. Pour respecter les normes standards de sécurité en cryptographie sur des courbes elliptiques, on prend typiquement un groupe à 2^{160} éléments, soit une extension de $\mathbf{F}_2$ dont le degré d est de l'ordre de 160. D'après Hasse, on peut se contenter de travailler avec une précision N d'environ 80 bits ; la bibliothèque GMP (*GNU Multiprecision Package*) permet de faire de l'arithmétique sur des entiers de cette taille. Le calcul modulaire dans $\mathbf{Z}_2[X]$ sera rendu possible avec la bibliothèque

NTL (*Number Theory Library*) dédiée. Ces deux bibliothèques implémentent des algorithmes de multiplication extrêmement efficaces (Karatsuba, Schönhage-Strassen), qui vont permettre de diminuer sensiblement la complexité de calcul pour la multiplication dans $(\mathbf{Z}/2^N\mathbf{Z})[X]/(P_N)$. Pour le calcul des différentes complexités, la multiplication de deux q-adiques à la précision N aura donc un coût en $O((dN)^\mu)$ où la constante μ est donnée par l'algorithme utilisé (par exemple $\mu = \log 3$ pour Karatsuba).

Etant donné que dans NTL, il n'y a pas de classes préexistantes pour le calcul des q-adiques (en particulier pour le changement de précision), quelques choix d'implémentation ont été faits en pratique :
- Le polynôme choisi pour l'extension $\mathbf{Z}_q$ de $\mathbf{Z}_2$ étant constant, on travaille dans la classe ZZ_X des polynômes ayant pour coefficients de grands entiers.
- On travaille donc avec des polynômes de degré $d-1$ à coefficients dans l'intervalle $[\![0; 2^N - 1]\!]$.
- L'addition et la multiplication modulo P_N de polyômes sont celles de la classe ZZ_X, on ramène ensuite les coefficients du résultat dans l'intervalle $[\![0; 2^N - 1]\!]$ par troncature des bits dans la classe ZZ.

Les deux paragraphes qui suivent présentent des méthodes de calcul d'inverse et de racine carrée. L'idée derrière ces deux algorithmes est d'utiliser un analogue pour les $\mathbf{Z}_q$-adiques des itérations de type Newton pour trouver les racines d'un polynôme.

3.2 Calcul d'inverse dans $\mathbf{Z}_q$

Soit $a \in \mathbf{Z}_q$ un élément inversible (i.e. non divisible par 2). L'algorithme suivant permet de trouver l'inverse z de a à la précision N, autrement dit tel que $az = 1 \bmod 2^N$.

Entrée : $a \in \mathbf{Z}_q$ inversible, $N \in \mathbf{N}$ la précision

Sortie : z l'inverse de a à la précision N

1. **si** $N = 1$ **alors**
2. $\quad z \leftarrow \frac{1}{a} \bmod 2$
3. **sinon**
4. $\quad z \leftarrow \text{Inverse}\left(a, \left\lfloor \frac{N+1}{2} \right\rfloor\right)$
5. $\quad z \leftarrow z + z(1 - az) \bmod 2^N$
6. **fin si**
7. **retourner** z

Remarque : A la ligne 1, il apparaît un calcul d'inverse dans $\mathbf{F}_q$ pour lequel on connaît des algorithmes efficaces. Cependant, dans l'implémentation de l'algorithme AGM, on aura $a = 1 \bmod 2$, donc l'approximation de l'inverse à la précision 1 sera toujours égale à 1.

Par ailleurs, cet algorithme donne une démonstration constructive du fait qu'un élément est inversible dans $\mathbf{Z}_q$ si et seulement si il est inversible (non nul) modulo 2.

Démonstration de l'algorithme.

Soit $z \in \mathbf{Z}_q$ tel que $1 - az = 0 \mod 2^{N'}$. A chaque étape de l'algorithme, on prend

$$z' = z + z(1 - az) \in \mathbf{Z}_q$$

1. On vérifie que $1 - az' = 0 \bmod 2^N$ où $N = 2N'$:
   Par hypothèse, $\exists k \in \mathbf{Z}_q,\ 1 - az = k2^{N'}$, donc

$$\begin{aligned} 1 - az' &= 1 - az - az(1 - az) \\ &= (1 - az)^2 \text{ (la convergence est quadratique)} \\ &= k^2 2^{2N'} \\ &= k^2 2^N \end{aligned}$$

   En particulier $1 - az = 0 \bmod 2^N$.
2. Il est clair que $z' = z \bmod 2^{N'}$, puisque $z' - z = 2^{N'} kz$.
3. La suite $(z_n)_{n \geq 1}$ obtenue à chaque itération de l'algorithme est une suite de Cauchy dans $\mathbf{Z}_q$:

$$\begin{aligned} & z_{n+p} - z_n = 0 \bmod 2^{N_n} \text{ où } N_n \text{ est la précision obtenue pour } z_n \\ \Rightarrow\ & |z_{n+p} - z_n|_2 \leq 2^{-N_n} \underset{n \to \infty}{\to} 0 \end{aligned}$$

   Et comme $(\mathbf{Z}_q, |.|_2)$ est complet, la suite (z_n) converge vers $z_\infty \in \mathbf{Z}_q$.
   De plus, à N fixé, on a que $\forall N_n > N,\ 1 - az_\infty = 1 - az_n = 0 \bmod 2^{N_n}$, par conséquent $1 - az_\infty = 0$ dans $\mathbf{Z}_q$ et z_∞ est bien l'inverse de a dans $\mathbf{Z}_q$.

□

Complexité du calcul de l'inverse dans $\mathbf{Z}_q$:
Le coût des additions étant négligeable par rapport à celui des multiplications, lors de la k-ième étape de l'algorithme (i.e. à précision 2^k), on effectue $O((d2^k)^\mu)$ opérations. Donc après $\log_2 N$ passages, la complexité en temps est de $O((dN)^\mu)$.

3.3 Calcul de racine carrée dans $\mathbf{Z}_q$

Soit a un carré dans $\mathbf{Z}_q$, alors a s'écrit sous la forme $a = u2^v$ où $v = \nu_2(a)$ est la valuation 2-adique (nécessairement paire) et $u \in \mathbf{Z}_q$ est inversible. Le calcul de $\sqrt{a} = 2^{\frac{v}{2}}\sqrt{u}$ se ramène donc au calcul de la racine carrée d'un élément inversible (i.e. non divisible par 2) de $\mathbf{Z}_q$.

L'algorithme suivant permet de trouver l'inverse de la racine carrée de a à la précision N, autrement dit z tel que $az^2 = 1 \mod 2^{N+1}$. Pour trouver la racine carrée de a à la précision N, il suffira alors de prendre az.

Entrée : $a \in \mathbf{Z}_q$ carré inversible, z_0 une approximation initiale de l'inverse de la racine carrée de a à l'ordre 2 et $N \in \mathbf{N}$ la précision

Sortie : z l'inverse de la racine carrée de a à la précision N

1. **si** $N \leq 2$ **alors**
2. $\quad z \leftarrow z_0$

3. **sinon**
4. $N' \leftarrow \left\lfloor \frac{N+2}{2} \right\rfloor$
5. $z \leftarrow \text{RacineCarreeInverse}(a, z_0, N')$
6. $x \leftarrow 1 - az^2 \bmod 2^{N+1}$
7. $z \leftarrow z + \frac{zx}{2} \bmod 2^N$
8. **fin si**
9. **retourner** z

Remarque : Contrairement à l'algorithme précédent, il faut ici une approximation à la précision 2 du calcul de l'inverse de la racine carrée de a, qui peut ne pas exister (les éléments inversibles de $\mathbf{Z}_q$ ne sont pas nécessairement des carrés). Cependant, dans l'implémentation de l'algorithme AGM, on aura $a = 1 \bmod 8$ (cf lemme 4.4), donc l'approximation de l'inverse de la racine carrée à la précision 2 sera toujours égale à 1.

Par ailleurs, cet algorithme donne une démonstration constructive du fait qu'un élément inversible est un carré dans $\mathbf{Z}_q$ si et seulement si il admet une racine carrée approchée modulo 4.

Démonstration de l'algorithme.
Soit $z \in \mathbf{Z}_q$ tel que $1 - az^2 = 0 \bmod 2^{N'+1}$. A chaque étape de l'algorithme, on prend

$$z' = z + \frac{z(1-az^2)}{2} \in \mathbf{Z}_q$$

1. On vérifie que $1 - az' = 0 \bmod 2^{N+1}$ où $N = 2N' - 1$:
   Par hypothèse, $\exists k \in \mathbf{Z}_q,\ 1 - az^2 = k2^{N'+1}$, donc

$$\begin{aligned} 1 - az'^2 &= 1 - a\left(z + \frac{z(1-az^2)}{2}\right)^2 \\ &= 1 - az^2 - az^2(1-az^2) - \frac{az^2(1-az^2)^2}{4} \\ &= (1-az^2)^2 - \frac{az^2(1-az^2)^2}{4} \\ &= \frac{(1-az^2)^2}{4}(4-az^2) \\ &= 2^{2N'}k^2(4-az^2) \end{aligned}$$

   En particulier $1 - az^2 = 0 \bmod 2^{N+1}$. D'autre part, on constate qu'il faut que $N' \geq 2$ pour gagner en précision (i.e. avoir $N > N'$). Ceci justifie la nécessité de connaître la solution approchée à la précision 2.
2. Il est clair que $z' = z \bmod 2^{N'}$, puisque $z' - z = \frac{z(1-az^2)}{2} = 2^{N'}kz$.
3. Avec la même démonstration que pour l'algorithme précédent, on voit que la suite $(z_n)_{n\geq 1}$ obtenue à chaque itération de l'algorithme est une suite de Cauchy dans $\mathbf{Z}_q$, et que donc elle converge vers z_∞ vérifiant $1 - az^2 = 0$ dans $\mathbf{Z}_q$.

□

Complexité du calcul de la racine carrée dans $\mathbf{Z}_q$:
En suivant le même raisonnement que pour l'algorithme précédent, on trouve une complexité en $O((dN)^\mu)$ pour le calcul de l'inverse de la racine carrée. Il faut rajouter la dernière multiplication az donnant la racine carrée à la précision N qui nécessite également $O((dN)^\mu)$ opérations élémentaires.

Il existe une légère amélioration de cet algorithme [KaMa], qui permet d'éviter cette dernière multiplication. On modifie la dernière étape de l'algorithme de la façon suivante : on calcule $b = az \bmod 2^{N'}$ et le résultat final est $b + z\frac{(a-b^2)}{2} \bmod 2^N$.

4 Approche AGM en caractéristique 2

4.1 Restriction du problème général du calcul de la cardinalité

En caractéristique 2, une courbe elliptique ordinaire est une courbe dont le j-invariant est non nul. En toute généralité, l'équation d'une telle courbe est de la forme

$$E \ : \ y^2 + xy = x^3 + a_2x^2 + a_6 \quad \text{où } a_6 \in \mathbf{F}_q^*$$

et son j-invariant ne dépend que de a_6 :

$$j = \frac{1}{a_6}$$

L'étude de la cardinalité de courbes elliptiques ordinaires définies sur $\mathbf{F}_q$ où $q = 2^d$ peut en fait se restreindre aux courbes ayant une équation de la forme :

$$E' \ : \ \tilde{y}^2 + \tilde{x}\tilde{y} = \tilde{x}^3 + a_6 \quad \text{où } a_6 \in \mathbf{F}_q^*$$

Il est en effet facile de voir que les équations de E et E' sont équivalentes sur $\mathbf{F}_q$ si et seulement si il existe un changement de coordonnées projectives de la forme :

$$\begin{cases} x = \tilde{x} \\ y = \tilde{y} + s\tilde{x} \end{cases} \qquad \text{où } s \text{ vérifie } s^2 + s - a_2 = 0 \text{ dans } \mathbf{F}_q$$

On distingue alors deux cas :

1. Si $\exists s \in \mathbf{F}_q,\ s^2 + s - a_2 = 0$, alors E et E' sont isomorphes et ont le même nombre de points rationnels dans $\mathbf{F}_q$. On peut donc supposer $a_2 = 0$.
2. Si $\forall s \in \mathbf{F}_q,\ s^2+s-a_2 \neq 0$, alors on peut trouver une solution dans $\mathbf{F}_{q^2}$, en particulier E et E' sont isomorphes et ont le même nombre de points rationnels dans $\mathbf{F}_{q^2}$. Dans ce cas, E' est appelée *la tordue* de E.
   Ainsi, on a
   $$\mathrm{Tr}(\Phi_{q^2}) = \mathrm{Tr}(\Phi'_{q^2})$$
   où $\Phi_{q^2} \in End(E)$, resp. $\Phi'_{q^2} \in End(E')$ sont les q^2-ièmes Frobenius sur E et E' respectivement.
   On en déduit que :
   $$\mathrm{Tr}(\Phi_{q^2}) = \mathrm{Tr}(\Phi_q)^2 - 2\det(\Phi_q) = \mathrm{Tr}(\Phi_q)^2 - 2q = \mathrm{Tr}(\Phi'_q)^2 - 2q$$

En particulier,

$$\mathrm{Tr}(\Phi_q) = \pm\,\mathrm{Tr}(\Phi'_q)$$

Il est donc possible de déduire la cardinalité de E connaissant celle de E'.

Dans la suite, on supposera donc que E désigne une courbe elliptique sur $\mathbf{F}_q$ ($q = 2^d$) d'équation

$$E: \quad y^2 + xy = x^3 + c \qquad\qquad c \in \mathbf{F}_q^*$$

On rappelle les deux résultats principaux utilisés pour le calcul de la cardinalité de E :

Théorème 4.1 (Relevé canonique de E)**.**
Il existe un unique relevé $\mathcal{E}$, appelé canonique, de E dans $\mathbf{Q}_q$ vérifiant :

(i) $\mathcal{E}$ admet une bonne réduction modulo 2, c'est-à-dire $\mathcal{E}$ admet une équation $F(x,y) = 0$, $F \in \mathbf{Z}_q[X]$ telle que

$$F(x,y) = y^2 + xy - x^3 - c \mod 2$$

(ii) $End(\mathcal{E}) \simeq End(E)$

En particulier, $\Phi_2 : E \to E^\sigma$ se relève en une isogénie $\mathcal{F}_2 : \mathcal{E} \to \mathcal{E}^\Sigma$ (avec $\Sigma \in Gal(\mathbf{Q}_q/\mathbf{Q}_p)$ la substitution de Frobenius) et plus généralement, $\Phi_{2,k} : E^{\sigma^k} \to E^{\sigma^{k+1}}$ se relève en $\mathcal{F}_{2,k} : \mathcal{E}^{\Sigma^k} \to \mathcal{E}^{\Sigma^{k+1}}$.

De même, Φ_q se relève en un endomorphisme $\mathcal{F}_q \in End(\mathcal{E})$ tel que $\mathcal{F}_{2,d-1} \circ \ldots \circ \mathcal{F}_{2,0} = \mathcal{F}_q$ et $\mathrm{Tr}(\mathcal{F}_q) = \mathrm{Tr}(\Phi_q)$.

Le calcul de la trace va être rendu possible par le théorème suivant :

Théorème 4.2 (Satoh)**.**
Soit $c \in \mathbf{Q}_q$ tel que $\mathcal{F}_q^*\omega = c\,\omega$ où ω est une différentielle invariante sur $\mathcal{E}$. Alors

$$\mathrm{Tr}(\mathcal{F}_q) = c + \frac{q}{c}$$

On est donc ramené à l'étude de l'action de $\mathcal{F}_q^*$ sur les différentielles de $\mathcal{E}$.

On suppose dans la suite que le relevé canonique $\mathcal{E}$ de E est isomorphe à la courbe $\mathcal{E}_{a,b}$ d'équation :

$$\mathcal{E}_{a,b}: \quad y^2 = x(x-a^2)(x-b^2)$$

avec $a, b \in \mathbf{Z}_q$ tels que

$$\begin{cases} a, b \in 1 + 4\mathbf{Z}_q \\ \dfrac{a}{b} \in 1 + 8\mathbf{Z}_q \end{cases} \tag{1}$$

(on verra en 4.4 comment il est possible de se ramener à ce cas).

Soit $\omega = \frac{dx}{y} \in \Omega(\mathcal{E}_{a,b})$ la différentielle holomorphe définie sur $\mathcal{E}_{a,b}$, on cherche donc à calculer $\mathcal{F}_q^*(\omega) = \mathcal{F}_{2,0}^* \circ \ldots \circ \mathcal{F}_{2,d-1}^*(\omega)$.

A cet effet, on va introduire, pour tout k, un isomorphisme entre $\mathcal{E}^{\Sigma^k}$ et une courbe $\mathcal{E}_{a_k,b_k}$ d'équation :

$$\mathcal{E}_{a_k,b_k}: \quad y^2 = x(x-a_k^2)(x-b_k^2)$$

où la suite $(a_k, b_k)_{k\geq 0}$ est une suite arithmético-géométrique.

4.2 Suite arithmético-géométrique

Soient $\alpha, \beta \in \mathbf{Z}_q$ tels que

$$\begin{cases} \alpha, \beta \in 1 + 4\mathbf{Z}_q \\ \dfrac{\alpha}{\beta} \in 1 + 8\mathbf{Z}_q \end{cases} \tag{2}$$

On définit récursivement une suite arithmético-géométrique (α_k, β_k) (AGM) par

$$\begin{cases} (\alpha_0, \beta_0) = (\alpha, \beta) \\ (\alpha_{k+1}, \beta_{k+1}) = \left(\dfrac{\alpha_k + \beta_k}{2}, \sqrt{\alpha_k \beta_k} \right) \quad (\text{noté } AGM(\alpha_k, \beta_k)) \end{cases} \tag{3}$$

On va montrer que l'hypothèse (2) est héréditaire (cf lemme 4.4). L'ambiguïté sur le choix de la racine carrée dans (3) est levée grâce au résultat suivant :

Lemme 4.3. *Si $c \in 1 + 8\mathbf{Z}_q$, alors il existe un unique $e \in 1 + 4\mathbf{Z}_q$ tel que $e^2 = c$.*

Démonstration.

- L'existence est assurée par l'algorithme d'extraction de racine carrée dans $\mathbf{Z}_q$: il suffit de prendre 1 pour approximation à la précision 2, on obtient ensuite e avec la précision souhaitée.
- Unicité : dans $\mathbf{Q}_q$, $X^2 - c = 0$ admet au plus deux racines $\pm e \in \mathbf{Q}_q$. Soit $e \in 1 + 4\mathbf{Z}_q$ tel que $e^2 = c$, supposons que $-e \in 1 + 4\mathbf{Z}_q$: alors $\exists t, t' \in \mathbf{Z}_q$ tels que $e = 1 + 4t$ et $-e = 1 + 4t'$.
  Par conséquent $e - e = 2(1 + 2(t + t')) = 0 \Rightarrow 1 + 2(t + t') = 0$ dans $\mathbf{Q}_q$, en particulier 2 est inversible dans $\mathbf{Z}_q$, ce qui est absurde.

□

Lemme 4.4. *Soient $\alpha, \beta \in 1 + 4\mathbf{Z}_q$ tels que $\frac{\alpha}{\beta} \in 1 + 8\mathbf{Z}_q$. Alors*

(i) $\alpha' = \frac{\alpha+\beta}{2} \in 1 + 4\mathbf{Z}_q$

(ii) $\alpha\beta \in 1 + 8\mathbf{Z}_q$, *donc en particulier* $\beta' = \sqrt{\alpha\beta} \in 1 + 4\mathbf{Z}_q$ *(lemme 4.3)*

(iii) $\frac{\alpha'}{\beta'} \in 1 + 8\mathbf{Z}_q$

Démonstration. Par hypothèse, $\exists t, t', t'' \in \mathbf{Z}_q$ tels que $\alpha = 1+4t$, $\beta = 1+4t'$ et $\frac{\alpha}{\beta} = 1+8t''$.

On remarque tout d'abord que si $\frac{\alpha}{\beta} = \frac{1+4t}{1+4t'} = 1+8t''$, alors $1+4t = 1+4(t'+2t'')+32t't''$, en particulier $t + t' \in 2\mathbf{Z}_q$.

(i) $\alpha' = 1 + 2(t + t') \in 1 + 4\mathbf{Z}_q$

(ii) $\alpha\beta = 1 + 4(t + t') + 16tt' \in 1 + 8\mathbf{Z}_q$ et $\beta' \in 1 + 4\mathbf{Z}_q$ avec le lemme 4.3

(iii) Le dernier point peut se voir en utilisant un développement en série entière de $(1 + \epsilon)^{-\frac{1}{2}}$ dans l'espace complet $(\mathbf{Z}_q, |.|_p)$:

$$
\begin{aligned}
\frac{\alpha'}{\beta'} &= \frac{\frac{\alpha}{\beta}+1}{2\sqrt{\frac{\alpha}{\beta}}} \\
&= \frac{2+8t''}{2\sqrt{1+8t''}} \text{ où } 8t'' \in 2\mathbf{Z}_q, \text{ donc } |8t''|_2 < 1 \\
&= (1+4t'')\sum_{k\geq 0}\frac{(-1)^k(2k)!}{2^{2k}(k!)^2}(8t'')^k \\
&= \sum_{k\geq 0}\frac{(-1)^k(2k)!}{(k!)^2}2^k(t'')^k + \sum_{k\geq 0}\frac{(-1)^k(2k)!}{(k!)^2}2^{k+2}(t'')^{k+1} \\
&= \sum_{k\geq 0}\frac{(-1)^k(2k)!}{(k!)^2}2^k(t'')^k + \sum_{k\geq 1}\frac{(-1)^{k-1}(2(k-1))!}{(k-1)!^2}2^{k+1}(t'')^k \\
&= 1+\sum_{k\geq 1}\frac{(-1)^{k-1}(2(k-1))!}{(k-1)!^2}2^k\left(2-\frac{(2k-1)2k}{k^2}\right)t''^k \\
&= 1+\sum_{k\geq 2}\frac{(-1)^k(2(k-1))!(k-1)}{(k-1)!^2k}2^{k+1}(t'')^k \\
&= 1+\sum_{k\geq 2}\frac{(-1)^k(2k-2)!}{(k-2)!k!}2^{k+1}(t'')^k \\
&= 1+\sum_{k\geq 2}(-1)^k C^k_{2k-2}2^{k+1}(t'')^k \\
\Rightarrow \frac{1+4t''}{\sqrt{1+8t''}} &\in 1+8\mathbf{Z}_q
\end{aligned}
$$

□

L'itération AGM donne des 2-isogénies entre les courbes, plus précisément on a le

Théorème 4.5. *Soient $\alpha, \beta \in 1+4\mathbf{Z}_q$ et $(\alpha', \beta') = AGM(\alpha, \beta)$.*

Alors $\mathcal{E}_{\alpha,\beta} : y^2 = x(x-\alpha^2)(x-\beta^2)$ et $\mathcal{E}_{\alpha',\beta'} : y^2 = x(x-\alpha'^2)(x-\beta'^2)$ sont 2-isogènes et l'isogénie est donnée par

$$
\begin{aligned}
\psi : \quad \mathcal{E}_{\alpha,\beta} &\to \mathcal{E}_{\alpha',\beta'} \\
(x,y) &\mapsto \left(\frac{(x+\alpha\beta)^2}{4x}, y\frac{(x-\alpha\beta)(x+\alpha\beta)}{8x^2}\right)
\end{aligned}
$$

L'action de ψ sur la différentielle holomorphe $\frac{dx'}{y'} \in \Omega^1(\mathcal{E}_{\alpha',\beta'})$ est donnée par

$$\psi^*\left(\frac{dx'}{y'}\right) = 2\frac{dx}{y}$$

où $\frac{dx}{y} \in \Omega^1(\mathcal{E}_{\alpha,\beta})$ est la différentielle holomorphe sur $\mathcal{E}_{\alpha,\beta}$.

Le noyau de ψ est composé de deux points :

$$\ker(\psi) = \{(0,0); O_{\mathcal{E}_{\alpha,\beta}}\}$$

4.3 Calcul de la trace du Frobenius

On reprend les notations de (1), et on note (a_k, b_k) la suite AGM initialisée à (a, b).

Le résultat suivant permet alors de faire le lien entre $\psi : \mathcal{E}_{a,b} \to \mathcal{E}_{a_1,b_1}$ et le 2-ième morphisme de Frobenius $\mathcal{F}_2 : \mathcal{E}_{a,b} \to \mathcal{E}^{\Sigma}_{a,b} = \mathcal{E}_{\Sigma(a),\Sigma(b)}$ défini, à isomorphisme près, sur le relevé canonique :

Proposition 4.6.

$$\ker(\mathcal{F}_2) = \ker(\psi)$$

En particulier, il existe un unique isomorphisme $\lambda : \mathcal{E}_{a_1,b_1} \to \mathcal{E}_{\Sigma(a),\Sigma(b)}$ tel que $\mathcal{F}_2 = \lambda \circ \psi$:

$$\begin{array}{ccc} \mathcal{E}_{a,b} & \xrightarrow{\psi} & \mathcal{E}_{a_1,b_1} \\ & \searrow^{\mathcal{F}_2} & \downarrow \lambda \\ & & \mathcal{E}^{\Sigma}_{a,b} \end{array}$$

Démonstration. En notant $\pi : \mathbf{Z}_q \to \mathbf{F}_q$ la réduction modulo 2, et $\mu : \mathcal{E}_{a,b} \to \mathcal{E}$ l'isomorphisme entre $\mathcal{E}_{a,b}$ et le relevé canonique, on a le diagramme commutatif suivant :

$$\begin{array}{ccc} \mathcal{E}_{a,b} & \longrightarrow & \mathcal{E}_{\Sigma(a),\Sigma(b)} \\ \downarrow \mu & & \downarrow \mu \\ \mathcal{E} & \xrightarrow{\mathcal{F}_2} & \mathcal{E}^{\Sigma} \\ \downarrow \pi & & \downarrow \pi \\ E & \xrightarrow{\Phi_2} & E^{\sigma} \end{array}$$

En identifiant $\mathcal{F}_2$ et $\mu^{-1} \circ \mathcal{F}_2 \circ \mu$, on a que $\mathcal{F}_2 : \mathcal{E}_{a,b} \to \mathcal{E}_{\Sigma(a),\Sigma(b)}$ est une isogénie qui relève $\Phi_2 : E \to E^{\sigma}$ à isomorphisme près, donc $\#\ker(\mathcal{F}_2) = \deg(\mathcal{F}_2) = \deg \Phi_2 = 2$.

En particulier, $\widehat{\mathcal{F}}_2 \circ \mathcal{F}_2 = [2]$ dont on déduit :

$$\ker(\mathcal{F}_2) \subset \mathcal{E}_{a,b}[2]$$

On détermine alors $\mathcal{E}_{a,b}[2]$ en constatant qu'en les points $(0,0)$, $(a^2,0)$ et $(b^2,0)$, $\mathcal{E}_{a,b}$ admet une tangente verticale. Ainsi $\{(0,0); (a^2,0); (b^2,0); O_{\mathcal{E}_{a,b}}\} \subset \mathcal{E}_{a,b}[2]$, et comme $\#\mathcal{E}_{a,b}[2] = 4$ (puisque $\mathrm{char}\,(\mathbf{Q}_q) = 0$) on a :

$$\mathcal{E}_{a,b}[2] = \{(0,0); (a^2,0); (b^2,0); O_{\mathcal{E}_{a,b}}\}$$

Par ailleurs, comme $\Phi_2 : E \to E^{\sigma}$ est purement inséparable, son noyau est réduit à O_E (le point à l'infini), en particulier Φ_2 est injective. Donc $\ker(\pi \circ \mathcal{F}_2) = \ker(\Phi_2 \circ \pi) = \ker \pi$ et

$$\ker(\mathcal{F}_2) \subset \ker \pi \cap \mathcal{E}_{a,b}[2]$$

Il reste à voir quels sont les points de $\mathcal{E}_{a,b}[2]$ envoyés sur O_E par π. Avec les équations de Weierstrass des courbes $\mathcal{E}$ et $\mathcal{E}_{a,b}$, on peut expliciter l'isomorphisme μ :

$$\mu(x,y) = \left(\frac{x - ab}{4}, \frac{y - x + ab}{8}\right)$$

Alors :

– $\mu(0,0) = [-2ab : ab : 8] \in \mathbf{P}^2(\mathbf{Z}_q)$ (puisque $ab \in 1 + 8\mathbf{Z}_q$) et $\pi(\mu(0,0)) = O_E$
– $\mu(a^2, 0) = \left[\frac{a(a-b)}{4} : \frac{a(b-a)}{8} : 1\right] \in \mathbf{P}^2(\mathbf{Z}_q)$ (puisque $b-a \in 8\mathbf{Z}_q$) et $\pi(\mu(a^2,0)) = \left(0, \frac{a-b}{8}\right) \neq O_E$
– $\pi(\mu(b^2, 0)) \neq O_E$ par un calcul similaire
– $\pi(\mu(O_{\mathcal{E}_{a,b}})) = \pi(O_{\mathcal{E}}) = O_E$
Donc

$$\ker(\mathcal{F}_2) = \{(0,0); O_{\mathcal{E}_{a,b}}\} = \ker(\psi)$$

□

Plus généralement, on aimerait associer à chaque $\mathcal{F}_{2,k} : \mathcal{E}_{\Sigma^k(a),\Sigma^k(b)} \to \mathcal{E}_{\Sigma^{k+1}(a),\Sigma^{k+1}(b)}$ et à l'isogénie $\psi_k : \mathcal{E}_{a_k,b_k} \to \mathcal{E}_{a_{k+1},b_{k+1}}$ vérifiant les propriétés énoncées dans le théorème 4.5, un isomorphisme $\lambda_k : \mathcal{E}_{a_{k+1},b_{k+1}} \to \mathcal{E}_{\Sigma^{k+1}(a),\Sigma^{k+1}(b)}$ qui permette de passer de l'une à l'autre.

Par récurrence, en répétant l'argument de la proposition 4.6, on démontre :

Théorème 4.7. *Soient (a_k, b_k) la suite AGM initialisée par (a, b), alors on a le diagramme commutatif suivant :*

$$\begin{array}{ccccccccc}
\mathcal{E}_{a,b} & \xrightarrow{\psi_0} & \mathcal{E}_{a_1,b_1} & \xrightarrow{\psi_1} & \mathcal{E}_{a_2,b_2} & \xrightarrow{\psi_2} & \cdots & \xrightarrow{\psi_{d-1}} & \mathcal{E}_{a_d,b_d} \\
\downarrow{\scriptstyle Id} & & \downarrow{\scriptstyle \lambda_1} & & \downarrow{\scriptstyle \lambda_2} & & \downarrow & & \downarrow{\scriptstyle \lambda_d} \\
\mathcal{E}_0 & \xrightarrow{\mathcal{F}_{2,0}} & \mathcal{E}_1 & \xrightarrow{\mathcal{F}_{2,1}} & \mathcal{E}_2 & \xrightarrow{\mathcal{F}_{2,2}} & \cdots & \xrightarrow{\mathcal{F}_{2,d-1}} & \mathcal{E}_d \simeq \mathcal{E}_0
\end{array}$$

où $\mathcal{E}_k = \mathcal{E}_{\Sigma^k(a),\Sigma^k(b)}$, $\mathcal{F}_{2,k} : \mathcal{E}_k \to \mathcal{E}_{k+1}$ est le relevé du 2-ième homomorphisme de Frobenius Φ_2 et les $\lambda_k : \mathcal{E}_{a_k,b_k} \to \mathcal{E}_k$ sont des isomorphismes.

Ainsi, on a trouvé un isomorphisme $\lambda_d : \mathcal{E}_{a_d,b_d} \to \mathcal{E}_{a_0,b_0}$ tel que $\mathcal{F}_q = \lambda_d \circ \psi_{d-1} \circ \ldots \circ \psi_1 \circ \psi_0$.

Les deux courbes $\mathcal{E}_{a_k,b_k}$ et $\mathcal{E}_{\Sigma^k(a),\Sigma^k(b)}$ admettant respectivement une équation de Weierstrass de la forme $y'^2 = x'(x'-a_k^2)(x'-b_k^2)$ et $y^2 = x(x-\Sigma^k(a)^2)(x-\Sigma^k(b)^2)$, l'isomorphisme λ_d est nécessairement de la forme :

$$x' = u^2 x + r \qquad y' = u^3 y \qquad u, r \in \mathbf{Q}_q$$

En utilisant les propriétés de la suite AGM, on peut montrer par récurrence sur k que $u = \pm\frac{\Sigma^k(a)}{a_k}$ et $r = 0$ ([CoFr] p.438).

En particulier,

$$\lambda_d(x', y') = \left(\left(\frac{a_0}{a_d}\right)^2 x', \pm\left(\frac{a_0}{a_d}\right)^3 y'\right)$$

et

$$\lambda_d^*\left(\frac{dx}{y}\right) = \frac{\lambda_d^*(dx)}{\lambda_d^*(y)} = \pm\frac{a_d}{a_0}\frac{dx'}{y'}$$

On en déduit un premier résultat sur le calcul de la trace du Frobenius :

$$\mathcal{F}_q^*\left(\frac{dx}{y}\right) = \psi_0^* \circ \ldots \circ \psi_{d-1}^* \circ \lambda_d^*\left(\frac{dx}{y}\right) = \pm 2^d \frac{a_d}{a_0}\frac{dx}{y}$$

donc avec le théorème de Satoh,

$$\mathrm{Tr}(\mathcal{F}_q) = \pm\frac{a_0}{a_d} \pm 2^d \frac{a_d}{a_0}$$

Il est possible de déterminer le signe de la trace en constatant :
– d'une part que $\frac{a_0}{a_d} \in 1 + 4\mathbf{Z}_q$ (car $a_0, a_d \in 1 + 4\mathbf{Z}_q$)
– d'autre part que $\mathrm{Tr}(\Phi_q) = 1 \bmod 4$:
en effet, on vérifie que le point $P = (\sqrt[4]{c}, \sqrt{c}) \in E$ est d'ordre 4 (le point $[2]P = (0, c)$ étant d'ordre 2), donc $\#(E(\mathbf{F}_q)) = 0 \bmod 4$, et comme $\#(E(\mathbf{F}_q)) = 1 + 2^d - \mathrm{Tr}(\Phi_q)$, on a bien $\mathrm{Tr}(\Phi_q) = 1 \bmod 4$.

On en déduit alors une approximation suffisante de la trace pour le calcul de la cardinalité de $E(\mathbf{F}_q)$:

Théorème 4.8.

$$\mathrm{Tr}(\mathcal{F}_q) = \mathrm{Tr}(\Phi_q) = \frac{a_0}{a_d} \bmod 2^{\lceil \frac{d}{2} \rceil + 2}$$

Avec ce théorème, on voit qu'il suffit de trouver a'_0 et a'_d approximant les valeurs a_0 et a_d à la précision $N - 1 = \lceil \frac{d}{2} \rceil + 2$. Le lemme suivant va permettre de déterminer la précision à laquelle on doit calculer la suite AGM (a_k, b_k).

Lemme 4.9. *Soient $\alpha, \beta \in \mathbf{Z}_q$ vérifiant les hypothèses (2). Soient $\alpha', \beta' \in \mathbf{Z}_q$ tels que*

$$\begin{cases} \alpha' = \alpha \bmod 2^{n-1} \\ \beta' = \beta \bmod 2^{n-1} \\ \frac{\alpha'}{\beta'} = \frac{\alpha}{\beta} \bmod 2^n \end{cases}$$

où $n \geq 3$ de telle sorte que α', β' vérifient les hypothèses (2). On note $(\alpha_1, \beta_1) = AGM(\alpha, \beta)$ et $(\alpha'_1, \beta'_1) = AGM(\alpha', \beta')$, alors

$$\begin{cases} \alpha'_1 = \alpha_1 \bmod 2^{n-1} \\ \beta'_1 = \beta_1 \bmod 2^{n-1} \\ \frac{\alpha'_1}{\beta'_1} = \frac{\alpha_1}{\beta_1} \bmod 2^{n+1} \end{cases}$$

Démonstration. Avec les hypothèses, on peut trouver $\zeta, \zeta' \in \mathbf{Z}_q$ tels que $\frac{\alpha}{\beta} = 1 + 8\zeta$, $\frac{\alpha'}{\beta'} = 1 + 8\zeta'$ et $\zeta = \zeta' \bmod 2^{n-3}$

Comme dans la preuve du lemme 4.4, on a $\begin{cases} \frac{\alpha_1}{\beta_1} = 1 + 8\zeta^2 + 16(\text{termes d'ordre supérieur}) \\ \frac{\alpha'_1}{\beta'_1} = 1 + 8\zeta'^2 + 16(\text{termes d'ordre supérieur}) \end{cases}$

et on voit que

$$\frac{\alpha'_1}{\beta'_1} = \frac{\alpha_1}{\beta_1} \bmod 2^{n+1}$$

On écrit ensuite $\beta'_1 = \beta' \sqrt{\frac{\alpha'}{\beta'}} = \beta \sqrt{\frac{\alpha}{\beta}} = \beta_1 \bmod 2^{n-1}$ et donc $\alpha'_1 = \beta'_1 \frac{\alpha'_1}{\beta'_1} = \beta_1 \frac{\alpha_1}{\beta_1} = \alpha_1 \bmod 2^{n-1}$. □

En pratique pour calculer la trace de $\mathcal{F}_q$, on va prendre $a'_0, b'_0 \in \mathbf{Z}_q$ tels que $(a'_0, b'_0) = (a, b) \bmod 2^N$ où $N = \lceil \frac{d}{2} \rceil + 3$. On a donc

$$\begin{cases} a'_0 = a \bmod 2^{N-1}, \quad b'_0 = b \bmod 2^{N-1} \\ \frac{a'_0}{b'_0} = \frac{a}{b} \bmod 2^N \end{cases}$$

On calcule par récurrence la suite (a'_k, b'_k) telle que $(a'_{k+1}, b'_{k+1}) = AGM(a'_k, b'_k) \bmod 2^N$. Le lemme assure alors que la suite approchée est correcte à la précision $N-1$: $(a'_k, b'_k) = (a_k, b_k) \bmod 2^{N-1}$. Donc $\frac{a'_0}{a'_d} = \frac{a_0}{a_d} \bmod 2^{N-1}$, i.e. à la précision souhaitée pour le calcul de la trace de $\mathcal{F}_q$.

4.4 Approximation du relevé canonique

Il reste à voir comment trouver les approximations (a'_0, b'_0) à la précision N de (a, b), où $\mathcal{E}_{a,b}$ est isomorphe au relevé canonique $\mathcal{E}$ de $E : y^2 + xy = x^3 + c,\ c \in \mathbf{F}_q^*$.

On utilise la proposition suivante qui montre que cela revient à approcher le j-invariant de $\mathcal{E}$:

Proposition 4.10. *Soit $n \in \mathbf{N}^*$. Soient (α, β) et (α', β') vérifiant les hypothèses (2).*

Alors on a équivalence entre

(i) $j(\mathcal{E}_{\alpha',\beta'}) = j(\mathcal{E}_{\alpha,\beta}) \bmod 2^n$

(ii) $\frac{\alpha'}{\beta'} = \left(\frac{\alpha}{\beta}\right)^{\pm 1} \bmod 2^{n+3}$

Démonstration. Il faut exprimer le j-invariant en fonction de α et β (en fait il ne dépend que du quotient $\frac{\alpha}{\beta}$). □

Pour initialiser le calcul de la trace, il suffit donc de trouver (a'_0, b'_0) vérifiant les hypothèses (2), et tels que $j(\mathcal{E}_{a'_0,b'_0}) = j(\mathcal{E}) = j(\mathcal{E}_{a,b}) \bmod 2^{N-3}$.

Ce qui est particulièrement remarquable dans l'approche AGM pour le comptage de points, est qu'en plus de permettre le calcul de la trace du Frobenius, elle donne une méthode efficace pour approcher le relevé canonique :

Théorème 4.11 (Approximation du relevé canonique)**.**
Soient $\alpha_0 = 1$ et $\beta_0 = 1 + 8\bar{c}$ où $\bar{c} \in \mathbf{Z}_q$ est un relevé de $c \in \mathbf{F}_q^*$. On note (α_k, β_k) la suite AGM initialisée à (α_0, β_0) et $\mathcal{E}_{\alpha_k,\beta_k}$ la courbe elliptique correspondante.

Les courbes $\mathcal{E}_{\alpha_k,\beta_k}$ approximent le relevé canonique $\mathcal{E}$ de la courbe $E : y^2 + xy = x^3 + c$ au sens suivant :

$$j(\mathcal{E}_{\alpha_k,\beta_k}) = \Sigma^{k+1}(j(\mathcal{E})) \bmod 2^{k+1}$$

Démonstration. Par récurrence sur k :

– $k = 0$:
on considère l'isomorphisme $\mu : (x, y) \mapsto \left(\frac{x-\alpha_0\beta_0}{4}, \frac{y-x+\alpha_0\beta_0}{8}\right)$ de $\mathcal{E}_{\alpha_0,\beta_0}$ sur son image $\mathcal{E}'$. On vérifie alors facilement que l'équation de $\mathcal{E}'$ se réduit modulo 2 à l'équation $E^\sigma : y^2 + xy = x^3 + c^2$. En particulier, $j(\mathcal{E}') = j(E^\sigma) = j(\mathcal{E}^\Sigma) \bmod 2$, i.e.

$$j(\mathcal{E}_{\alpha_0,\beta_0}) = \Sigma(j(\mathcal{E})) \bmod 2$$

– $k \to k+1$:
Avec les mêmes notations que précédemment, on a des 2-isogénies $\psi : \mathcal{E}_{\alpha_k,\beta_k} \to \mathcal{E}_{\alpha_{k+1},\beta_{k+1}}$ et $\mathcal{F}_2 : \mathcal{E}^{\Sigma^{k+1}} \to \mathcal{E}^{\Sigma^{k+2}}$. En notant $\mathbf{\Phi}_2(X, Y)$ le 2-ième polynôme modulaire, on a

$$\begin{cases} \mathbf{\Phi}_2(j(\mathcal{E}_{\alpha_k,\beta_k}), j(\mathcal{E}_{\alpha_{k+1},\beta_{k+1}})) = 0 \\ \mathbf{\Phi}_2(j(\mathcal{E}^{\Sigma^{k+1}}), j(\mathcal{E}^{\Sigma^{k+2}})) = 0 \end{cases}$$

Par hypothèse de récurrence, $j(\mathcal{E}_{\alpha_k,\beta_k}) = j(\mathcal{E}^{\Sigma^{k+1}}) \bmod 2^{k+1}$ donc, en utilisant un résultat dû à [VePr] concernant les solutions d'équations polynomiales sur $\mathbf{Z}_q$, $j(\mathcal{E}_{\alpha_{k+1},\beta_{k+1}}) = j(\mathcal{E}^{\Sigma^{k+2}}) = \Sigma^{k+2}(j(\mathcal{E})) \bmod 2^{k+2}$.

□

Ainsi on peut prendre $(\alpha_{N-4}, \beta_{N-4})$ comme valeurs pour (a'_0, b'_0). Cependant, on peut encore limiter la précision du calcul de la première suite AGM (α_k, β_k), en prenant la suite (α'_k, β'_k) telle que :

$$\begin{cases} (\alpha'_0, \beta'_0) = (\alpha_0, \beta_0) \bmod 2^4 \\ (\alpha'_k, \beta'_k) = AGM(\alpha'_{k-1}, \beta'_{k-1}) \bmod 2^{k+4} \end{cases}$$

En effet, il suffit de voir que pour tout $k \geq 0$

$$j(\mathcal{E}_{\alpha'_k,\beta'_k}) = j(\mathcal{E}_{\alpha_k,\beta_k}) = j(\mathcal{E}^{\Sigma^{k+1}}) \bmod 2^{k+1}$$

ce que l'on peut vérifier avec la proposition 4.10, en montrant par récurrence sur k que

$$\frac{\alpha'_k}{\beta'_k} = \frac{\alpha_k}{\beta_k} \bmod 2^{k+4} :$$

– $k = 0$: par hypothèse, $\frac{\alpha'_0}{\beta'_0} = \frac{\alpha_0}{\beta_0} \bmod 2^4$

– $k - 1 \to k$: on suppose $\frac{\alpha'_{k-1}}{\beta'_{k-1}} = \frac{\alpha_{k-1}}{\beta_{k-1}} \bmod 2^{k+3}$, alors $\frac{\alpha'_k}{\beta'_k} = \frac{1+\frac{\alpha'_{k-1}}{\beta'_{k-1}}}{2\sqrt{\frac{\alpha'_{k-1}}{\beta'_{k-1}}}} = \frac{\alpha_k}{\beta_k} \bmod 2^{k+4}$ avec le même argument que dans le lemme 4.9.

Remarques :

1. Si on prend $(a'_0, b'_0) = (\alpha_{N-4}, \beta_{N-4})$, alors $\mathcal{E}_{a'_0,b'_0}$ est en fait une approximation de $\mathcal{E}^{\Sigma^{N-3}}$ et donc ce qu'on calcule est la trace de $\mathcal{F}_q \in End(\mathcal{E}^{\Sigma^{N-3}})$. On obtient donc le nombre de points rationnels de la courbe $E^{\sigma^{N-3}}$. Ceci n'est pas gênant dans la mesure où le morphisme de Frobenius $\Phi_2^{N-3} : E \to E^{\sigma^{N-3}}$ est bijectif et donc préserve le nombre de points rationnels.

2. Le théorème 4.11 montre que, bien que la suite AGM ne converge pas (contrairement au cas réel), celle-ci fournit une approximation du relevé canonique. On peut en extraire une sous-suite convergente $(\alpha_{\varphi(k)}, \beta_{\varphi(k)})$ telle que

$$\lim_{k\to\infty} j(\mathcal{E}_{\alpha_{\varphi(k)},\beta_{\varphi(k)}}) = j(\mathcal{E}) = j(\mathcal{E}_{\alpha_\infty,\beta_\infty})$$

ce qui justifie l'hypothèse sur la forme du relevé canonique faite en (1).

4.5 Algorithme AGM et complexité, vérification et analyse des résultats

Algorithme AGM

Entrée : Une courbe elliptique ordinaire $E : y^2 + xy = x^3 + c$ avec $c \in \mathbf{F}^*_{2^d}$

Sortie : Le nombre de points rationnels de $E(\mathbf{F}_{2^d})$

Variables : un entier N pour la précision, deux polynômes a et b pour la suite arithmético-géométrique, un grand entier t pour la trace

$N \leftarrow \lceil \frac{d}{2} \rceil + 3$
$a \leftarrow 1 \bmod 2^4$
$b \leftarrow 1 + 8c \bmod 2^4$
pour $i = 5$ à N **faire**
    $(a, b) \leftarrow \left(\frac{a+b}{2}, \sqrt{ab}\right) \bmod 2^i$
fin pour
$a_0 \leftarrow a$
pour $i = 0$ à $d - 1$ **faire**
    $(a, b) \leftarrow \left(\frac{a+b}{2}, \sqrt{ab}\right) \bmod 2^N$
fin pour
$t \leftarrow \frac{a_0}{a} \bmod 2^{N-1}$
si $t^2 > 2^{d+2}$ **alors**
    $t \leftarrow t - 2^{N-1}$
fin si
retourner $2^d + 1 - t$

On renvoie à la section 3 pour le choix d'implémentation des entiers q-adiques et la description des opérations élémentaires sur ces entiers.

Le listing détaillé du programme est donné en annexe (A).

Voici un exemple d'exécution :

```
nina:~/Documents/MAA/JOUX/PROJET vanessavitse$ ./comptage
Entrer d tel que q=2^d : 100
P = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
Generer automatiquement une courbe ? (0/1) 1
c = [0 1 0 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 1
 0 0 0 1 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0
```

```
 1 0 1 1 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 1 0 0 0 1]
--- calcul en cours...
Debut 1ere phase d'AGM
Debut 2eme phase d'AGM
t = [4396500919896153]
nb de points = 126765060022822950859541067972 0
Duree du calcul : 0 mn 3 s 10 ms
Verification de la cardinalite de E:y2+xy=x3+c
Le point initial choisi est : ([1 0 1 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 1 1 1 1 0
0 1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0
1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1 1 0 0 1],[1 1 0 0 1
1 1 1 1 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 0 0 1 1 0 0 1 0 1 0
0 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1 1 1
1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1],[1])
youpi !
Voulez-vous tester avec un autre point initial ? (0/1) 0
Voulez-vous calculer la cardinalite d'une autre courbe ? (0/1) 0
--------FIN--------
```

Vérification des résultats

Pour s'assurer de la validité du résultat, on vérifie que l'ordre d'un point P rationnel de la courbe dont l'abscisse a été choisie aléatoirement, est bien un diviseur du nombre de points rationnels trouvé, cf le listing pour plus de détails.

Complexités

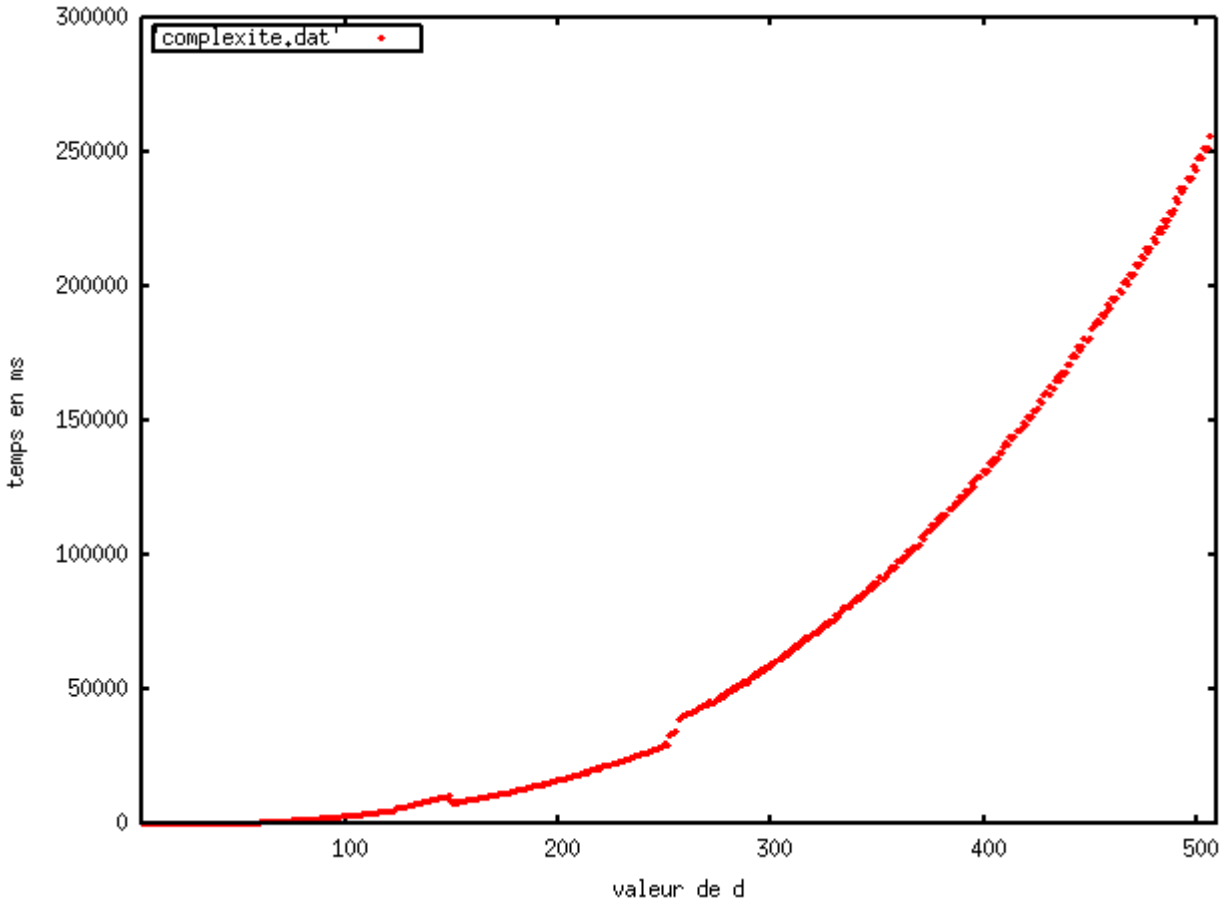
- Complexité en mémoire : on stocke $O(1)$ éléments de $\mathbf{Z}_q$ à la précision $N = O(d)$ (ayant donc d coefficients dans $\mathbf{Z}/2^N\mathbf{Z}$), soit une complexité en $O(d^2)$.
- Complexité en temps : on effectue de l'ordre de $2d$ calculs de racines carrés (à la précision N) et produits (à la précision $N+1$), de coût en $O((dN)^\mu) = O(d^{2\mu})$, soit une complexité totale en $O(d^{2\mu+1})$.

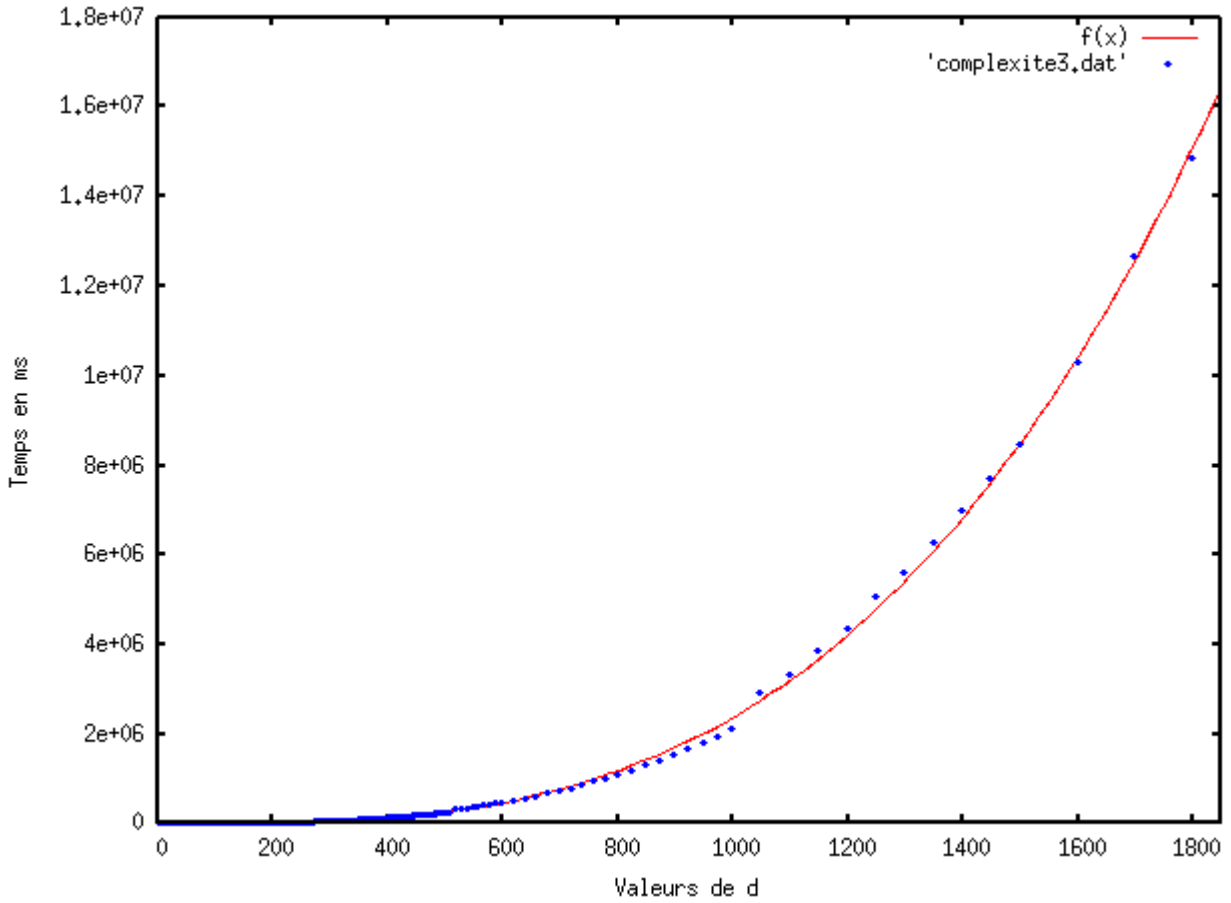
Résultats de l'implémentation

On a calculé le temps mis en moyenne pour calculer le nombre de points rationnels d'une courbe elliptique d'équation $E : y^2 + xy = x^3 + c$ définie sur $\mathbf{F}_{2^d}$, pour des valeurs de d variant de 5 à 1800, pendant environ 90h sur un ordinateur iMac avec un processeur Intel Core Duo à 2 GHz, et une mémoire de 1 Go à 667 MHz (DDR2 SDRAM).

Le nombre n de courbes testées pour le calcul de la moyenne a été choisi de façon à limiter le temps de calcul à environ 1mn pour chaque valeur de $d \leq 250$ (soit $n \simeq 3.10^7 d^{-3}$), et a été fixé à 2 pour $d > 250$. Pour les valeurs supérieures à 500, on n'a testé que quelques valeurs de d, de plus en en plus espacées.

Les figures de la page suivante illustrent les résultats obtenus.


'complexite.dat'
300000
250000
200000
150000
100000
50000
0
100
200
300
400
500
temps en ms
valeur de d


f(x)
'complexite3.dat'
1.8e+07
1.6e+07
1.4e+07
1.2e+07
1e+07
8e+06
6e+06
4e+06
2e+06
0
0
200
400
600
800
1000
1200
1400
1600
1800
Temps en ms
Valeurs de d

On constate des décrochements de la courbe aux valeurs $d = 150$, $d = 256 = 2^8$, $d \simeq 2^9$, $d \simeq 2^{10}$. Le premier est assez mystérieux, il est probablement dû à un changement d'algorithme de multiplication dans NTL. Les suivants soulignent une perte de performance qui s'explique certainement par un saut dans la façon de stocker les valeurs en mémoire.

Une régression non linéaire sous Gnuplot permet de retrouver la complexité théorique en $O(d^{2\mu+1})$ avec une complexité observée en $d^{3.148}$.

A Listing du programme

```
#include <NTL/ZZX.h>
#include <NTL/GF2XFactoring.h>
#include <NTL/GF2E.h>
#include <string>
#include <sstream>
#include <math.h>
#include <time.h>

NTL_CLIENT

typedef struct{
  GF2E x;
  GF2E y;
  GF2E t;
} point;//coord. d'un point de la courbe definie sur Fq

//polynome definissant l'extension Qq/Qp et donc Fq/Fp
ZZX glob_P;
ZZX un;

/*reduc reduit tous les coeff du polynome x modulo 2^precision*/
void reduc(ZZX& x, int precision){
  long degre = deg(x);
  long i;
  ZZ c;
  ZZ unit;
  unit=1;
  for (i=0 ; i<=degre ; i++){
    if ((c = coeff(x,i)) < 0){
      if ((c = trunc_ZZ(c,precision))!=0)
        c = (unit<<precision)-c;
    }
    else
      c = trunc_ZZ(c,precision);
    SetCoeff(x,i,c);
  }
}

/*reduction retourne un pol obtenu en reduisant les coeff
 *de x modulo 2^precision*/
ZZX reduction(const ZZX& x, int precision){
  long degre = deg(x);
  long i;
  ZZ c;
  ZZ unit;
```

```
  unit =1;
  ZZX res;
  res = 0;
  for (i=0 ; i<=degre ; i++){
    if ((c = coeff(x,i)) < 0){
      if ((c = trunc_ZZ(c,precision))!=0)
        c = (unit<<precision)-c;
    }
    else
      c = trunc_ZZ(c,precision);
    SetCoeff(res,i,c);
  }
  return res;
}

/*addition retourne la somme de a et b dans
 *(ZZ/2^precision ZZ)[X]/(glob_P)
 *les pol a et b sont supposes deja reduits
 *mod 2^precision et mod glob_P*/
ZZX addition(const ZZX& a, const ZZX& b, int precision){
  ZZX x = a+b;
  reduc(x,precision);
  return x;
}

/*retourne la difference de a et b dans
 *(ZZ/2^precision ZZ)[X]/(glob_P)
 *les pol a et b sont supposes deja reduits
 *modulo 2^precision et mod glob_P*/
ZZX soustraction(const ZZX& a, const ZZX& b, int precision){
  ZZX x = a-b;
  reduc(x,precision);
  return x;
}

/*multiplication retourne le produit de a et b
 *dans (ZZ/2^precision ZZ)[X]/(glob_P)
 *les pol a et b sont supposes deja reduits
 *modulo 2^precision et mod glob_P*/
ZZX mult(const ZZX& a, const ZZX& b, int precision){
  //MulMod effectue la multipliction dans ZZ[X]/(glob_P)
  ZZX  x = MulMod(a,b,glob_P);
  reduc(x,precision);
  return x;
}

/*inverse retourne l'inverse de a
```

```
 *dans (ZZ/2^precision ZZ)[X]/(glob_P)
 *l'algo requiert une approximation res0
 *de l'inverse a la precision 1*/
ZZX inverse(const ZZX& a, int precision,const ZZX& res0){
  ZZX res,z;
  if (precision == 1){
    res = reduction(res0,1);
    //verif que l'inverse donne pour la precision 1 est correct
    /*if (!IsOne(mult(a,res,1))){
      cout << "erreur d'inverse approche \n";
      exit(1);
      }*/
    return res;
  }
  z = inverse(a,(precision+1)/2,res0);
  res = addition(z,mult(z,1-mult(a,z,precision),precision),
                 precision);
  return res;
}

/*div2N retourne le polynome z/2^N
 *lorsque z est divisible par 2^N*/
ZZX div2N(ZZX& z, int N){
  ZZX res;
  long degre = deg(z);
  long i;
  ZZ c;
  /*if (!IsZero(reduction(z,N))){
    cout << "erreur de div2N \n";
    exit(1);
    }*/
  res = 0;
  for (i=0 ; i<=degre ; i++){
    c = (coeff(z,i) >> N);
    SetCoeff(res,i,c);
    }
  return(res);
}

/*invRC retourne l'inv de la racine carree a
 *dans (ZZ/2^precision)[X]/(glob_P)
 *l'algo requiert une approximation res0 de l'inverse
 *a la precision 2*/
ZZX invRC(const ZZX& a, int precision, const ZZX& res0){
  ZZX res,z,inter;
  if (precision <= 2){
    res = reduction(res0,precision);
```

```
    //verif que l'inv de la racine a la precision 2 est correct
    /*if (!IsOne(mult(a,mult(res,res,precision+1),
      precision+1))){
      cout << "erreur d'invRC approche \n";
      exit(1);
      }*/
    return res;
  }
  z = invRC(a,(precision+2)/2,res0);
  inter = 1-mult(a,mult(z,z,precision+1),precision+1);
  res = addition(z,mult(z,div2N(inter,1),precision),precision);
  return res;
}

/*sqrt retourne la racine carree de a
 *dans (ZZ/2^precision ZZ)[X]/(glob_P)*/
ZZX sqrt(const ZZX& a, int precision){
  ZZX res,z,b,inter;
  if (precision <= 2){
    res = 1;
    return res;
  }
  //version amelioree du calcul de racine carree
  z = invRC(a,(precision+2)/2,un);
  b = mult(a,z,(precision+2)/2);
  inter = soustraction(a,mult(b,b,precision+1),precision+1);
  res = addition(b,mult(z,div2N(inter,1),precision), precision);
  return res;
}

/*AGM retourne le nombre de points rationnels
 *dans Fq de la courbe y^2 + xy = x^3 + c'
 *ou c dans ZZ[X]/(glob_P) est un releve de c'*/
ZZ AGM(const ZZX& c){
  ZZX a,b,inter,a0;
  ZZ t;//la trace
  ZZ unit;
  unit=1;
  long i;
  long d = deg(glob_P);
  long N = (d+1)/2+3;//la precision
  //1ere phase : approximation du releve canonique
  cout << "Debut 1ere phase d'AGM\n";
  a = 1;
  inter = 8;
  b = addition(un,mult(inter,c,4),4);
  for (i=5 ; i<=N ; i++){
```

```
    inter = addition(a,b,i+1);
    inter = div2N(inter,1);
    b = sqrt(mult(a,b,i+1),i);
    a = inter;
  }
  //2eme phase : calcul de la trace
  cout << "Debut 2eme phase d'AGM\n";
  a0 = a;
  for (i=0 ; i<d ; i++){
    inter = addition(a,b,N+1);
    inter = div2N(inter,1);
    b = sqrt(mult(a,b,N+1),N);
    a = inter;
  }
  inter = mult(a0,inverse(a,N-1,un),N-1);
  t = ConstTerm(inter);
  cout << "t = " << inter << "\n";
  if((t*t)>(unit<<(d+2)))
    return ((unit<<d)+1-t+(unit<<(N-1)));
  return ((unit<<d)+1-t);
}


/*2eme partie du programme :
 *methode permettant de verifier que le nombre de points
 *trouve est vraisemblable*/

/*doubler remplace les coord. de q par celles de [2]q*/
void doubler(point* q){
  GF2E a,b,c,d,e;
  if (!IsZero(q->t)){//Q!=O
    a = sqr((q->x));
    b = a+(q->y)*(q->t);
    c = (q->x)*(q->t);
    d = sqr(c);
    e = sqr(b)+b*c;
    q->x = c*e;
    q->y = (b+c)*e+sqr(a)*c;
    q->t = c*d;
  }
}

/*sommer remplace les coord. de q par celles de p+q*/
void sommer(point* q, point* p)
{
  GF2E a,b,c,d,e;
  if (!IsZero(p->t)){ //P!=O
```

```
    if (IsZero(q->t)){//Q=O
      q->x = p->x;
      q->y = p->y;
      q->t = p->t;
    }
    else{//Q!=O
      if ((p->x)/(p->t)!=(q->x)/(q->t)){//P!=+-Q
        a = (p->y)*(q->t)+(p->t)*(q->y);
        b = (p->x)*(q->t)+(p->t)*(q->x);
        c = sqr(b);
        d = (p->t)*(q->t);
        e = (sqr(a)+a*b)*d+b*c;
        (q->x) = b*e;
        (q->y) = c*(a*(p->x)+(p->y)*b)*(q->t)+(a+b)*e;
        (q->t) = power(b,3)*d;
      }
      else{//P=+-Q
        if ((p->y)/(p->t)!=(q->y)/(q->t)){//P=-Q
          (q->x) = 0;
          (q->y) = 1;
          (q->t) = 0;
        }
        else//P=Q
          doubler(q);
      }
    }
  }
}

/*initialiser permet de trouver un point rationnel p
 *sur la courbe y^2 + xy = x^3 + c, a partir d'une
 *abcisse non nulle choisie au hasard*/
void initialiser(point* p, const GF2E& c){
  GF2E x,e,u,s1,s2;
  ZZ unit;
  long i,j,d;
  unit = 1;
  d = GF2E::degree();
  //tir au hasard de l'abscisse non nulle
  do{
    while (IsZero(x = random_GF2E()));
    e = x + c/sqr(x);
  }
  while(IsOne(trace(e)));
  //calcul de l'ordonnee cf p.228 handbook elliptic crypto
  if ((d%2)==0){
    do{
```

```
      u = random_GF2E();
    }
    while(IsZero(trace(u)));
    s1=0;
    for (i=0 ; i<d ; i++){
      s2=0;
      for (j=0 ; j<=i ; j++)
        s2+=power(e,(unit<<j));
      s1+=s2*power(u,(unit<<i));
    }
  }
  else{
    s1=0;
    for(i=0 ; i<=(d-3)/2 ; i++)
      s1+=power(e,(unit<<(2*i+1)));
  }
  (p->x) = x;
  (p->y) = x*s1;
  (p->t) = 1;
  //verification du calcul
  if (!IsZero(sqr(p->y)*(p->t)+(p->x)*(p->y)*(p->t)
              +power((p->x),3)+c*power((p->t),3)))
    cout << "le point INITIAL n'est pas sur la courbe\n";
  else
    cout << "Le point initial choisi est : "
         << "("<< p->x <<","<< p->y <<","<< p->t <<")\n";
}

/*verif renvoie :
 *0 si pour un point initial p aleatoire,
    le point [nb]p est envoye a l'infini
 *1 sinon*/
int verif(const GF2E& c, ZZ nb)
{
  point p,q;//point initial
  long d;
  long nBits;
  long i;
  d = GF2E::degree();
  initialiser(&p,c);
  q.x = 0;
  q.y = 1;
  q.t = 0;
  nBits = NumBits(nb);
  for (i=nBits-1 ; i>=0 ; i--){
    doubler(&q);
    if (bit(nb,i)==1)
```

```
      sommer(&q,&p);
    //verification du calcul
    if (!IsZero(sqr(q.y)*(q.t)+(q.x)*(q.y)*(q.t)
                +power((q.x),3)+c*power((q.t),3)))
      cout << "le point n'est pas sur la courbe\n";
  }
  if (!IsZero(q.t))//le point n'est pas le point a l'infini
    return 1;
  else return 0;
}

/*duree du calcul et conversion en min sec msec*/
void duree(clock_t begin, clock_t end)
{
  long mn=0, s=0, ms=0;
  long dureeCalc = 1000*(end-begin)/CLOCKS_PER_SEC;
  ms = dureeCalc % 1000;
  s = (dureeCalc/1000) % 60;
  mn = (dureeCalc/60000);
  cout<<"Duree du calcul : "<<mn<<" mn "
      <<s<<" s "<<ms<<" ms"<< endl;
}

int main()
{
  /*calcul du nombre de point de E : y^2 + xy = x^3 + c*/
  ZZX c;
  ZZ res;
  int precision;
  int d,rep,i;
  GF2X P;
  stringstream s;
  un = 1;

  /*initialisation de glob_P*/
  do{
  cout << "Entrer d tel que q=2^d : ";
  cin >> d;
  P = BuildSparseIrred_GF2X(d);
  s << P;
  s >> glob_P;
  cout << "P = "<< glob_P << "\n";

  /*choix de la courbe*/
  //initialisation du generateur pseudo-aleatoire
  srand(time(NULL));
  cout << "Generer automatiquement une courbe ? (0/1) ";
```

```
  cin >> rep;
  if (rep == 1){
    c = 0;
    do{
      for (i=0 ; i<d ; i++)
        SetCoeff(c,i,rand()%2);
    }while(IsZero(c));
    cout << "c = "<<c<<"\n";
  }
  else{
    cout << "Entrer le polynome c = ";
    cin >> c;
  }
  cout << "--- calcul en cours...\n";
  clock_t begin = clock();
  res = AGM(c);
  clock_t end = clock();
  cout << "nb de points = "<< res <<"\n";
  duree(begin,end);

  /*verification avec la methode verif du resultat obtenu*/
  cout << "Verification de la cardinalite de E:y2+xy=x3+c \n";
  GF2E::init(P);
  GF2E cp;
  s << c;
  s >> cp;
  do{
  if (verif(cp,res)==0)
    cout << "youpi !\n";
  else
    cout << "arghh...\n";
  cout << "Voulez-vous tester avec un autre "
       << "point initial ? (0/1) ";
  cin >> rep;
  }
  while (rep == 1);
  cout << "Voulez-vous calculer la cardinalite "
       << "d'une autre courbe ? (0/1) ";
  cin >> rep;
  }
  while (rep==1);
  cout << "---------FIN---------\n";
}
```

Références

[Deu] M. Deuring, *Die Typen der Multiplikatorenringe elliptischer Funktionenkörper*, Abh. Math. Sem. Hansischen Univ. 14 (1941), 197-272.

[GMP] T. Granlund et al., *The GNU multiple precision arithmetic library*, 2007. Version 4.2, `http ://gmplib.org/manual/`

[CoFr] H. Cohen, G. Frey et al., *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, CRC Press, Inc., 2006.

[Fou] M. Fouquet, *Anneau d'endomorphismes et cardinalité de courbes elliptiques : aspects algorithmiques*, thèse de doctorat de l'École polytechnique, décembre 2001.

[KaMa] A.H. Karp & P. Markstein, *High-precision division and square root*, ACM Trans. Math. Software 23 N°4 (1997), 561-589

[Kob] N. Koblitz, *Elliptic curve cryptosystems*, Mathematics of Computation 48, 1987, n°177, pages 156-157.

[Lang] S. Lang, *Algebra*, Addison-Wesley Publishing Compagny, 1984, second edition.

[Ler] R. Lercier, *Courbes Elliptiques et cryptographie*. Dans Direction des Centres d'Expertise et d'Essais, numéro 64 dans Revue Scientifique et Technique de la Defense, pages 59-66. Délégation générale pour l'armement, Juin 2004.

[Mes] J.-F. Mestre, *Lettre à Gaudry et Harley*. Disponible à `http ://www.math.jussieu.fr/~mestre`, 2001.

[Mil] V.-S.Miller, *Use of elliptic curves in cryptography*, Advances in Cryptology - CRYPTO'85, Lecture Notes in Comput. Sci., pages 417-426.

[Satoh] T. Satoh, *The canonical lift of an ordinary elliptic curve over a finite field and its point counting*, J. Ramanujan Math. Soc., 2000, volume 152, pages 47270.

[Sero] G. Seroussi, *Table of low-weight binary irreductible polynomials*, Tech. Report HPL-98-135, Hewlett-Packard, Août 1998.

[Shoof] R. Shoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Mathematics of Computations, 1985, volume 44, pages 483-494.

[Shoup] V. Shoup, NTL 5.3 : A library for doing number theory, 2002. `www.shoup.net/ntl`

[Sil] J. H. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, vol. 106, Springer-Verlag, Berlin, 1986.

[VePr] F. Vercauteren, B. Preneel et J. Vandewalle, *A Memory Efficient Version of Satoh's Algorithm*, Advances in Cryptology - Eurocrypt 2001, Lecture Notes in Comput. Sci., vol. 2045, Springer-Verlag, Berlin, 2001, 1-13.