THALES

université PARIS-SACLAY

Méthode GLV en dimension 4 sur les $\mathbb{Q}$-courbes

Simon Masson

Encadré par Olivier Bernard, Renaud Dubois et Olivier Orcière

Table des matières

1 Courbes de Smith **5**
1.1 $\mathbb{Q}$-courbes quadratiques 5
1.2 Réduction modulo p : construction d'un endomorphisme efficace 6
1.3 La méthode GLS : un cas dégénéré de $\mathbb{Q}$-courbes 10
1.4 Exemples explicites 10

2 Multiplication complexe **13**
2.1 Anneau des endomorphismes 13
2.2 Calcul de l'endomorphisme CM 14
2.3 Des courbes CM parmi les familles de $\mathbb{Q}$-courbes 15
2.4 Recherche exhaustive 18

3 La courbe $4\mathbb{Q}$**-255-19** **20**
3.1 Définition et cardinal 20
3.2 Endomorphismes pour GLV 20
3.3 Décomposition du scalaire 22
3.4 Résultats expérimentaux 24

13 Mars 2017 – 8 Septembre 2017

Introduction

Présentation de l'entreprise

Aujourd'hui, le groupe Thales est un leader mondial des hautes technologies pour les marchés de l'Aéronautique, de l'Espace, de la Défense et des Transports. Fort de 64 000 collaborateurs dans 56 pays, Thales offre une capacité unique pour créer et déployer des équipements, des systèmes et des services pour répondre aux besoins de sécurité les plus complexes. Son implantation internationale exceptionnelle lui permet d'agir au plus près de ses clients partout dans le monde.

Avec Thales Communications & Security, le groupe Thales conforte sa position de numéro un européen des systèmes d'informations et de communications sécurisées pour les marchés mondiaux de la défense, de la sécurité et du transport terrestre.

La laboratoire Chiffre fait partie du service SCC (Service Cryptologie et Composants) lui-même dépendant du service SSI (Sécurité des Systèmes d'Information) de Thales Communications & Security. Il est composé de spécialistes en mathématiques et algorithmie appliquées à la cryptographie. Ses principales missions sont la réalisation d'étude en amont en cryptographie fondamentale, l'intégration d'algorithmes et de mécanismes cryptographiques dans les composants gouvernementaux, la réalisation de dossiers cryptographiques sur des équipements ou systèmes et la participation à des projets de recherches collaboratifs.

Cryptographie et courbes elliptiques

Les courbes elliptiques sont des courbes algébriques qui possèdent une structure particulière : les points d'une courbe elliptique forment un groupe commutatif. On peut utiliser cette structure pour des algorithmes cryptographiques, notamment avec le protocole d'échange de clés ECDH (Elliptic Curve Diffie-Helman) et le protocole de signature ECDSA (Elliptic Curve Digital Signature Algorithm). A titre d'exemple, voici ce qui apparaît sur la page web de *Google* concernant les détails techniques de chiffrement de la page, où apparaissent clairement ECDH et ECDSA :

Connexion chiffrée
(clés TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, 128 bits, TLS 1.2)

La loi est notée $+$ et on peut alors calculer $[2]P = P + P, [3]P = P + P + P$, etc. En cryptographie, on utilise des courbes possédant un sous-groupe cyclique G d'ordre N premier. Tous les éléments du groupe G s'expriment alors de la forme $[k]P$ pour un entier k, et on parle alors de multiplication scalaire sur la courbe.

Les protocoles cryptographiques basés sur les courbes elliptiques utilisent actuellement [1] le problème difficile suivant : si on dispose de deux points P et $[k]P$, il est difficile de retrouver l'entier k. On dit que le problème du logarithme discret sur les courbes elliptiques est difficile. À ce jour, la meilleure attaque générique a une complexité en $\mathcal{O}(\sqrt{N})$. Lors

1. De nouveaux protocoles post-quantiques tels que "supersingular isogeny Diffie Helman" utilisent les courbes elliptiques et se basent sur un autre problème difficile : trouver une isogénie de degré donné entre deux courbes elliptiques. Ces protocoles ne sont pour l'instant pas utilisés en pratique.

d'une signature ECDSA ou d'un échange de clés ECDH, le calcul de $[k]P$ est assez coûteux, et on souhaite avoir des algorithmes efficaces pour ce calcul.

L'objectif du stage est d'étudier la multiplication scalaire par la méthode GLV en dimension 4. Costello et Longa obtiennent dans [1] une courbe avec une sécurité de 128 bits. Dans le cadre d'applications hyper sécurisées relevant de la sécurité nationale, des niveaux de sécurité très conservatifs sont utilisés. On souhaite trouver des courbes pour lesquelles la méthode s'applique apportant des niveaux de sécurité élevés – typiquement 256 bits. Cela permet de limiter l'impact d'éventuelles futures avancées en cryptanalyse, mais également de tenir compte du fait que les preuves de sécurité des protocoles utilisent généralement des hypothèses plus fortes que le logarithme discret, ce qui réduit le niveau de sécurité effectif de la courbe utilisé.

Actuellement, aucune courbe permettant le GLV en dimension 4 avec une telle sécurité n'a été proposée. On devra donc développer des outils de recherche de courbes.

Enfin, on souhaite mesurer l'efficacité de la méthode dans la librairie propriétaire de Thales.

Historique sur la méthode GLV

2001 Gallant, Lambert et Vanstone présentent dans l'artice [2] une multiplication scalaire efficace basée sur des courbes définies sur $\mathbb{F}_p$. Ces courbes utilisent un endomorphisme CM. L'endomorphisme CM ψ agit sur le groupe G comme une multiplication scalaire par λ (c'est-à-dire $\psi(Q) = [\lambda]Q$ pour $Q \in G$). On décompose ensuite $k = k_1 + \lambda k_2 \mod N$ avec k_1, k_2 de taille $\log(k)/2$, puis on calcule $[k]P = [k_1]P + [k_2]\psi(P)$ grâce à une multiexponentiation.
La courbe `secp256k1` qui permet de signer une transaction de *BitCoins* a été choisie parce qu'elle permet une décomposition GLV [2] en dimension 2.
La méthode GLV nécessite un endomorphisme CM de petit degré pour être efficace, et sont en fait peu nombreuses [3]. On souhaite gagner en efficacité avec un second endomorphisme, permettant une décomposition en dimension 4.

2009 Galbraith, Lin et Scott construisent dans [3] de nouveaux endomorphismes efficaces pour des courbes définies sur $\mathbb{F}_{p^2}$, en utilisant les twists quadratiques. Par construction, ces courbes ne sont pas twist-secure (voir l'annexe A).

2013 Smith montre dans [4] comment utiliser la famille des $\mathbb{Q}$-courbes pour construire des endomorphismes efficacement calculables sur $\mathbb{F}_{p^2}$. Cette construction est intéressante dans la mesure où elle fournit un grand nombre de courbes. On peut alors trouver des $\mathbb{Q}$-courbe possédant deux endomorphismes ψ et Ψ, afin d'obtenir une décomposition en dimension 4

$$[k]P = [k_1]P + [k_2]\psi(P) + [k_3]\Psi(P) + [k_4]\psi \circ \Psi(P)$$

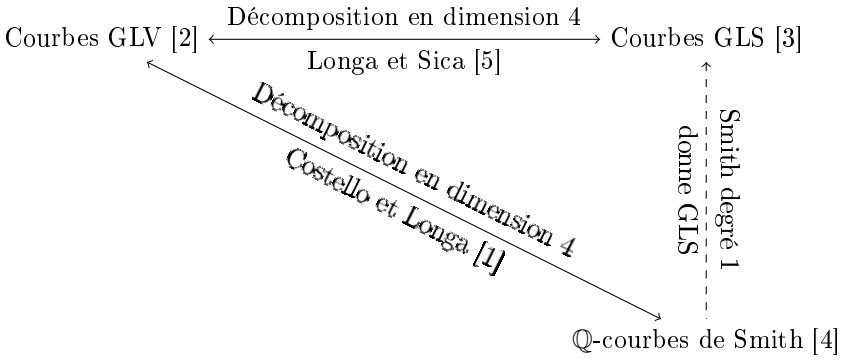
avec $\log(k_i) \simeq \log(k)/4$.

2014 Longa et Sica montrent dans [5] comment construire une courbe GLS avec un endomorphisme CM de petit degré. On peut alors utiliser les deux endomorphismes ψ, Ψ pour appliquer GLV en dimension 4.

2. L'endomorphisme utilisé agit comme une mutliplication scalaire par une racine sixième de l'unité.

3. La méthode GLV de cet article a été implémentée grâce à la librairie GMP avant le stage. Le projet de `C` était limité notamment à cause du peu de courbes proposées dans l'article. Nous allons ici trouver beaucoup plus de courbes, et surtout obtenir une meilleure efficacité.

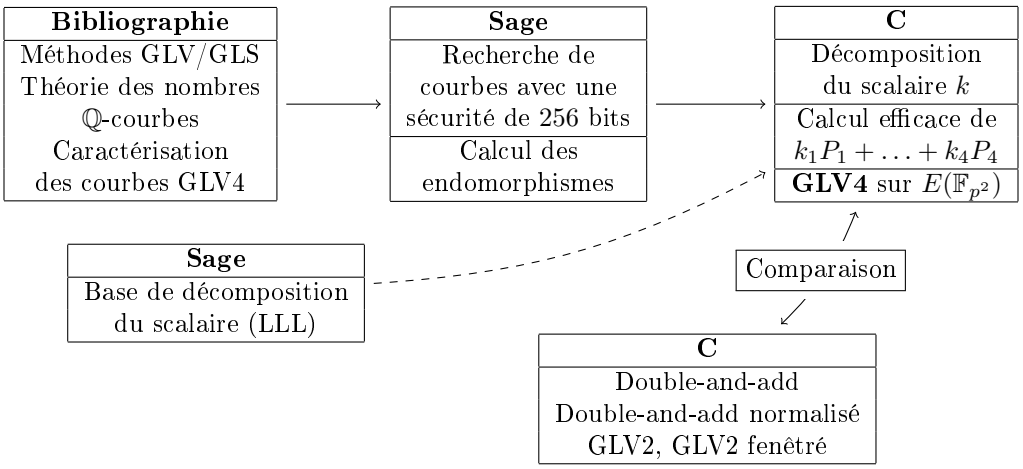
2015 De la même manière, Costello, Longa et Smith montrent dans [1] et [4] comment exhiber des $\mathbb{Q}$-courbes possédant un endomorphisme CM de petit degré. Ils obtiennent la courbe Four$\mathbb{Q}$ définie sur $\mathbb{F}_{p^2}$ avec $p = 2^{127} - 1$, munie de deux endomorphismes, apportant une sécurité de 128 bits.



Contributions

Durant le stage, j'ai dû :

- Lire et comprendre des articles faisant appel à une algorithmie et des notions variées : théorie des nombres, twists et isogénies de courbes elliptiques, $\mathbb{Q}$-courbes, développement en fraction continues, polynôme de classes de Hilbert, etc.
- M'approprier le résultat de la méthode GLV en dimension 4 pour l'étendre sur de nouvelles courbes
- Développer des notions dans plusieurs langages :
  - Sage pour la recherche de courbes, le calcul de la base de décomposition du scalaire, et le calcul des endomorphismes (utilisation des séries de Laurent, de l'algorithme LLL, du polynôme de classes de Hilbert et du calcul d'isogénies)
  - Le langage C pour le calcul efficace de la multiplication scalaire
- Intégrer mon code dans une librairie propriétaire, et donc comprendre le fonctionnement et les contraintes de celle-ci (représentation de Montgomery, représentation des entiers multiprécision)
- Développer dans la librairie propriétaire les outils nécessaires : courbes sur $\mathbb{F}_{p^2}$ génériques, méthode d'exponentiation simultanée $[a]P + [b]Q$ avec fenêtre, et méthode GLV en dimension 2 puis 4.



Nous obtenons ici une décomposition GLV en dimension 4 à partir de la courbe suivante définie sur $\mathbb{F}_{p^2}$ avec $p = 2^{255} - 19$:

$$E(\mathbb{F}_{p^2}) : y^2 = x^3 + \left(-30 + \frac{140}{11} \cdot \sqrt{2}\right) x + \left(56 - \frac{560}{11} \cdot \sqrt{2}\right)$$

Le cardinal de $E(\mathbb{F}_{p^2})$ est de la forme $4 \cdot N$ avec N premier de 508 bits, ce qui donne *une sécurité de 254 bits*. A notre connaissance, il n'y a actuellement aucune proposition publique de courbe GLV4 avec 256 bits de sécurité.

La méthode GLV en dimension 4 sur cette courbe apporte un gain de temps conséquent :

- un gain de *55% par rapport à un Double-and-add* normalisé
- un gain de *24% par rapport à un GLV en dimension 2.*

On présente dans ce rapport :

1. Les courbes de Smith avec une attention particulière sur celles de degé 2, 3, 5 et 7
2. La théorie de la multiplication complexe afin de trouver des courbes de Smith avec un endomorphisme CM de bas degré permettant l'utilisation d'une décomposition en dimension 4
3. La courbe obtenue, ses endomorphismes, et des résultats expérimentaux sur la performance des différentes méthodes de multiplication scalaire.

1 Courbes de Smith

1.1 $\mathbb{Q}$-courbes quadratiques

Soit K est une extension de degré 2 d'un corps k. Dans ce cas, $\mathrm{Gal}(K/k) \simeq \mathbb{Z}/2\mathbb{Z}$. L'automorphisme non-trivial de $\mathrm{Gal}(K/k)$ est appelé la conjugaison et est noté σ.

Définition 1 (courbe, isogénie conjuguée)**.** *Soit $E : y^2 = x^3 + Ax + B$ une courbe elliptique définie sur K. La courbe conjuguée de E est définie par ${}^\sigma E : y^2 = x^3 + {}^\sigma A x + {}^\sigma B$ avec σ l'automorphisme non-trivial de $\mathrm{Gal}(K/k)$. De la même manière, pour une isogénie $\varphi : E_1 \longrightarrow E_2$, on définit ${}^\sigma\varphi : {}^\sigma E_1 \longrightarrow {}^\sigma E_2$ en appliquant σ aux équations définissant φ.*

Proposition 2. *Si $\varphi : E_1 \to E_2$ est une isogénie de degré d, alors ${}^\sigma\varphi$ est aussi de degré d.*

Démonstration. Rappelons la définition du degré de φ. L'isogénie $\varphi : E_1 \longrightarrow E_2$ induit un morphisme $\varphi^* : K(E_2) \longrightarrow K(E_1)$, défini par $\varphi^*(f) = f \circ \varphi$. C'est un morphisme de corps, donc est injectif. On peut donc considérer $\varphi^*(K(E_2))$ comme un sous-corps de $K(E_1)$. On définit alors

$$d := [K(E_1) : \varphi^*(K(E_2))]$$

Soit $(\alpha_1, \ldots, \alpha_d)$ une $\varphi^*(K(E_2))$-base de $K(E_1)$. On montre que $({}^\sigma\alpha_1, \ldots, {}^\sigma\alpha_d)$ est une $({}^\sigma\varphi)^*(K({}^\sigma E_2))$-base de $K({}^\sigma E_1)$.
Soit $g \in K({}^\sigma E_1)$. Il existe $g_1 \in K(E_1)$ tel que $g = {}^\sigma g_1$. On peut alors écrire, pour $a_k \in \varphi^*(K(E_2))$ et $b_k \in K(E_2)$ tel que $a_k = b_k \circ \varphi$,

$$g_1 = \sum_{k=1}^{d} a_k \alpha_k = \sum_{k=1}^{d} (b_k \circ \varphi)\alpha_k$$

En appliquant σ, on obtient :

$$g = {}^\sigma g_1 = \sum_{k=1}^{d} {}^\sigma b_k \circ {}^\sigma\varphi\, {}^\sigma\alpha_k$$

et ${}^\sigma b_k \circ {}^\sigma\varphi \in ({}^\sigma\varphi)^*(K({}^\sigma E_2))$. Finalement, comme les $(\alpha_k)_{1\leq k\leq d}$ forment une base de $K(E_1)$, les $({}^\sigma\alpha_k)_{1\leq k\leq d}$ sont algébriquement indépendants et forment donc une base de $K({}^\sigma E_1)$. □

Définition 3 (corps de nombres quadratique)**.** *Un corps de nombres est une extension finie de $\mathbb{Q}$. Un corps de nombres quadratique est donc un corps K tel que $[K : \mathbb{Q}] = 2$.*

Dans la suite, on considère un corps de nombres quadratique qu'on note $\mathbb{Q}(\sqrt{\Delta})$. La conjugaison sur $\mathbb{Q}(\sqrt{\Delta})$ est notée $\sigma : \sqrt{\Delta} \mapsto -\sqrt{\Delta}$.

Définition 4 ($\mathbb{Q}$-courbe quadratique)**.** *Une $\mathbb{Q}$-courbe quadratique de degré d est une courbe elliptique $\tilde{E}$ définie sur un corps de nombres quadratique K telle que :*

- $\mathrm{End}(\tilde{E}) = \mathbb{Z}$
- *Il existe une isogénie de degré d de $\tilde{E}$ sur sa courbe conjuguée ${}^\sigma\tilde{E}$.*

Remarque 5. Si une courbe $\tilde{E}$ possède une isogénie vers ${}^\sigma\tilde{E}$, mais que $\mathrm{End}(\tilde{E}) \supsetneq \mathbb{Z}$, on parlera de $\mathbb{Q}$-courbe CM.

Par la proposition 2 et en utilisant que ${}^\sigma({}^\sigma\tilde{E}) = \tilde{E}$, on dispose donc de deux isogénies de degré d :

$$\tilde{E} \underset{{}^{\sigma}\tilde{\varphi}}{\overset{\tilde{\varphi}}{\rightleftarrows}} {}^{\sigma}\tilde{E}$$

González montre dans [6] que $\tilde{\varphi}$ et ${}^{\sigma}\tilde{\varphi}$ sont définies sur une extension quadratique de $\mathbb{Q}(\sqrt{\Delta})$ de la forme $\mathbb{Q}(\sqrt{\Delta}, \sqrt{\pm d})$. Dans la suite, on considère des courbes pour lesquelles les isogénies sont définies sur $\mathbb{Q}(\sqrt{\Delta}, \sqrt{-d})$.

En composant ${}^{\sigma}\tilde{\varphi}$ avec $\tilde{\varphi}$, on obtient un endomorphisme ${}^{\sigma}\tilde{\varphi} \circ \tilde{\varphi} \in \mathrm{End}(\tilde{E})$. De manière similaire, on a aussi $\tilde{\varphi} \circ {}^{\sigma}\tilde{\varphi} \in \mathrm{End}({}^{\sigma}\tilde{E})$. La proposition 2 montre que $\tilde{\varphi}$ et ${}^{\sigma}\tilde{\varphi}$ sont de degré d. La composée est donc de degré d^2. La courbe n'ayant pas de multiplication complexe, cet endomorphisme est donc forcément $[\pm d]$ (les seules multiplications par des entiers de degré d^2). On a donc

$$ {}^{\sigma}\tilde{\varphi} \circ \tilde{\varphi} = [\epsilon d]_{\tilde{E}} \qquad \tilde{\varphi} \circ {}^{\sigma}\tilde{\varphi} = [\epsilon d]_{{}^{\sigma}\tilde{E}} \qquad \epsilon \in \{\pm 1\} \tag{1}$$

Remarque 6. En fait, on a montré que sur une $\mathbb{Q}$-courbe $\tilde{E}$, l'isogénie duale de $\tilde{\varphi} : \tilde{E} \longrightarrow {}^{\sigma}\tilde{E}$ est *au signe près* ${}^{\sigma}\tilde{\varphi} : {}^{\sigma}\tilde{E} \longrightarrow \tilde{E}$.

Remarque 7. On a utilisé l'abus de notation ${}^{\sigma}\tilde{\varphi}$ au lieu de ${}^{\tilde{\sigma}}\tilde{\varphi}$, avec $\tilde{\sigma}$ le prolongement de σ à $\mathbb{Q}(\sqrt{\Delta}, \sqrt{-d})$. En fait, le signe ϵ dépend du prolongement de σ choisi dans $\mathrm{Gal}(\mathbb{Q}(\sqrt{\Delta}, \sqrt{-d})$. Si τ est l'autre extension de σ à $\mathbb{Q}(\sqrt{\Delta}, \sqrt{-d})$, ${}^{\tau}\tilde{\varphi} \circ \tilde{\varphi}$ est un autre endomorphisme de $\tilde{E}$, donc forcément ${}^{\tau}\tilde{\varphi} \circ \tilde{\varphi} = [-\epsilon d]_{\tilde{E}}$.

1.2 Réduction modulo p : construction d'un endomorphisme efficace

On souhaite obtenir une courbe définie sur un corps fini. On va donc réduire notre $\mathbb{Q}$-courbe modulo un nombre premier p. Pour que cela ait un sens, on définit $\tilde{E}$ sur l'anneau des entiers $\mathcal{O}_K$ (où $K = \mathbb{Q}(\sqrt{\Delta})$), puis on considère l'anneau quotient $\mathcal{O}_K/p\mathcal{O}_K$.

Ici, on veut garder la structure de $\mathbb{Q}$-courbe et on choisit donc p inerte : $p\mathcal{O}_K$ est premier (c'est-à-dire que $X^2 - \Delta$ est irréductible mod p), et

$$\mathcal{O}_K/p\mathcal{O}_K \simeq \mathbb{F}_p[X]/(X^2 - \Delta) \simeq \mathbb{F}_p[\sqrt{\Delta \bmod p}] \simeq \mathbb{F}_{p^2}$$

On souhaite obtenir modulo p une courbe elliptique. En particulier, il faut que p ne divise pas le discriminant de $\tilde{E}$. On dit que p est de bonne réduction pour $\tilde{E}$.

Enfin, on veut que $\bar{\varphi}$, issue de la réduction mod p des coefficients de φ, soit une isogénie de degré d. On admet que si p et d sont premiers entre eux, le diagramme suivant commute :

$$\begin{array}{ccc} E_1 & \xrightarrow{\quad\varphi\quad} & E_2 \\ \Big\downarrow{\scriptstyle \text{Réduction mod } p} & & \Big\downarrow{\scriptstyle \text{Réduction mod } p} \\ \bar{E}_1 & \xrightarrow{\quad\bar{\varphi}\quad} & \bar{E}_2 \end{array}$$

On a alors les injections suivantes :

$$\langle (p) \rangle := \mathrm{Gal}(\mathbb{F}_p(\sqrt{\Delta})/\mathbb{F}_p) \hookrightarrow \mathrm{Gal}(\mathbb{Q}(\sqrt{\Delta}), \mathbb{Q}) \hookrightarrow \mathrm{Gal}(\mathbb{Q}(\sqrt{\Delta}, \sqrt{-d}), \mathbb{Q})$$

La première injection envoie (p) sur $\sigma : \sqrt{\Delta} \mapsto -\sqrt{\Delta}$ car p est inerte. Pour la seconde injection, il faut choisir un prolongement $\tilde{\sigma}$ de σ dans $\mathrm{Gal}(\mathbb{Q}(\sqrt{\Delta}, \sqrt{-d}), \mathbb{Q})$. On choisit l'image de $\sqrt{-d}$ tel que $\tilde{\sigma}(\sqrt{-d})$ corresponde à l'action de (p) sur $\sqrt{-d \bmod p}$:

$$\tilde{\sigma}(\sqrt{-d}) = \begin{cases} \sqrt{-d} & \text{si } {}^{(p)}\sqrt{-d} = \sqrt{-d} \\ -\sqrt{-d} & \text{sinon} \end{cases} = \left(\frac{-d}{p}\right)\sqrt{-d}$$

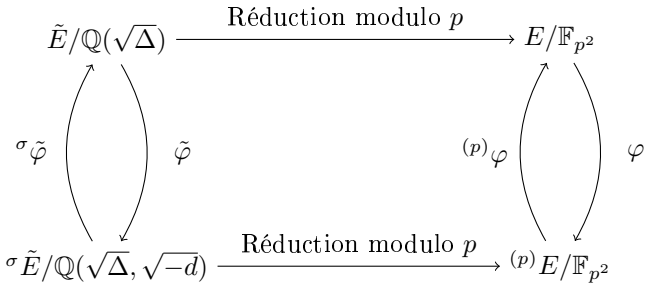
Finalement, pour $\alpha, \beta, \gamma, \delta \in \mathbb{Q}$,

$${}^{\tilde{\sigma}}(\alpha + \beta\sqrt{\Delta} + \gamma\sqrt{-d} + \delta\sqrt{-d\Delta}) = \alpha - \beta\sqrt{\Delta} + (-d/p)(\gamma\sqrt{-d} - \delta\sqrt{-d\Delta})$$

car (p) fixe $\sqrt{-d}$ si $\sqrt{-d} \in \mathbb{F}_p$, c'est-à-dire si $-d$ est un carré modulo p.

Remarque 8. Par définition, ${}^{\tilde{\sigma}}\tilde{\varphi}$ est l'isogénie $\tilde{\varphi}$ à laquelle on a appliqué σ sur ses coefficients. On obtient modulo p une isogénie ${}^{(p)}\varphi$, qui est φ à laquelle on a élevé à la puissance p tous ses coefficients. (p) correspond à la conjugaison de $\mathrm{Gal}(\mathbb{F}_p(\sqrt{\Delta})/\mathbb{F}_p)$.

On peut alors réduire modulo p les courbes $\tilde{E}, {}^{\sigma}\tilde{E}$ et les isogénies $\tilde{\varphi}, {}^{\sigma}\tilde{\varphi}$. On obtient :



On a alors le résultat analogue sur les courbes définies sur $\mathbb{Q}$ et sur les réductions mod p :

$${}^{(p)}\varphi \circ \varphi = [\epsilon_p d]_E \qquad \varphi \circ {}^{(p)}\varphi = [\epsilon_p d]_{{}^{(p)}E}$$

ϵ_p dépend bien de p pour la même raison que celle indiquée en remarque 7.
L'application $(x, y) \mapsto (x^p, y^p)$ définit deux isogénies de degré p :

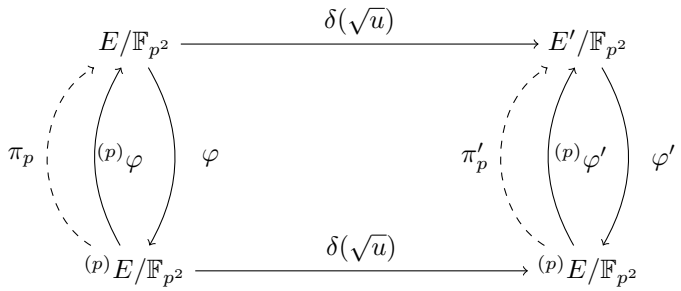
$$\pi_p : {}^{(p)}E \longrightarrow E \qquad {}^{(p)}\pi_p : E \longrightarrow {}^{(p)}E$$

On note $\pi_E : (x, y) \mapsto (x^q, y^q)$ pour une courbe E définie sur $\mathbb{F}_q$. On a ici $\pi_p \circ {}^{(p)}\pi_p = \pi_E$. On considère maintenant $\psi := \pi_p \circ \varphi \in \mathrm{End}(E)$. C'est un endomorphisme de degré pd. Si d est petit (disons $d \leq 10$), on peut calculer ψ facilement puisque φ est définie par des polynômes de degré environ [4] d et que π_p se calcule simplement par une inversion de signe dans $\mathbb{F}_p$.

Théorème 9 (Smith, [4]). *L'endomorphisme ψ vérifie $\psi^2 = [\epsilon_p d]\pi_E$. Il existe $r \in \mathbb{Z}$ tel que $dr^2 = 2p + \epsilon_p t_E$, pour lequel $[r]\psi = [p] + \epsilon_p \pi_E$. Le polynôme caractéristique de ψ est $P_\psi(T) = T^2 - rdT + dp$.*

4. Voir l'annexe B

Remarque 10. On a sur le twist quadratique[5] de E un endomorphisme correspondant. En notant $\delta(\sqrt{u})$ l'isomorphisme entre la courbe et son twist quadratique, avec $u \notin \mathbb{F}_{p^2}^{\times 2}$,



ψ' vérifie les propriétés suivantes : $(\psi')^2 = [-\epsilon_p d]\pi_{E'}$, $[r]\psi' = [p] - \epsilon_p \pi_E$ et $P_{\psi'}(T) = P_\psi(T)$.

Démonstration. Par la remarque 8, on sait que ${}^{(p)}\varphi$ est l'isogénie φ avec les coefficients élevés à la puissance p. On rappelle qu'en caractéristique p, le Frobénius est un isomorphisme de corps, et donc

$$P(x)^p = \left(\sum_{k=0}^{n} a_k x^k\right)^p = \sum_{k=0}^{n} a_k^p (x^p)^k = {}^{(p)}P(x^p)$$

Dans notre cas, on a pour $P = (x, y) \in E(\mathbb{F}_{p^2})$,

$$\begin{aligned}\pi_p \circ \varphi(x,y) &= (\varphi_1(x,y)^p, \varphi_2(x,y)^p) \\ &= ({}^{(p)}\varphi_1(x^p, y^p), {}^{(p)}\varphi_2(x^p, y^p)) \\ &= {}^{(p)}\varphi \circ {}^{(p)}\pi_p(x,y)\end{aligned}$$

c'est-à-dire $\pi_p \circ \varphi = {}^{(p)}\varphi \circ {}^{(p)}\pi_p$. On en déduit

$$\psi^2 = \pi_p \varphi \pi_p \varphi = \pi_p ({}^{(p)}\varphi)^{(p)}\pi_p = \pi_p [\epsilon_p d]_{{}^{(p)}E}\, {}^{(p)}\pi_p = [\epsilon_p d]_E \pi_p\, {}^{(p)}\pi_p = [\epsilon_p d]\pi_E$$

Comme ψ est de degré dp, son polynôme caractéristique est de la forme $P_\psi(T) = T^2 - aT + dp$ pour un entier a. D'où

$$[a]\psi = \psi^2 + [dp] = [\epsilon_p d]\pi_E + [dp]$$

En élevant au carré, on a

$$[a^2]\psi^2 = ([\epsilon_p d]\pi_E)^2 + [2\epsilon_p d^2 p]\pi_E + [d^2 p^2]$$

En modifiant l'expression du dernier terme (en utilisant le polynôme caractéristique de π_E), on obtient

$$[a^2]\underbrace{[\epsilon_p d]\pi_E} = (\underbrace{[\epsilon_p d]\pi_E})^2 + [2pd]\underbrace{[\epsilon_p d]\pi_E} + [\epsilon_p d t_E]\underbrace{[\epsilon_p d]\pi_E} - (\underbrace{[\epsilon_p d]\pi_E})^2$$

$$\left([a^2] - [2pd] - [\epsilon_p d t_E]\right)[\epsilon_p d]\pi_E = 0$$

et donc

$$a^2 = 2pd + \epsilon_p d t_E$$

5. voir l'annexe A pour la définition du twist quadratique d'une courbe

On obtient alors que d divise a^2, mais d n'est pas un carré donc il existe un entier r tel que $a = dr$. On a donc $d^2r^2 = 2pd + \epsilon_p dt_E$ et en simplifiant par d, r vérifie bien l'équation du théorème. Il suffit alors de remplacer $a = dr$ dans l'équation $[a]\psi = [\epsilon_p d]\pi_E + [dp]$ pour obtenir que $[r]\psi = [p] + \epsilon_p \pi_E$. En utilisant le polynôme caractéristique de π_E et P_ψ, on montre que $a = dr$ pour un entier r qui vérifie l'équation du théorème.

□

Quand $r \neq 0$, on peut donc obtenir une valeur propre de ψ et pouvoir appliquer la décomposition de k. On montre par la suite que $r \neq 0 \iff E$ et son twist sont ordinaires. On commence par rappeler les définitions de courbes ordinaire et super-singulière, données dans [7].

Définition 11 (courbe super-singulière, ordinaire)**.** *Une courbe elliptique E définie sur un corps de caractéristique p est dite super-singulière si, et seulement si, $E[p] \simeq \{0\}$. Si $E[p] \simeq \mathbb{Z}/p\mathbb{Z}$, on dit que E est ordinaire.*

Proposition 12. *Une courbe E définie sur $\mathbb{F}_{p^n}$ est super-singulière si, et seulement si, $p \mid t_E$.*

Démonstration.

$\implies$ Si E est super-singulière, on a $\ker[p] = \{0\} = \ker \pi_p \hat{\pi}_p \supset \ker \hat{\pi}_p$ donc $\ker \hat{\pi}_p = \{0\}$. Comme $\deg(\hat{\pi}_p) = p > 1$, $\hat{\pi}_p$ est inséparable. On en déduit que $\hat{\pi}_p^n = \widehat{\pi_p^n} = \hat{\pi}_E$ est inséparable. De plus, π_E est inséparable. On en déduit que $[t_E]$ est inséparable car $[t_E] = \pi_E + \hat{\pi}_E$. Donc $p \mid \deg([t_E]) = t_E^2$ d'où $p \mid t_E$.

$\impliedby$ Si $p \mid t_E$, $p \mid t_E^2 = \deg([t_E])$. Donc $[t_E]$ est inséparable. De plus $\hat{\pi}_E$ est inséparable car $\hat{\pi}_E = [t_E] - \pi_E$. Donc $\hat{\pi}_p^n$ est inséparable et donc $\hat{\pi}_p$ aussi. Donc $\ker \hat{\pi}_p = \{0\}$ puisque $\hat{\pi}_p$ est de degré premier p. Le même raisonnement est valable pour π_p et on en déduit que le noyau de $[p] = \hat{\pi}_p \pi_p$ est trivial, et donc E est super-singulière.

□

Dans le cas des $\mathbb{Q}$-courbes, on obtient :

Proposition 13. *E et E' sont ordinaires si, et seulement si, $r \neq 0$.*

Démonstration. E et E' sont super-singulières $\iff$ p divise t_E $\iff$ p divise r (car $dr^2 = 2p + \epsilon_p t_E$ et p ne divise pas d). Le théorème de Hasse donne $|t_E| \leq 2p$ donc $|r| \leq 2\sqrt{p/d}$. Le seul entier divisible par p dans cet intervalle est $r = 0$. □

On a alors le corollaire du théorème 9 :

Corollaire 14 (Smith, [4])**.** *Soit E une courbe elliptique ordinaire. Si $G \subset E(\mathbb{F}_{p^2})$ est un sous-groupe cyclique d'ordre N tel que $\psi(G) \subset G$, alors la valeur propre de ψ sur G est*

$$\lambda_\psi \equiv \frac{p + \epsilon_p}{r} \mod N$$

Sur le twist quadratique, on a l'analogue avec $\lambda_{\psi'} \equiv \frac{p - \epsilon_p}{r} \mod \#G'$ pour G' sous-groupe du twist quadratique de E.

Pour avoir une méthode GLV efficace, il faut que les endomorphismes ne soient pas trop difficiles à calculer. Ici, pour une $\mathbb{Q}$-courbe de degré d, on doit calculer ψ (de degré pd) à partir de φ de degré d. On va donc chercher une $\mathbb{Q}$-courbe de degré ≤ 7.

1.3 La méthode GLS : un cas dégénéré de $\mathbb{Q}$-courbes

Si $\tilde{\varphi} : \tilde{E} \longrightarrow {}^{\sigma}\tilde{E}$ est de degré 1, les courbes $\tilde{E}$ et ${}^{\sigma}\tilde{E}$ sont isomorphes. On a alors

$$j(\tilde{E}) = j({}^{\sigma}\tilde{E}) = {}^{\sigma}j(\tilde{E})$$

donc $j(\tilde{E}) \in \mathbb{Q}$ et $\tilde{E}$ est isomorphe à une courbe définie sur $\mathbb{Q}$. On peut donc supposer que $\tilde{E}$ est définie sur $\mathbb{Q}$ et étendue à $\mathbb{Q}(\sqrt{\Delta})$. Alors, $\tilde{E} = {}^{\sigma}\tilde{E}$ et on peut choisir $\tilde{\varphi} = \mathrm{id}$. En réduisant modulo p, on obtient que $\psi^2 = \pi_p^2 = \pi_E$. Donc ψ a pour valeurs propres ± 1 sur un des sous-groupes de $E(\mathbb{F}_{p^2})$. Sur le twist $E'(\mathbb{F}_{p^2})$, ψ' a pour valeur propre $\sqrt{-1}$ sur un sous-groupe : on a retrouvé la méthode GLS. On pourra consulter [3] pour plus de détails.

L'un des inconvénient de ces courbes est qu'elles ne sont pas twist-secure : modulo p, la courbe $E(\mathbb{F}_{p^2})$ a un j-invariant dans $\mathbb{F}_p$, et donc $E(\mathbb{F}_p)$ est un sous-groupe de $E(\mathbb{F}_{p^2})$. Finalement, $\#E(\mathbb{F}_p) \mid \#E(\mathbb{F}_{p^2})$ et $\#E(\mathbb{F}_{p^2})$ possède un facteur de la taille de p.

1.4 Exemples explicites

Courbes de degré 2

Hasegawa donne dans [8] une famille de $\mathbb{Q}$-courbes de degré 2 définies par

$$\tilde{E}_{2,\Delta,s} : y^2 = x^3 + 2(C_{2,\Delta}(s) - 24)x - 8(C_{2,\Delta}(s) - 16)$$

avec $C_{2,\Delta}(s) := 9(1 + s\sqrt{\Delta})$, Δ entier sans facteur carré et $s \in \mathbb{Q}$. Pour se convaincre que $\tilde{E}_{2,\Delta,s}$ est bien une $\mathbb{Q}$-courbe, explicitons l'isogénie de degré 2 de $\tilde{E}_{2,\Delta,s}$ sur ${}^{\sigma}\tilde{E}_{2,\Delta,s}$. On observe déjà que ${}^{\sigma}\tilde{E}_{2,\Delta,s} = \tilde{E}_{2\Delta,-s}$. La courbe possède un point de 2-torsion rationnel $(4,0)$ qui nous permet de trouver une isogénie [6]

$$f : \tilde{E}_{2,\Delta,s} \longrightarrow \tilde{E}_{2,\Delta,s}/\langle(4,0)\rangle$$

Les formules de Vélu permettent aussi d'expliciter les coefficents qui définissent la courbe $\tilde{E}_{2,\Delta,s}/\langle(4,0)\rangle$, et on peut alors vérifier que $\tilde{E}_{2,\Delta,s}/\langle(4,0)\rangle = ({}^{\sigma}\tilde{E}_{2,\Delta,s})^{\sqrt{-2}}$. En twistant par $1/\sqrt{-2}$, on obtient une isogénie

$$\begin{array}{rccc} \tilde{\varphi}_{2,\Delta,s} : & \tilde{E}_{2,\Delta,s} & \xrightarrow[\text{Vélu}]{f} \tilde{E}_{2,\Delta,s}/\langle(4,0)\rangle = \left({}^{\sigma}\tilde{E}_{2,\Delta,s}\right)^{\sqrt{-2}} & \xrightarrow[\text{twist}]{\delta(1/\sqrt{-2})} {}^{\sigma}\tilde{E}_{2,\Delta,s} \\ & (x,y) & \longmapsto \left(-\frac{x}{2} - \frac{C_{2,\Delta}(s)}{x-4}, \frac{y}{\sqrt{-2}}\left(\frac{-1}{2} + \frac{C_{2,\Delta}(s)}{(x-4)^2}\right)\right) & \end{array}$$

Proposition 15.

$${}^{\sigma}\tilde{\varphi}_{2,\Delta,s} \circ \tilde{\varphi}_{2,\Delta,s} = [\epsilon 2], \text{ avec } \epsilon = \begin{cases} -1 & \text{si } {}^{\sigma}\sqrt{-2} = \sqrt{-2} \\ +1 & \text{si } {}^{\sigma}\sqrt{-2} = -\sqrt{-2} \end{cases}$$

Démonstration. L'isogénie $\tilde{\varphi}_{2,\Delta,s}$ a un facteur de normalisation [7] $\Lambda(\tilde{\varphi}_{2,\Delta,s}) = \sqrt{-2}$. De plus, on peut montrer que $\Lambda([n]) = n$. On peut alors calculer le facteur de normalisation du dual de $\tilde{\varphi}_{2,\Delta,s}$: comme $\widehat{\tilde{\varphi}_{2,\Delta,s}} \circ \tilde{\varphi}_{2,\Delta,s} = [2]$, on a $\Lambda(\widehat{\tilde{\varphi}_{2,\Delta,s}}) \cdot \Lambda(\tilde{\varphi}_{2,\Delta,s}) = 2$, et donc $\Lambda(\widehat{\tilde{\varphi}_{2,\Delta,s}}) =$

6. grâce aux formules de Vélu, en annexe B
7. Voir l'annexe B pour plus de détails

$-\sqrt{-2}$. De même, on a vu en (1) que ${}^\sigma\tilde{\varphi}_{2,\Delta,s} \circ \tilde{\varphi}_{2,\Delta,s} = [\epsilon \cdot 2]$ donc $\Lambda({}^\sigma\tilde{\varphi}_{2,\Delta,s}) = -\epsilon\sqrt{-2}$. Finalement, $\Lambda({}^\sigma\tilde{\varphi}_{2,\Delta,s}) = {}^\sigma\Lambda(\tilde{\varphi}_{2,\Delta,s}) = {}^\sigma\sqrt{-2}$ et donc

$$\epsilon = \frac{-{}^\sigma\sqrt{-2}}{\sqrt{-2}} = \begin{cases} -1 & \text{si } \sigma \text{ fixe } \sqrt{-2} \text{ (i.e si } -2 \in \mathbb{F}_p^{\times 2}) \\ 1 & \text{sinon} \end{cases}$$

□

Le discriminant et le j-invariant de $\tilde{E}_{2,\Delta,s}$ sont donnés par les formules suivantes :

$$\mathrm{Disc}(\tilde{E}_{2,\Delta,s}) = 2^9 \cdot C_{2,\Delta}(s)^2 \cdot {}^\sigma C_{2\Delta}(s)$$

$$j(\tilde{E}_{2,\Delta,s}) = \frac{-12^3(C_{2,\Delta}(s) - 24)^3}{C_{2,\Delta}(s)^2 \cdot {}^\sigma C_{2,\Delta}(s)}$$

En réduisant modulo un $p > 3$ inerte, on obtient une courbe définie sur $\mathbb{F}_{p^2}$ pour chaque valeur de $s \in \mathbb{F}_p$. Smith obtient alors un premier endomorphisme de $\tilde{E}_{2,\Delta,s}$ grâce au théorème 9 et au corollaire 14.

Théorème 16 (Smith, [4]). *Soit $p > 3$ premier, et Δ non-carré modulo p, de sorte que $\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{\Delta})$. Soient $E_{2,\Delta,s}$ et $\varphi_{2,\Delta,s}$ les réductions modulo p de $\tilde{E}_{2,\Delta,s}$ et $\tilde{\varphi}_{2,\Delta,s}$. Pour chaque $s \in \mathbb{F}_p$, la courbe $E_{2,\Delta,s}/\mathbb{F}_{p^2}$ possède l'endomorphisme*

$$\psi_{2,\Delta,s} := \pi_p \circ \varphi_{2,\Delta,s}$$

de degré $2p$ tel que $\psi_{2,\Delta,s}^2 = [\epsilon_p 2]\pi_{E_{2,\Delta,s}}$ et $(\psi'_{2,\Delta,s})^2 = [-\epsilon_p 2]\pi_{E'_{2,\Delta,s}}$ avec

$$\epsilon_p := -(-2/p) = \begin{cases} -1 & \text{si } p \equiv 1, 3 \mod 8 \\ 1 & \text{si } p \equiv 5, 7 \mod 8 \end{cases}$$

Il existe un entier r vérifiant $2r^2 = 2p + \epsilon_p t_{E_{2,\Delta,s}}$ tel que

$$[r]\psi_{2,\Delta,s} = [p] + \epsilon_p \pi_{E_{2,\Delta,s}} \qquad [r]\psi'_{2,\Delta,s} = [p] - \epsilon_p \pi_{E'_{2,\Delta,s}}$$

Si $E_{2,\Delta,s}$ est ordinaire et possède un sous-groupe cyclique G d'ordre N stable par $\psi_{2,\Delta,s}$, alors la valeur propre de $\psi_{2,\Delta,s}$ sur G est donnée par

$$\lambda_{2,\Delta,s} \equiv (p + \epsilon_p)/r \equiv \pm\sqrt{\epsilon_p 2} \mod N$$

Courbes de degré 3, 5 et 7

On peut raisonner de même pour les courbes de degré $d = 3, 5$ et 7, données par Hasegawa dans [8]. On construit un sous-groupe d'ordre d grâce aux polynômes de division. Lorsque p ne divise pas d (ici, $d \ll p$ premier donc c'est bien le cas), les points de d-torsion forment un espace vectoriel de dimension 2 sur $\mathbb{F}_p$:

$$E[d] \simeq \mathbb{Z}/d\mathbb{Z} \times \mathbb{Z}/d\mathbb{Z}$$

Les abscisses des points de $E[d]$ sont les racines du polynôme de division P_d. On choisit donc une racine qui va générer un sous-groupe $\mathbb{Z}/d\mathbb{Z}$ de $E[d]$, d'ordre d. Grâce aux formules de

Vélu, on obtient l'isogénie de E sur sa courbe conjuguée, puis on utilise le Frobenius pour avoir un endomorphisme de E.

On n'explicitera pas les endomorphismes obtenus ici, mais tous les détails sont donnés dans [4, §6–8]. Pour les formules de Vélu, on pourra lire [9, page 121]. On donne ici simplement les équations des courbes et les j-invariants associés, car on en aura besoin par la suite. Notons que dans le cas du degré 5, la famille de $\mathbb{Q}$-courbes d'Hasegawa est paramétrée par $s \in \mathbb{Q}$ pour Δ fixé vérifiant la condition de [8, proposition 2.3]. On utilise ici $\Delta = -1$. Pour d'autres valeurs de Δ, on peut obtenir de nouvelles courbes.

Degré 3

$$\tilde{E}_{3,\Delta,s} : y^2 = x^3 - 3(2C_{3,\Delta}(s) + 1)x + (C_{3,\Delta}(s)^2 + 10C_{3,\Delta}(s) - 2)$$

$$\text{avec } C_{3,\Delta}(s) := 2(1 + s\sqrt{\Delta})$$

$$j(\tilde{E}_{3,\Delta,s}) = \frac{2^8 \cdot 3^3 \cdot (2C_{3,\Delta}(s) + 1)^3}{C_{3,\Delta}(s) \cdot {}^{\sigma}C_{3,\Delta}(s)}$$

Degré 5, $\Delta = -1$

$$\tilde{E}_{5,-1,s} : y^2 = x^3 + A_{5,-1}(s)x + B_{5,-1}(s)$$

$$\text{avec } A_{5,-1}(s) := -27s(11s - 2)(3(6s^2 + 6s - 1) - 20s(s - 1)\sqrt{-1})$$
$$\text{et } B_{5,-1}(s) := 54s^2(11s - 2)^2((13s^2 + 59s - 9) - 2(s - 1)(20s + 9)\sqrt{-1})$$

$$j(\tilde{E}_{5,-1,s}) = \frac{-64(3(6s^2 + 6s - 1) - 20(s^2 - s)\sqrt{-1})^3}{(1 + s^2)(1 + s\sqrt{-1})^4}$$

Degré 7

$$\tilde{E}_{7,\Delta,s} : y^2 = x^3 + A_{7,\Delta}(s)x + B_{7,\Delta}(s)$$

$$\text{avec } A_{7,\Delta}(s) := -3C_{7,\Delta}(s)(85 + 96s\sqrt{\Delta} + 15s^2\Delta)$$
$$B_{7,\Delta}(s) := 14C_{7,\Delta}(s)(9(3s^4\Delta^2 + 130s^2\Delta + 171) + 16(9s^2\Delta + 163)s\sqrt{\Delta})$$
$$\text{et } C_{7,\Delta}(s) := 7(27 + s^2\Delta)$$

$$j(\tilde{E}_{7\Delta,s}) = \frac{(27 + s^2\Delta)(85 + 96s\sqrt{\Delta} + 15s^2\Delta)^3}{(1 - s^2\Delta)(1 - s\sqrt{\Delta})^6}$$

2 Multiplication complexe

L'objectif est d'obtenir une décomposition en dimension 4, et on a donc besoin de deux endomorphismes. Les courbes de Smith nous ont permis de construire un endomorphisme issu de la structure galoisienne de la $\mathbb{Q}$-courbe. Le second sera trouvé par le caractère CM de la courbe. On note $\mathrm{End}(E)$ l'anneau des endomorphismes d'une courbe elliptique E.

2.1 Anneau des endomorphismes

On aura besoin de la notion d'ordre sur un corps quadratique imaginaire pour décrire l'anneau des endomorphismes d'une courbe elliptique. On commence par définir l'ordre maximal : l'anneau des entiers. Sur un corps quadratique imaginaire $K = \mathbb{Q}(\sqrt{a})$ avec a négatif sans facteur carré, on définit le discriminant de K

$$d_K := \begin{cases} a & \text{si } a \equiv 1 \mod 4 \\ 4a & \text{sinon.} \end{cases}$$

L'anneau des entiers de $\mathbb{Q}(\sqrt{a})$ est alors

$$\mathcal{O}_{\mathbb{Q}(\sqrt{a})} = \mathbb{Z}\left[\frac{d_K + \sqrt{d_K}}{2}\right] = \begin{cases} \mathbb{Z}\left[\frac{1+\sqrt{a}}{2}\right] & \text{si } a \equiv 1 \mod 4 \\ \mathbb{Z}[\sqrt{a}] & \text{si } a \not\equiv 1 \mod 4 \end{cases}$$

Définition 17 (Courbe CM)**.** *Une courbe est à multiplication complexe si* $\mathrm{End}(E) \neq \mathbb{Z}$.

Remarque 18. On a toujours $\mathbb{Z} \subset \mathrm{End}(E)$ car les multiplications scalaires $[n]$ sont des endomorphismes de E.

Remarque 19. Une courbe définie sur un corps fini $\mathbb{F}_q$ est CM car $\mathbb{Z}[\sqrt{t^2 - 4q}] \subset \mathrm{End}(E)$. En effet, le Frobenius est annulé par le polynôme $X^2 - tX + q$, de discriminant $\sqrt{t^2 - 4q}$.

On souhaite donc caractériser l'anneau des endomorphismes d'une courbe. On a le théorème suivant :

Théorème 20. *L'anneau des endomorphismes d'une courbe elliptique est soit $\mathbb{Z}$, soit un ordre sur un corps quadratique imaiginaire, soit un ordre sur une algèbre de quaternions.*

Démonstration. Voir [7, §9.4 page 102]. □

Il s'avère qu'en caractéristique 0, comme c'est le cas sur $\mathbb{Q}(\sqrt{\Delta})$, $\mathrm{End}(E)$ n'est jamais un ordre sur une algèbre de quaternions. On cherche donc ici des courbes telles que $\mathrm{End}(E)$ soit un ordre sur un corps quadratique imaginaire :

$$\mathbb{Z} \subsetneq \mathrm{End}(E) \subseteq \mathcal{O}_K$$

Rappelons la notion d'ordre sur un corps quadratique :

Définition 21 (ordre d'un corps quadratique)**.** *Un ordre d'un corps quadratique K est un sous-ensemble $\mathcal{O} \subset K$ tel que*

1. *$\mathcal{O}$ est un sous-anneau de K contenant* 1
2. *$\mathcal{O}$ est un $\mathbb{Z}$-module de type fini*
3. *$\mathcal{O}$ contient une $\mathbb{Q}$-base de K.*

Remarque 22. L'anneau des entiers $\mathcal{O}_K$ de K est un ordre de K, et c'est même l'ordre maximal de K.

Proposition 23. *Soit $\mathcal{O}$ un ordre d'un corps quadratique K. Alors, $\mathcal{O}$ est d'incide fini dans $\mathcal{O}_K$.*
Pour $f := [\mathcal{O}_K : \mathcal{O}]$, $\mathcal{O} = \mathbb{Z} + f\mathcal{O}_K$ et une base de $\mathcal{O}$ est $[1, fw_K]$ avec $w_K = \frac{d_K + \sqrt{d_K}}{2}$.

Démonstration. Voir [10] Lemme 7.2 p133. □

f est appelé le conducteur de $\mathcal{O}$. Un invariant de $\mathcal{O}$ est son discriminant D. Si $[\alpha, \beta]$ est une base de $\mathcal{O}$, et si $\alpha \mapsto \alpha'$ est l'automorphisme non-trivial de K (la conjugaison),

$$D = \left(\det \begin{pmatrix} \alpha & \beta \\ \alpha' & \beta' \end{pmatrix}\right)^2$$

D est indépendant de la base choisie, et avec la base $[1, fw_K]$, on obtient que $D = f^2 d_K$. On peut vérifier que $\mathrm{Disc}(\mathcal{O}_K) = d_K$.

Exemple 24. Quand $K = \mathbb{Q}(\sqrt{-3})$, $\mathcal{O} = \mathbb{Z}[\sqrt{-3}]$ est un ordre de K de conducteur 2 et de discriminant -12, et $\mathcal{O} \subsetneq \mathcal{O}_K = \mathbb{Z}\left[\frac{1+\sqrt{-3}}{2}\right]$ est l'ordre maximal de K de discriminant $-3 = d_K$.

On termine cette sous-section avec la définition du polynôme de classes Hilbert H_D, où $\mathcal{O}_D$ désigne l'ordre de discriminant D.

$$H_D(X) := \prod_{E/End(E)=\mathcal{O}_D} (X - j(E))$$

Nous admettrons par la suite que ce polynôme est à coefficients entiers, unitaire et irréductible sur $\mathbb{Z}$. Pour une preuve de ce résultat, on pourra lire [10, théorème 11.1].

2.2 Calcul de l'endomorphisme CM

Afin d'obtenir une deuxième description de la multiplication complexe sur les courbes elliptiques, on explique le lien entre courbes elliptiques et réseaux.

Définition 25. *Un réseau est un sous-groupe additif L de $\mathbb{C}$ généré par deux nombres complexes w_1, w_2 linéairement indépendants sur $\mathbb{R}$. On note $L = [w_1, w_2]$.*

Définition 26. *Une fonction elliptique est une fonction f définie sur $\mathbb{C}$ (à l'exception de singularités isolées), méromorphe sur $\mathbb{C}$, et qui vérifie $f(z + w) = f(z)$ pour tout $w \in L$.*

Nous introduisons ici la fonction elliptique de Weierstrass :

$$\wp(z, L) = \frac{1}{z^2} + \sum_{w \in L, w \neq 0} \left(\frac{1}{(z-w)^2} - \frac{1}{w^2}\right)$$

Lorsque le réseau L est fixé, on utilise l'abus de langage $\wp(z)$. On pourra se référrer à Cox pour quelques propriétés concernant la fonction de Weierstrass. Un résultat important est qu'elle vérifie l'équation différentielle

$$\wp'(z)^2 = 4\wp(z)^3 - g_2(L)\wp(z) - g_3(L)$$

avec

$$g_2(L) = 60 \sum_{w \in L, w \neq 0} \frac{1}{w^4} \qquad g_3(L) = 140 \sum_{w \in L, w \neq 0} \frac{1}{w^6}$$

En ce sens, la fonction de Weierstrass permet de paramétriser une courbe elliptique sur $\mathbb{C}$, grâce au paramétrage $(\wp(z), \wp'(z))$ avec $z \in \mathbb{C}$.
Le lien avec la multiplication complexe est donné dans le théorème suivant [10, page 209] :

Théorème 27. *Soit L un réseau, et $\wp(z)$ la fonction de Weierstrass associée. Alors, pour $\alpha \in \mathbb{C} - \mathbb{Z}$, on a les équivalences entre*

1. *$\wp(\alpha z)$ est une fraction rationnelle en $\wp(z)$*
2. *$\alpha L \subset L$*
3. *Il existe un ordre quadratique $\mathcal{O}$ d'un corps quadratique imaginaire K tel que $\alpha \in \mathcal{O}$.*

De plus, si ces conditions sont vérifiées, $\wp(\alpha z)$ peut être écrit de la forme

$$\wp(\alpha z) = \frac{A(\wp(z))}{B(\wp(z))}$$

avec $A(x)$ et $B(x)$ premiers entre eux de degrés

$$\deg(A) = \deg(B) + 1 = N(\alpha)$$

Ce théorème montre que si une fonction elliptique possède une multiplication par $\alpha \in \mathbb{C} - \mathbb{R}$, elle possède une multiplication par tous les éléments de l'ordre quadratique $\mathcal{O}$. Le problème de la multiplication complexe est donc de trouver $\alpha \in \mathbb{C} - \mathbb{Z}$ qui stabilise le réseau L.

On peut utiliser cette paramétrisation de la courbe elliptique pour expliciter l'expression de l'endomorphisme $[\alpha]$ grâce à l'algorithme de Stark.

Si notre courbe possède une multiplication complexe par α, le théorème 27 montre que $\wp(\alpha z)$ est une fraction rationnelle en $\wp(z)$:

$$\wp(\alpha z) = \frac{f(\wp(z))}{g(\wp(z))}$$

avec $\deg(f) = \deg(g) + 1 = N(\alpha)$.
Pour déterminer cette fraction rationnelle, on utilise l'algorithme de développement en fractions continues expliqué dans [11, page 157]. On pourra aussi consulter le code Sage en annexe D. Une fois cette fraction trouvée, si $P = (\wp(z), \wp'(z))$, on a

$$x_{[\alpha]P} = \wp(\alpha z) = \frac{f(\wp(z))}{g(\wp(z))} = \frac{f(x_P)}{g(x_P)}$$

De manière analogue, on trouve l'expression de $y_{[\alpha]P}$ avec les dérivées de f et g, et on a donc explicité l'expression de la multiplication par $[\alpha]$.

2.3 Des courbes CM parmi les familles de $\mathbb{Q}$-courbes

Nous allons désormais rechercher les courbes CM parmi les courbes de Smith. Ce sont des courbes E telles que $\mathrm{End}(E)$ est un ordre sur un corps quadratique imaginaire. On note

D le discriminant de $\mathcal{O} := \mathrm{End}(E)$. Par exemple, si $\mathcal{O} = \mathbb{Z}[\sqrt{-3}]$, on a un endomorphisme $[\sqrt{-3}]$.

On suit l'exposition de Benjamin Smith [4, §9]. Si on note $\mathcal{O}_D$ l'anneau d'endomorphismes de $E_{d,\Delta,s}$, $\mathrm{End}(E_{d,\Delta,s}) = \mathrm{End}({}^{\sigma}E_{d,\Delta,s})$. Autrement dit, $j(E_{d,\Delta,s})$ et ${}^{\sigma}j(E_{d,\Delta,s})$ sont deux racines du polynome de Hilbert $H_D \in \mathbb{Z}[X]$. Comme il est irréductible, on en déduit que ces deux j-invariants sont les deux seuls possibles quand $\mathrm{End}(E) = \mathcal{O}_D$, et $\deg(H_D) = 1$ ou 2. Or, il n'y a qu'un nombre fini de discriminants possibles pour un polynôme de Hilbert de degré 1 ou 2 :

$-D_0$	-3	-3	-3	-4	-4	-7	-7	-8	-11	-19	-43	-67	-163
f	1	2	3	1	2	1	2	1	1	1	1	1	1
D	-3	-12	-27	-4	-16	-7	-28	-8	-11	-19	-43	-67	-163

Discriminant $D = -D_0 \cdot f^2$ pour $\deg(H_D) = 1$

$-D_0$	-3	-3	-3	-4	-4	-4	-7	-8	-8	-11	-15	-15
f	4	5	7	3	4	5	4	2	3	3	1	2
D	-48	-75	-147	-36	-64	-100	-112	-32	-72	-99	-15	-60

$-D_0$	-20	-24	-35	-40	-51	-52	-88	-91	-115	-123	-148	-187
f	1	1	1	1	1	1	1	1	1	1	1	1
D	-20	-24	-35	-40	-51	-52	-88	-91	-115	-123	-148	-187

$-D_0$	-232	-235	-267	-403	-427
f	1	1	1	1	1
D	-232	-235	-267	-403	-427

Discriminant $D = -D_0 \cdot f^2$ pour $\deg(H_D) = 2$

Chaque discriminant caractérise un ordre qui est l'anneau d'endomorphismes d'une (ou deux) courbe(s) E. On peut trouver ces courbes en calculant les racines du polynôme de classes de Hilbert. En pratique, on utilise la fonction suivante sous Sage :

```
sage.schemes.elliptic_curves.cm.hilbert_class_polynomial(D,algorithm=None)
```

Par exemple, pour le discriminant $D = -88$, l'ordre correspondant est $\mathcal{O} = \mathbb{Z}[\sqrt{-3}]$. On calcule le polynôme de classes de Hilbert :

$$\begin{aligned} H_D(X) &= X^2 - 6294842640000 \cdot X + 15798135578688000000 \\ &= (X - 216000(14571395 + 10303524\sqrt{2}))(X - 216000(14571395 - 10303524\sqrt{2})) \end{aligned}$$

La factorisation de H_D donne les j-invariants $216000(14571395 \pm 10303524\sqrt{2})$.

En raisonnant ainsi, on retrouve sur Sage les tables 1 et 2 de l'article [4] de Benjamin

Smith. On pourra consulter l'annexe C pour une implémentation en Sage.

$-D_0 \cdot f^2$	j-invariant
$-3 \cdot 1^2$	0
$-3 \cdot 2^2$	$2^4 \cdot 3^3 \cdot 5^3$
$-3 \cdot 3^2$	$-2^{15} \cdot 3 \cdot 5^3$
$-4 \cdot 1^2$	$2^6 \cdot 3^3$
$-4 \cdot 2^2$	$2^3 \cdot 3^3 \cdot 11^3$
$-7 \cdot 1^2$	$-3^3 \cdot 5^3$
$-7 \cdot 2^2$	$3^3 \cdot 5^3 \cdot 17^3$
$-8 \cdot 1^2$	$2^6 \cdot 5^3$
$-11 \cdot 1^2$	-2^{15}
$-19 \cdot 1^2$	$-2^{15} \cdot 3^3$
$-43 \cdot 1^2$	$-2^{18} \cdot 3^3 \cdot 5^3$
$-67 \cdot 1^2$	$-2^{15} \cdot 3^3 \cdot 5^3 \cdot 11^3$
$-163 \cdot 1^2$	$-2^{18} \cdot 3^3 \cdot 5^3 \cdot 23^3 \cdot 29^3$

$-D_0 \cdot f^2$	j-invariant
$-3 \cdot 4^2$	$40500(35010 \pm 20213\sqrt{3})$
$-3 \cdot 5^2$	$884736(-369830 \pm 165393\sqrt{5})$
$-3 \cdot 7^2$	$331776000(-52518123 \pm 11460394\sqrt{21})$
$-4 \cdot 3^2$	$192(399849 \pm 230888\sqrt{3})$
$-4 \cdot 4^2$	$54(761354780 \pm 538359129\sqrt{2})$
$-4 \cdot 5^2$	$1728(12740595841 \pm 5697769392\sqrt{5})$
$-7 \cdot 4^2$	$3375(40728492440 \pm 15393923181\sqrt{7})$
$-8 \cdot 2^2$	$1000(26125 \pm 18473\sqrt{2})$
$-8 \cdot 3^2$	$8000(23604673 \pm 9636536\sqrt{6})$
$-11 \cdot 3^2$	$180224(-104359189 \pm 18166603\sqrt{33})$
$-15 \cdot 1^2$	$135/2(-1415 \pm 637\sqrt{5})$
$-15 \cdot 2^2$	$135/2(274207975 \pm 122629507\sqrt{5})$
$-20 \cdot 1^2$	$320(1975 \pm 884\sqrt{5})$
$-24 \cdot 1^2$	$1728(1399 \pm 988\sqrt{2})$
$-35 \cdot 1^2$	$163840(-360 \pm 161\sqrt{5})$

$-D_0 \cdot f^2$	j-invariant
$-40 \cdot 1^2$	$8640(24635 \pm 11016\sqrt{5})$
$-51 \cdot 1^2$	$442368(-6263 \pm 1519\sqrt{17})$
$-52 \cdot 1^2$	$216000(15965 \pm 4428\sqrt{13})$
$-88 \cdot 1^2$	$216000(14571395 \pm 10303524\sqrt{2})$
$-91 \cdot 1^2$	$884736(-5854330 \pm 1623699\sqrt{13})$
$-115 \cdot 1^2$	$4423680(-48360710 \pm 21627567\sqrt{5})$
$-123 \cdot 1^2$	$110592000(-6122264 \pm 956137\sqrt{41})$
$-148 \cdot 1^2$	$216000(91805981021 \pm 15092810460\sqrt{37})$
$-187 \cdot 1^2$	$940032000(-2417649815 \pm 586366209\sqrt{17})$
$-232 \cdot 1^2$	$216000(1399837865393267 \pm 259943365786104\sqrt{29})$
$-235 \cdot 1^2$	$5887918080(-69903946375 \pm 31261995198\sqrt{5})$
$-267 \cdot 1^2$	$55296000(-177979346192125 \pm 18865772964857\sqrt{89})$
$-403 \cdot 1^2$	$110592000(-11089461214325319155 \pm 3075663155809161078\sqrt{13})$
$-427 \cdot 1^2$	$147197952000(-53028779614147702 \pm 6789639488444631\sqrt{61})$

On connaît une expression du j-invariant des courbes de Smith de degré 2, 3, 5 et 7 :

$$j(\tilde{E}_{2,\Delta,s}) = \frac{-12^3(C_{2,\Delta}(s)-24)^3}{C_{2,\Delta}(s)^2 \cdot {}^{\sigma}C_{2,\Delta}(s)} \qquad j(\tilde{E}_{3,\Delta,s}) = \frac{2^8 \cdot 3^3 \cdot (2C_{3,\Delta}(s)+1)^3}{C_{3,\Delta}(s) \cdot {}^{\sigma}C_{3,\Delta}(s)}$$

$$j(\tilde{E}_{5,-1,s}) = \frac{-64(3(6s^2+6s-1)-20(s^2-s)\sqrt{-1})^3}{(1+s^2)(1+s\sqrt{-1})^4}$$

$$j(\tilde{E}_{7\Delta,s}) = \frac{(27+s^2\Delta)(85+96s\sqrt{\Delta}+15s^2\Delta)^3}{(1-s^2\Delta)(1-s\sqrt{\Delta})^6}$$

On résoud alors $j(\tilde{E}_{d,\Delta,s}) = j$ pour j dans les tables précédentes, avec $s \in \mathbb{Q}$ et $\Delta \in \mathbb{Z}$ sans facteur carré. On retrouve les résultats de [4, théorème 6] :

Degré 2

$s\sqrt{\Delta}$	Disc(End(E))
$\frac{5}{9}\sqrt{-7}$	$-7 \cdot 1^2$
0	$-8 \cdot 1^2$
$\frac{7}{12}\sqrt{3}$	$-4 \cdot 3^2$
$\frac{161}{360}\sqrt{5}$	$-4 \cdot 5^2$
$\frac{20}{49}\sqrt{6}$	$-8 \cdot 3^2$
$\frac{1}{2}\sqrt{5}$	$-20 \cdot 1^2$
$\frac{2}{3}\sqrt{2}$	$-24 \cdot 1^2$
$\frac{4}{9}\sqrt{5}$	$-40 \cdot 1^2$
$\frac{5}{18}\sqrt{13}$	$-52 \cdot 1^2$
$\frac{70}{99}\sqrt{2}$	$-88 \cdot 1^2$
$\frac{145}{882}\sqrt{37}$	$-148 \cdot 1^2$
$\frac{1820}{9801}\sqrt{29}$	$-232 \cdot 1^2$

Degré 3

$s\sqrt{\Delta}$	Disc(End(E))
0	$-3 \cdot 2^2$
$\frac{5}{2}\sqrt{-2}$	$-8 \cdot 1^2$
$\frac{1}{4}\sqrt{-11}$	$-11 \cdot 1^2$
$\frac{5}{9}\sqrt{3}$	$-3 \cdot 4^2$
$\frac{9}{20}\sqrt{5}$	$-3 \cdot 5^2$
$\frac{55}{252}\sqrt{21}$	$-3 \cdot 7^2$
$1\sqrt{5}$	$-15 \cdot 1^2$
$\frac{11}{25}\sqrt{5}$	$-15 \cdot 2^2$
$\frac{1}{2}\sqrt{2}$	$-24 \cdot 1^2$
$\frac{1}{4}\sqrt{17}$	$-51 \cdot 1^2$
$\frac{5}{32}\sqrt{41}$	$-123 \cdot 1^2$
$\frac{53}{500}\sqrt{89}$	$-267 \cdot 1^2$

Degré 7

$s\sqrt{\Delta}$	Disc(End(E))
$3\sqrt{-3}$	$-3 \cdot 1^2$
$\frac{5}{3}\sqrt{-3}$	$-3 \cdot 2^2$
$\frac{1}{5}\sqrt{-3}$	$-3 \cdot 3^2$
0	$-7 \cdot 2^2$
$\frac{1}{3}\sqrt{-19}$	$-19 \cdot 1^2$
$\frac{1}{3}\sqrt{7}$	$-7 \cdot 4^2$
$1\sqrt{5}$	$-35 \cdot 1^2$
$\frac{1}{3}\sqrt{13}$	$-91 \cdot 1^2$
$\frac{5}{39}\sqrt{61}$	$-427 \cdot 1^2$

En degré 5, on doit fixer Δ pour obtenir une famille de courbes. On choisit ici $\Delta = -1$ [4, §7]. On obtient seulement deux courbes pour $s = 1$ et $-9/13$, de j-invariant 66^3, et d'ordre de discriminant $-4 \cdot 2^2$.

Remarque 28. En faisant tendre $s \to \infty$, on obtient $j(E_{2,\Delta,\infty}) = 1728$, $j(E_{3,\Delta,\infty}) = 0$ et $j(E_{7,\Delta,\infty}) = -3375$.

2.4 Recherche exhaustive

On dispose donc d'une liste finie de $\mathbb{Q}$-courbes CM. Il reste à les réduire modulo un nombre premier p pour obtenir une courbe sur $\mathbb{F}_{p^2}$. Ce nombre premier p doit vérifier plusieurs conditions :

2.4.1 Taille du nombre premier

Pour obtenir une sécurité de 256 bits, il faut utiliser un sous-groupe de la courbe avec un cardinal de 512 bits. En effet, la meilleure attaque générique sur le logarithme discret sur les courbes elliptiques est en $\mathcal{O}(\sqrt{\#G})$. Le théorème de Hasse montre que $\#E(\mathbb{F}_q) \simeq q$. On choisit donc p de 256 bits pour que $\#E(\mathbb{F}_{p^2}) = \mathcal{O}(p^2)$ (qui fait donc 512 bits).

2.4.2 Arithmétique efficace

On souhaite utiliser une arithmétique efficace sur $\mathbb{F}_{p^2}$. Certains nombres premiers sont plus intéressants que d'autres. Nous avons testé les nombres premiers de la forme

$$2^{256 \pm k} \pm \epsilon \qquad 0 \leq k \leq 8 \quad -2^{12} \leq \epsilon \leq 2^{12}$$

ainsi que les nombres premiers creux

$$2^{256} \pm 2^{256-s} \pm \ldots \qquad s = 30, 31, 32, 62, 63, 64$$

Nous avons porté une attention particulière aux trois nombres premiers suivants :

$$2^{255} - 19, \quad 2^{256} + 2^{96} - 1, \quad 2^{256} + 2^{252} - 1$$

2.4.3 Conditions de réduction de la $\mathbb{Q}$-courbe

Pour ces nombres premiers p, il faut vérifier les conditions du §1.2. Il faut donc que Δ soit non-carré modulo p, et que p ne divise pas le discriminant de $\tilde{E}$.[8] On souhaite obtenir une courbe ordinaire (notamment pour appliquer le corollaire 14). Les courbes super-singulières vérifient le critère $p \mid t_E$. La borne de Hasse donne $|t_E| \leq 2p$, si bien que

$$t_E \in \{-2p, -p, 0, p, 2p\}$$

En fait, on peut montrer que pour une courbe super-singulière, $r = 0$ et donc en utilisant le théorème 9 ($dr^2 = 2p + \epsilon_p t_E$), $t_E = \pm 2p$. Les courbes super-singulières de trace $\pm 2p$ ont un cardinal $\#E(\mathbb{F}_{p^2}) = (p \pm 1)^2$. On peut vérifier ce critère en prenant un P quelconque et en vérifiant si $[p \pm 1]P = \mathcal{O}$.

Comme attendu en théorie, la moitié des courbes sont super-singulières, et on obtient finalement très peu de courbes candidates, et il reste à vérifier que les courbes ont un cardinal intéressant.

2.4.4 Cardinal de la courbe obtenue

Pour chaque nombre premier p, on obtient donc une courbe elliptique définie sur $\mathbb{F}_{p^2}$ qui possède deux endomorphismes pour appliquer GLV en dimension 4, et une arithmétique efficace sur le corps fini $\mathbb{F}_{p^2}$. Il reste à vérifier que le cardinal de la courbe donne 256 bits de sécurité. On calcule donc le cardinal $\#E(\mathbb{F}_{p^2})$, puis on souhaiterait connaître ses plus gros facteurs premiers. Factoriser un nombre de 512 bits est assez coûteux, et on peut se passer de la factorisation complète. On commence par rechercher les petits facteurs premiers grâce à `ECM` ou `Trial`. Si le facteur restant est composé, on considère que $\#E(\mathbb{F}_{p^2})$ n'a pas d'assez gros facteurs. S'il est premier, et qu'il n'est pas trop petit, on pourra utiliser la courbe pour des applications cryptographiques.

8. La condition $p \wedge d = 1$ n'est pas vérifiée car elle est évidente avec p de cette taille.

3 La courbe $4\mathbb{Q}$-255-19

3.1 Définition et cardinal

Notre courbe est définie sur $\mathbb{F}_{p^2}$ avec $p = 2^{255} - 19$. L'arithmétique sur ce corps fini peut être implémentée de manière efficace en bénéficiant de la structure spécifique de p. Ce corps est celui utilisé par Bernstein pour sa courbe Ed25519 dans [12, §3].

En utilisant la construction $\mathbb{F}_{p^2} = \mathbb{F}_p[X]/(X^2 - 2) = \mathbb{F}_p(a)$, on obtient l'équation de la courbe :

$$E(\mathbb{F}_{p^2}) : y^2 = x^3 + \underbrace{\left(-30 + \frac{140}{11} \cdot \sqrt{2}\right)}_{A} x + \underbrace{\left(56 - \frac{560}{11} \cdot \sqrt{2}\right)}_{B}$$

$A = 368429374846006076347725861391279706805859042117947249216455949116087230672 53a - 30$
$B = 263163839175718625962661329565199790575613601512819463726039963654348021908 35a + 56$
$j(E) = -2225561184000a + 3147421320000.$
La courbe est construite à partir de la $\mathbb{Q}$-courbe $E_{2,2,70/99}$ et possède un anneau d'endomorphismes de discriminant -88 :

$$\mathrm{End}(E) = \mathbb{Z}[\sqrt{-22}]$$

La courbe possède un cardinal avec un facteur premier de 508 bits :

$$\#E(\mathbb{F}_{p^2}) = 4 \cdot N$$

$$\begin{aligned} N = \quad & 837987995621412318723376562387865382967460363787024 \\ & 586107722590232610251879073047955441365222409345448 \\ & 472682727742170061679779878946355915266474990239807 \end{aligned}$$

En utilisant ce groupe G d'ordre N, on obtient une sécurité de 254 bits : la meilleure attaque sur le logarithme discret est en $\mathcal{O}(\#G^{\frac{1}{2}}) = \mathcal{O}(2^{254})$. Cette courbe est donc intéressante d'un point de vue cryptographique.

Le cardinal de la forme $4 \cdot N$ permet aussi d'utiliser la courbe sous forme d'Edward twisté, de façon à obtenir une artihmétique plus efficace pour la loi de groupe de la courbe.

Le twist de la courbe a un cardinal qui possède un facteur de 288 bits. Cela n'est pas aussi bien qu'espéré, mais on obtient environ 15 bits de sécurité de plus qu'avec les courbes GLS.

On peut alors appliquer la méthode GLV en dimension 4. Explicitons les endomorphismes.

3.2 Endomorphismes pour GLV

En tant que $\mathbb{Q}$-courbe de degré 2, la courbe possède un premier endomorphisme $\psi = [\sqrt{2}]$ dont l'expression est donnée en section 1.4 :

$$\psi : (x, y) \mapsto \left(\left(-\frac{x}{2} - \frac{C_{2,2}(70/99)}{x - 4}\right)^p, \left(\frac{y}{\sqrt{-2}}\left(\frac{-1}{2} + \frac{C_{2,2}(70/99)}{(x-4)^2}\right)\right)^p\right)$$

Le calcul de ψ n'est pas très coûteux car il fait intervenir des polynômes de degré au plus 2, ainsi qu'une élévation à la puissance p qui n'est autre que la conjugaison dans $\mathbb{F}_{p^2}$: $(u + va)^p = u - va$.

La courbe possède un second endomorphisme $\Psi = [\sqrt{-22}]$ de par son caractère CM, que l'on peut expliciter grâce à l'algorithme de Stark (annexe D) en Sage. Cet algorithme utilise la fonction de Weierstrass $\wp(z)$. A l'aide d'un développement en fractions continues, on exprime $\wp(\sqrt{-22}z)$ comme une fraction rationnelle en $\wp(z)$. On a alors, pour $P = (\wp(z), \wp'(z))$,

$$x_{\sqrt{-22}P} = \wp(\sqrt{-22}z) = \frac{f(\wp(z))}{g(\wp(z))} = \frac{f(x_P)}{g(x_P)}$$

$$= \frac{(\begin{smallmatrix}37613158894081478430028537853878989032120413485714578956837852875767327926994a\\ +500623502701805065866078605900951470307083690691490361368996962691889586286 46)\, x_P^{22} + \ldots}{(\begin{smallmatrix}40951173610078940216154554779821550192875788306583493245499116792467257905367a\\ +56549186429190809330336917104785843857115727135126845382782522156974206568768)\, x_P^{21} + \ldots}$$

Les détails de cet algorithme sont donnés dans [11, page 157]. Concernant le calcul effectif de Ψ, le théorème 27 montre que les polynômes sont de degré $N(-22) = 22$, et on doit aussi faire un calcul de dérivée pour obtenir l'ordonnée du point. Malgré des optimisations comme l'utilisation de la méthode de Horner pour évaluer les polynômes, l'endomorphisme $\sqrt{-22}$ coûte cher et on souhaiterait trouver un endomorphisme moins coûteux.

On va donc chercher un autre endomorphisme. L'idée de GLV en dimension 4 est de décomposer l'endomorphisme dans la base $(1, \sqrt{2}, \sqrt{-22}, \sqrt{2}\sqrt{-22})$. On voit ici un troisième endomorphisme qui montre qu'on peut expliciter $\sqrt{-11}$. Aurore Guillevic suggère l'idée de reproduire le raisonnement effectué en section 1.4 :
Grâce au polynôme de division P_{11}, on peut générer le groupe de 11-torsion

$$E[11] \simeq \mathbb{Z}/11\mathbb{Z} \times \mathbb{Z}/11\mathbb{Z}$$

d'ordre 121. Ce polynôme est de degré $(11^2 - 1)/2 = 60$ et se factorise sur $\mathbb{F}_{p^2}$ en un polynôme de degré 5, et un de degré 55. Le premier facteur irréductible de P_{11} permet de générer un sous-groupe d'ordre 11. On obtient un sous-groupe d'ordre 11 de $E[11]$. On peut alors utiliser les formules de Vélu pour expliciter une isogénie de degré 11.

$$\begin{array}{rccl} f: & E & \longrightarrow & E/G \\ & (x,y) & \longmapsto & \left(\frac{x^{11}+34ax^{10}+\ldots+(309\cdots831a+870\cdots644)}{x^{10}+34ax^9+\ldots+(159\cdots154a+164\cdots276)}, y \cdot \frac{x^{15}+\ldots+(157\cdots606a+254\cdots294)}{x^{15}+\ldots+(484\cdots274a+373\cdots111)}\right) \end{array}$$

E/G est une courbe isomorphe à ${}^{(p)}E$. On note g cet isomorphisme.

$$g : E/G \longrightarrow {}^{(p)}E$$

g est de la forme $(x, y) \mapsto (u^2x, u^3y)$ avec $u = \sqrt[4]{A_{E/G}/A_E} = \sqrt[6]{B_{E/G}/B_E}$.

$$u = 408474049982952188581701060660851528383726120587686671784543249580648409247 89a$$

Enfin, on utilise le Frobenius

$$\pi_p : {}^{(p)}E \longrightarrow E$$

pour obtenir l'endomorphisme $[\sqrt{-11}]$ recherché :

$$\begin{array}{rccl} [\sqrt{-11}]: & E & \xrightarrow{f} E/G \xrightarrow{g} {}^{(p)}E \xrightarrow{\pi_p} E & \\ & (x,y) & \mapsto \left(\left(u^2 \frac{x^{11}+34ax^{10}+(280a+1706)x^9+\ldots}{x^{10}+34ax^9+\ldots}\right)^p, \left(u^3 y \cdot \frac{x^{15}+51ax^{14}+\ldots}{x^{15}+51ax^{14}+\ldots}\right)^p\right) & \end{array}$$

On pourra consulter l'annexe E pour une implémentation en Sage.

3.3 Décomposition du scalaire

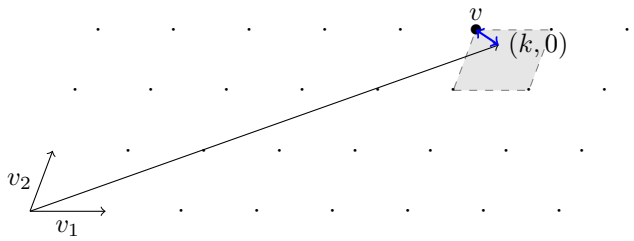
3.3.1 Dimension 2

La méthode GLV en dimension 2 permet de calculer $[k]P = [k_1]P + [k_2]\psi(P)$ avec k_1, k_2 de taille $\log(N)/2$, c'est-à-dire $\|(k_1, k_2)\|_\infty \simeq \sqrt{N}$.
Pour cela, on décompose $k \equiv k_1 + \lambda k_2 \mod N$ où λ est la valeur propre associée à l'endomorphisme ψ. L'idée est la suivante :

1. $\tilde{v}_1 = (-\lambda, 1)$ et $\tilde{v}_2 = (N, 0)$ vérifient $v_x + \lambda v_y \equiv 0 \mod N$.
   Ils forment une base d'un réseau d'éléments qui vérifient tous $v_x + \lambda v_y \equiv 0 \mod N$. En appliquant LLL sur ces deux vecteurs, on obtient une base courte (v_1, v_2) du réseau : $\|v_i\|_\infty \simeq \det(\mathcal{L})^{1/2} = \sqrt{N}$.
2. $(k_1, k_2) = (k, 0)$ est solution du problème, mais k_1 est beaucoup trop grand.
3. On trouve un élément $v = (v_x, v_y)$ du réseau qui est proche de $(k, 0)$. On utilise la méthode de Babaï : on décompose $(k, 0)$ dans la base (v_1, v_2) de $\mathbb{Q}^2$: $(k, 0) = \alpha_1 v_1 + \alpha_2 v_2$, puis on pose $v := a_1 v_1 + a_2 v_2$ où les a_i sont les entiers les plus proches des α_i.
4. On pose $(k_1, k_2) := (k, 0) - v$. Par linéarité,

$$k_1 + \lambda k_2 \equiv (k + \lambda \cdot 0) - (v_x + \lambda v_y) \equiv k - 0 \equiv k \mod N$$

   Comme (k_1, k_2) est dans le parallélogramme engendré par v_1 et v_2, on peut montrer que $|k_1|$ et $|k_2|$ sont majorés par $\max(\|v_1\|_\infty, \|v_2\|_\infty) \simeq \sqrt{N}$.



De façon un peu plus formelle, la décomposition en dimension 2 se déroule en 3 étapes :

1. Appliquer LLL sur $(\tilde{v}_1, \tilde{v}_2)$ et obtenir (v_1, v_2).
2. Décomposer $(k, 0)$ dans la base (v_1, v_2) de $\mathbb{Q}^2$: $(k, 0) = \alpha_1 v_1 + \alpha_2 v_2$.
3. Trouver l'élément du réseau le plus proche : $v = a_1 v_1 + a_2 v_2$. La solution est alors $(k_1, k_2) = (k, 0) - v$.

3.3.2 Dimension 4

Pour la décomposition en dimension 4, on utilise le même principe avec les trois étapes précédentes. Cette fois, on dispose de deux endomorphismes ψ et Ψ agissant comme λ et μ sur le groupe d'ordre N. On veut alors décomposer :

$$kP = k_1 P + k_2 \psi(P) + k_3 \Psi(P) + k_4 \psi \circ \Psi(P) = k_1 P + k_2 \lambda P + k_3 \mu P + k_4 \lambda \mu P$$

avec $\log(k_i) \simeq \log(N)/4$. On cherche donc à décomposer

$$k = k_1 + k_2 \lambda + k_3 \mu + k_4 \lambda \mu \mod N$$

1. *Recherche d'une base courte du réseau.*
   On pose

$$\begin{array}{rccl} f: & \mathbb{Z}^4 & \longrightarrow & \mathbb{Z}/N\mathbb{Z} \\ & (a_1,\ldots,a_4) & \longmapsto & a_1+\lambda a_2+\mu a_3+\lambda\mu a_4 \mod N \end{array}$$

   et on considère son noyau

$$\mathcal{L} := \ker f = \left\{(a_1,\ldots,a_4)\in\mathbb{Z}^4, a_1+\lambda a_2+\mu a_3+\lambda\mu a_4 \equiv 0 \mod N\right\}$$

   C'est un réseau de rang 4. Une base de $\mathcal{L}$ est donnée par

$$\langle (N,0,0,0),(-\lambda,1,0,0),(-\mu,0,1,0),(-\lambda\mu,0,0,1)\rangle$$

   En appliquant l'algorithme LLL sur ces quatre vecteurs, on obtient une base réduite de $\mathcal{L}$ que l'on note (v_1,v_2,v_3,v_4) :

$$\begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & v_3 & v_4 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} = \mathrm{LLL}\begin{pmatrix} N & -L & -M & -LM \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

   On a alors $\|v_i\|_\infty \simeq \det(\mathcal{L})^{1/4} = \sqrt[4]{N}$.
2. *Décomposition dans $\mathbb{Q}^4$.*
   Comme en dimension 2, on décompose $(k,0,0,0)$ dans la $\mathbb{Q}$-base (v_1,v_2,v_3,v_4) :

$$(k,0,0,0) = \sum_{i=1}^{4} \alpha_i v_i \qquad \alpha_i \in \mathbb{Q}$$

   Cela correspond simplement à une inversion de matrice :

$$\mathbb{Q}^4 \ni \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & v_3 & v_4 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}^{-1} \begin{pmatrix} k \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

   Cette inversion peut être précalculée car la matrice ne dépend pas de k.
3. *Recherche d'un vecteur proche dans le réseau.*
   On détermine les entiers a_i les plus proches des rationnels α_i :

$$a_1 = \lfloor \alpha_1 \rceil \qquad a_2 = \lfloor \alpha_2 \rceil \qquad a_3 = \lfloor \alpha_3 \rceil \qquad a_4 = \lfloor \alpha_4 \rceil$$

   Le vecteur $v := \sum_{i=1}^{4} a_i v_i$ est un vecteur du réseau $\mathcal{L}$, proche de $(k,0,0,0)$.
   On pose enfin

$$(k_1,k_2,k_3,k_4) := (k,0,0,0) - v = \sum_{i=1}^{4} (\alpha_i - \lfloor \alpha_i \rceil) v_i$$

   Comme $|x - \lfloor x \rceil| \leq 1/2$ pour $x \in \mathbb{Q}$, on obtient

$$\|(k_1,\ldots,k_4)\|_\infty \leq \sum_{i=1}^{4} \frac{\|v_i\|_\infty}{2} \leq \frac{4}{2} \max_i(\|v_i\|_\infty) \simeq 2\sqrt[4]{N}$$

   et les k_i sont de taille environ $\log(N)/4 + 1$.

Remarques 29.

- Les paramètres de l'algorithme LLL ne dépendent pas de k. On peut donc calculer la base courte en amont sur Sage.
- Pour le calcul d'inverse matricielle, on utilise la formule

$$M^{-1} := \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ v_1 & v_2 & v_3 & v_4 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}^{-1} = \frac{1}{\det(M)} {}^t\mathrm{Com}(M)$$

  Comme on applique la matrice sur le vecteur ${}^t(k,0,0,0)$, on ne calcule que la première ligne. LLL ne modifie pas le déterminant (au signe près) et donc $\det(M) = \pm N$. Finalement, il faut simplement calculer la première colonne de $\mathrm{Com}(M)$.
- L'entier le plus proche de $\frac{n}{d}$ est simplement $\left\lfloor \frac{2n+d}{2d} \right\rfloor$.

3.4 Résultats expérimentaux

L'addition sur la courbe elliptique est assez coûteuse. Celle-ci peut être implémentée de manière plus ou moins efficace suivant le modèle utilisé. On utilise ici l'addition dans le modèle projectif (sans inversions). On va estimer les coût de nos algorithmes en nombres d'opérations sur $\mathbb{F}_{p^2}$. On utilise les notations suivantes :

a Nombre d'opération pour calculer $u + v$ sur $\mathbb{F}_{p^2}$

m Nombre d'opération pour calculer $u \cdot v$ sur $\mathbb{F}_{p^2}$

inv Nombre d'opération pour calculer $1/u$ sur $\mathbb{F}_{p^2}$.

On dispose de quatre fonctions dont on connaît le coût :

A Addition de points $P + Q$: $\mathbf{A} = 14 \cdot \mathbf{m} + 7 \cdot \mathbf{a}$

$\mathbf{A}_n$ Addition $P \oplus_n Q$ (Addn) avec P quelconque, Q normalisé : $\mathbf{A}_n = 11 \cdot \mathbf{m} + 7 \cdot \mathbf{a}$

D Doublement de point $[2]P$: $\mathbf{D} = 12 \cdot \mathbf{m} + 18 \cdot \mathbf{a}$

N Normalisation de point : $\mathbf{N} = 1 \cdot \mathbf{inv} + 2 \cdot \mathbf{m}$.

On va comparer plusieurs algorithmes, dans lesquels le parcours des bits est toujours MSB-LSB (du plus fort au plus faible).

- Le double-and-add (**DAA**) est l'analogue du square-and-multiply pour un groupe additif.

```
DAA(k,P) :
  R = O
  for b bit de k :
    R = [2]R
    if b == 1 :
      R = R + P
  return R
```

- Le double-and-add normalisé (**DAAn**) utilise des Addn au lieu des Add durant le parcours des bits de k, moyennant la normalisation du point P en précalcul. Un Addn permet d'additionner un point quelconque avec un point normalisé. Cela coûte moins cher qu'un Add.

```
DAAn(k,P) :
  R = O
  P = Norm(P)
  for b bit de k :
    R = [2]R
    if b == 1 :
      R = R +n P
  return R
```

- Le GLV en dimension 2 (**GLV2**) commence par une décomposition de k en (k_1, k_2). Pour calculer ensuite $k_1P+k_2\psi(P)$, on précalcule $\psi(P)$ et $P+\psi(P)$, puis on parcourt simultanément les bits de k_1 et de k_2 :

```
GLV2(k,P):
  R = O
  Prec = [[O,psi(P)], [P,P+Q]]
  for i from len(k1,k2)-1 to 0 :
    R = [2]R
    R = R + Prec[k1[i],k2[i]]
  return R
```

- Le GLV en dimension 2 fenêtré (**GLV2-w2**) parcourt les bits de k_1 et k_2 en les regroupant par blocs de 2. On doit cependant précalculer

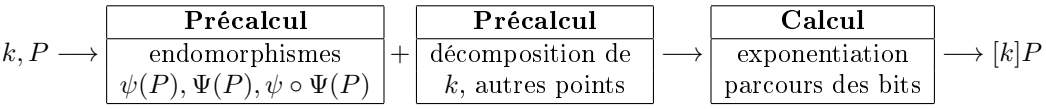
$$\{iP + j\psi(P), 0 \leq i, j \leq 3\}$$

  c'est-à-dire 13 points car O, P et $\psi(P)$ sont déjà calculés.
- Le GLV en dimension 4 (**GLV4-22**) calcule $k_1P + k_2\psi(P) + k_3\Psi(P) + k_4\psi\Psi(P)$ de la même manière que **GLV2**. On doit cependant précalculer

$$\{iP + j\psi(P) + k\Psi(P) + l\psi \circ \Psi(P), 0 \leq i, j, k, l \leq 1\}$$

  c'est-à-dire 11 points car O, P, $\psi(P)$, $\Psi(P)$ et $\psi \circ \Psi(P)$ sont déjà calculés.
- Le GLV en dimension 4 avec $\sqrt{-11}$ (**GLV4-11**) est l'analogue de **GLV4-22** avec un endomorphisme moins coûteux.

Nous allons décomposer les algorithmes en plusieurs parties :



3.4.1 Coût des endomorphismes

Pour $\psi(P)$, on doit utiliser un point normalisé puis évaluer l'endomorphisme ψ. Au total, $\mathbf{N} + 2 \cdot \mathbf{inv} + 10 \cdot \mathbf{m} + 13 \cdot \mathbf{a} = 3 \cdot \mathbf{inv} + 12 \cdot \mathbf{m} + 13 \cdot \mathbf{a}$. Le point obtenu est lui aussi normalisé.

Pour $\Psi_{22}(P)$, cet endomorphisme coûte plus cher : $1 \cdot \mathbf{inv} + 95 \cdot \mathbf{m} + 89 \cdot \mathbf{a}$. On doit aussi utiliser un point normalisé, mais le calcul de Ψ est toujours accompagné de celui de ψ. La normalisation n'est donc calculée qu'une seule fois pendant la méthode **GLV4**.

Pour $\psi \circ \Psi(P)$, on évalue ψ sur $\Psi(P)$ qui a été calculé précédemment. Le coût est donc de $2 \cdot \mathbf{inv} + 10 \cdot \mathbf{m} + 13 \cdot \mathbf{a}$. Il n'y a pas de normalisation puisque $\Psi(P)$ est lui-aussi déjà sous forme normalisée.

Pour $\Psi_{11}(P)$, ce sont les même calculs que pour Ψ_{22}, mais l'endomorphisme coûte environ deux fois moins cher : $1 \cdot \mathbf{inv} + 42 \cdot \mathbf{m} + 33 \cdot \mathbf{a}$.

3.4.2 Coût des autres précalculs

Le double-and-add ne nécessite aucun précalcul. La version normalisée précalcule le point P sous forme normalisée : $\mathbf{N} = 1 \cdot \mathbf{inv} + 2 \cdot \mathbf{m}$.

Le GLV en dimension 2 commence par une décomposition de k en (k_1, k_2) : $2 \cdot \mathbf{inv} + 6 \cdot \mathbf{m} + 8 \cdot \mathbf{a}$. On doit ensuite précalculer $\{\mathcal{O}, P, \psi(P), P + \psi(P)\}$, mais les trois premiers sont déjà calculés. Il reste donc une addition de points (avec Addn car $\psi(P)$ est sous forme normalisée) et une normalisation pour l'exponentiation : $\mathbf{A}_n + \mathbf{N}$. Au total, $3 \cdot \mathbf{inv} + 19 \cdot \mathbf{m} + 15 \cdot \mathbf{a}$.

L'utilisation d'une fenêtre nécessite le précalcul de 13 points. En optimisant l'utilisation des Addn, Add, et doublements, on obtient : $3 \cdot \mathbf{D} + 7 \cdot \mathbf{A}_n + 3 \cdot \mathbf{A} = 155 \cdot \mathbf{m} + 124 \cdot \mathbf{a}$. On ajoute à cela la décomposition de k et on obtient finalement $2 \cdot \mathbf{inv} + 161 \cdot \mathbf{m} + 132 \cdot \mathbf{a}$.

En dimension 4, on doit précalculer 11 points. Le coût est de $10 \cdot \mathbf{A}_n + \mathbf{A} = 124 \cdot \mathbf{m} + 77 \cdot \mathbf{a}$. La décomposition de k en $(k_1, \ldots, k_4)$ coûte $4 \cdot \mathbf{inv} + 8 \cdot \mathbf{m} + 16 \cdot \mathbf{a}$. Finalement, on obtient $4 \cdot \mathbf{inv} + 132 \cdot \mathbf{m} + 93 \cdot \mathbf{a}$

3.4.3 Coût de l'exponentiation

Le double-and-add parcourt les bits de k et effectue pour chaque bit un doublement, et une addition avec probabilité $1/2$ suivant la valeur du bit. On obtient donc

$$\log(k) \cdot \mathbf{D} + \frac{\log(k)}{2} \cdot \mathbf{A} \text{ (ou } \mathbf{A}_n)$$

La méthode GLV en dimension 2 permet d'obtenir k_1, k_2 de taille $\log(k)/2$. Les précalculs permettent de faire une addition avec probabilité $3/4$ durant le parcours des bits. Finalement, on obtient

$$\frac{\log(k)}{2} \cdot \mathbf{D} + \frac{3\log(k)}{8} \cdot \mathbf{A}_n$$

L'utilisation d'une fenêtre de taille 2 permet de réduire le nombre d'addition à $\log(k)/4$. On ne peut pas toujours utiliser Addn car les précalculs ne sont pas tous normalisé, et on obtient donc

$$\frac{\log(k)}{2} \cdot \mathbf{D} + \frac{\log(k)}{4} \left(\frac{2}{16} \cdot \mathbf{A}_n + \frac{13}{16} \cdot \mathbf{A} \right)$$

Dans **GLV4**, $k_1, \ldots, k_4$ sont de taille $\log(k)/4$ et là non-plus, on ne peut pas toujours utiliser Addn. On a alors

$$\frac{\log(k)}{4} \cdot \mathbf{D} + \frac{\log(k)}{4} \left(\frac{4}{16} \cdot \mathbf{A}_n + \frac{11}{16} \cdot \mathbf{A} \right)$$

3.4.4 Coût total

Algo.	Endomorphismes	Autres précalculs	Exponentiation
DAA	0	0	$19\log(k)\mathbf{m}, 21.5\log(k)\mathbf{a}$
DAAn	0	$1\mathbf{inv}, 2\mathbf{m}$	$17.5\log(k)\mathbf{m}, 21.5\log(k)\mathbf{a}$
GLV2	$3\mathbf{inv}, 12\mathbf{m}, 13\mathbf{a}$	$3\mathbf{inv}, 19\mathbf{m}, 15\mathbf{a}$	$10.13\log(k)\mathbf{m}, 11.63\log(k)\mathbf{a}$
GLV2-w2	$3\mathbf{inv}, 12\mathbf{m}, 13\mathbf{a}$	$2\mathbf{inv}, 161\mathbf{m}, 132\mathbf{a}$	$9.19\log(k)\mathbf{m}, 10.64\log(k)\mathbf{a}$
GLV4-22	$6\mathbf{inv}, 117\mathbf{m}, 115\mathbf{a}$	$4\mathbf{inv}, 132\mathbf{m}, 93\mathbf{a}$	$6.09\log(k)\mathbf{m}, 6.14\log(k)\mathbf{a}$
GLV4-11	$6\mathbf{inv}, 64\mathbf{m}, 59\mathbf{a}$	$4\mathbf{inv}, 132\mathbf{m}, 93\mathbf{a}$	$6.09\log(k)\mathbf{m}, 6.14\log(k)\mathbf{a}$

On peut alors calculer le temps de calcul de chaques algorithme en pratique. Comme $k \in \mathbb{Z}/N\mathbb{Z}$, $\log(k) = 508$. On peut alors donner le nombre d'opérations sur $\mathbb{F}_{p^2}$ au total.

Algorithme	**Coût sur $\mathbb{F}_{p^2}$**	**Temps ($\times 500$)**
Double-and-add	$0 \cdot \mathbf{inv} + 9650 \cdot \mathbf{m} + 10920 \cdot \mathbf{a}$	2.88 secondes
Double-and-add normalisé	$1 \cdot \mathbf{inv} + 8890 \cdot \mathbf{m} + 10920 \cdot \mathbf{a}$	2.54 secondes
GLV en dimension 2	$4 \cdot \mathbf{inv} + 5170 \cdot \mathbf{m} + 5930 \cdot \mathbf{a}$	1.57 seconde
GLV en dimension 2 fenêtré	$5 \cdot \mathbf{inv} + 4840 \cdot \mathbf{m} + 5550 \cdot \mathbf{a}$	1.50 seconde
GLV en dimension 4 avec $\sqrt{-22}$	$10 \cdot \mathbf{inv} + 3340 \cdot \mathbf{m} + 3330 \cdot \mathbf{a}$	1.36 seconde
GLV en dimension 4 avec $\sqrt{-11}$	$10 \cdot \mathbf{inv} + 3290 \cdot \mathbf{m} + 3270 \cdot \mathbf{a}$	1.23 seconde

Les résultats au niveau des temps de calculs sont intéressants : on gagne 52% sur un double-and-add normalisé, et 18% sur un GLV en dimension 2.

Concernant le nombre d'opérations sur $\mathbb{F}_{p^2}$, le nombre d'inversions est un peu mystérieux. On aimerait exprimer **inv** en fonction de **m** et **a**. Une inversion modulaire revient à faire un algorithme d'Euclide étendu. On connaît la complexité de cet algorithme *en pire cas* : quadratique en la taille des nombres. Ici, on voudrait une complexité dans le cas moyen, et on calcule donc ce coût en pratique. On obtient environ $60 \cdot \mathbf{m} + 20 \cdot \mathbf{a}$.

Finalement, les inversions sont assez coûteuses, et on peut encore optimiser notre algorithme **GLV4** en utilisant des variantes de ψ et Ψ qui retournent des points sous forme projective. Cela réduit le nombre d'inversions, mais on utilise alors moins de Addn. Finalement, on obtient :

1. Décomposition de k en (k_1, k_2, k_3, k_4) : $4 \cdot \mathbf{inv} + 8 \cdot \mathbf{m} + 16 \cdot \mathbf{a}$.
2. Précalcul des endomorphismes sous forme projectif :

   Calcul de $\psi(P)$ en projectif On doit aussi faire la normalisation de P car il n'est pas sous forme normalisé, puis appliquer ψ. On obtient un coût de : $\mathbf{N} + 11 \cdot \mathbf{m} + 13 \cdot \mathbf{a} = 1 \cdot \mathbf{inv} + 13 \cdot \mathbf{m} + 13 \cdot \mathbf{a}$.

   Calcul de $\Psi(P)$ en projectif P a déjà été normalisé pour ψ. On supprime les inversions et on obtient : $52 \cdot \mathbf{m} + 45 \cdot \mathbf{a}$.

   Calcul de $\psi \circ \Psi(P)$ en projectif On doit aussi faire la normalisation de $\Psi(P)$ car il n'est pas sous forme normalisé, puis appliquer ψ. On obtient comme pour $\psi(P)$: $1 \cdot \mathbf{inv} + 13 \cdot \mathbf{m} + 13 \cdot \mathbf{a}$.

   Au total, $2 \cdot \mathbf{inv} + 78 \cdot \mathbf{m} + 71 \cdot \mathbf{a}$.
3. Précalculs des 11 autres points : $3 \cdot \mathbf{A}_n + 8 \cdot \mathbf{A} = 145 \cdot \mathbf{m} + 77 \cdot \mathbf{a}$.
4. Exponentiation simultanée :

$$\frac{\log(k)}{4} \cdot \mathbf{D} + \frac{\log(k)}{4} \left(\frac{1}{16} \cdot \mathbf{A}_n + \frac{14}{16} \cdot \mathbf{A} \right) = 6.235 \log(k) \cdot \mathbf{m} + 6.14 \log(k) \cdot \mathbf{a}$$

Au total, $6 \cdot \mathbf{inv} + (6.235 \log(k) + 231) \cdot \mathbf{m} + (6.14 \log(k) + 164) \cdot \mathbf{a}$.

Algorithme	**Coût sur $\mathbb{F}_{p^2}$**	**Temps ($\times 500$)**
Double-and-add normalisé	$1 \cdot \mathbf{inv} + 8890 \cdot \mathbf{m} + 10920 \cdot \mathbf{a}$	2.54 secondes
GLV en dimension 2 fenêtré	$5 \cdot \mathbf{inv} + 4840 \cdot \mathbf{m} + 5550 \cdot \mathbf{a}$	1.50 seconde
GLV en dimension 4 optimisé	$6 \cdot \mathbf{inv} + 3400 \cdot \mathbf{m} + 3280 \cdot \mathbf{a}$	1.14 seconde

La méthode GLV en dimension 4 nous permet d'obtenir :

- un **gain de 55%** sur le Double-and-add
- un **gain de 24%** sur le GLV en dimension 2.

Pour obtenir une multiplication scalaire encore plus optimisée, on pourra utiliser l'arithmétique des courbes sous forme d'Edward twisté (car $\#E(\mathbb{F}_{p^2}) = 4 \cdot N$), ainsi que l'implémentation de Bernstein de $\mathbb{F}_{p^2}$ ($p = 2^{255} - 19$). Ces améliorations n'ont pas été implémentées mais cela ne change pas l'analyse : la comparaison est identique dans la mesure où ces optimisations s'appliquent aussi sur le GLV en dimension 2 et sur le Double-and-add.

Conclusion

La méthode GLV en dimension 4 permet un gain de temps lors du calcul de $[k]P$ grâce à l'utilisation d'endomorphismes facilement calculables.

Dans le cas d'un algorithme utilisant un point P fixe, on peut gagner en efficacité sans utiliser les endomorphismes GLV :

- Précalculer $P_2 = [2^{n/4}]P$, $P_3 = [2^{n/2}]P$ et $P_4 = [2^{3n/4}]P$
- Décomposer $k = k_1 + k_2 \cdot 2^{n/4} + k_3 \cdot 2^{n/2} + k_4 \cdot 2^{3n/4}$ avec $\log(k_i) = \log(k)/4$
- Calculer $[k]P = [k_1]P + [k_2]P_2 + [k_3]P_3 + [k_4]P_4$.

Le gain de temps est alors similaire à GLV en dimension 4 car les k_i font la même taille que ceux de GLV en dimension 4. Le coût des précalculs est cependant très élevé, et si le point de base n'est pas fixe, cela coûte beaucoup trop cher. Parmi les protocoles utilisés actuellement, seule la signature utilise un point P fixe. On utilisera donc GLV en dimension 4 pour une vérification de signature ou un échange de clés.

Après avoir étudié des familles de $\mathbb{Q}$-courbes de l'article d'Hasegawa, nous obtenons une courbe permettant une sécurité de 254 bits. Cette courbe est munie de deux endomorphismes agissant comme $[\sqrt{2}]$ et $[\sqrt{-11}]$ sur $\mathbb{F}_{p^2}$. On peut alors appliquer la méthode GLV et obtenir un gain de temps de 55% par rapoprt à un Double-and-add, et 24% sur un GLV en dimension 2.

Le twist de notre courbe est moins sécurisé : le plus gros facteur a 288 bits, ce qui donne 144 bits de sécurité. Cela donne un peu plus de robustesse que les courbes GLS pour lesquelles le cardinal du twist a un facteur de la taille de p, c'est-à-dire seulement 128 bits de sécurité pour un p de 256 bits.

Une piste est alors de rechercher de nouvelles courbes. Pour cela, on souhaiterait comprendre les $\mathbb{Q}$-courbes d'Hasegawa de manière générale. Par exemple, on souhaite pouvoir retrouver les formules des $\mathbb{Q}$-courbes de degré 5 avec $\Delta \neq -1$ ou s'intéresser aux courbes de degré non-premier. On pourra aussi se restreindre aux courbes d'anneau d'endomorphismes $\mathrm{End}(E)$ de petit discriminant $-D_0 \cdot f^2$ pour obtenir un endomorphisme facilement calculable et ne pas se retrouver face au problème de $[\sqrt{-22}]$ auquel nous nous sommes confrontés.

Pour le calcul du cardinal des courbes, nous avons utilisé l'algorithme SEA. Ce calcul peut être accéléré en utilisant le caractère CM de la courbe : l'idéal engendré par π_p divise $p\mathbb{Z}[\pi_p]$. En factorisant (p), on peut calculer la trace t de π_p et en déduire $\#E(\mathbb{F}_{p^2}) = p^2+1-t$.

Il a été restitué dans ce rapport le choix d'une courbe particulière sur lequel une attention particulière a été portée. Le sujet reste toutefois vaste, et d'autres corps spécifiques sont encore à l'étude, ainsi que des optimisations suggérées dans les paragraphes précédents. Suite à des échanges avec B. Smith, cette étude fera l'objet d'une présentation au laboratoire d'informatique de l'Ecole Polytechnique. Ce stage sera également le point de départ d'une thèse CIFRE réalisée dans le laboratoire en partenariat avec le Loria (Nancy) où travaille A. Guillevic. Je profite de ces lignes pour la remercier pour son aide dans la recherche de calcul efficace de l'endomorphisme Ψ. Enfin, je remercie Olivier Bernard, Renaud Dubois et Olivier Orcière pour le temps accordé durant ces six mois.

Références

[1] C. Costello and P. Longa. Four$\mathbb{Q}$: four-dimensional decompositions on a $\mathbb{Q}$-curve over the Mersenne prime. *Advances in Cryptology – ASIACRYPT 2015, Lecture Notes in Computer Science vol. 9452, Springer Berlin Heidelberg*, pages 214–235, Dec 2015.

[2] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. *Advances in Cryptology — CRYPTO 2001 : 21st Annual International Cryptology Conference, Santa Barbara, California, USA*, 2001.

[3] S. D. Galbraith, X. Lin, and M. Scott. Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves. *Advances in Cryptology – EUROCRYPT 2009, Lecture Notes in Computer Science vol. 5479, Springer Berlin Heidelberg*, pages 518–535, 2009.

[4] B. Smith. The $\mathbb{Q}$-curve construction for endomorphism-accelerated elliptic curves. *Journal of Cryptology, Springer Verlag*, Oct 2016.

[5] P. Longa and F. Sica. Four-Dimensional Gallant-Lambert-Vanstone Scalar Multiplication. *Journal of Cryptology, Springer US*, pages 248–283, Apr 2014.

[6] J. González. Isogenies of polyquadratic $\mathbb{Q}$-curves to their Galois conjugates. *Archiv der Mathematik, vol. 77*, pages 383–390, Nov 2001.

[7] J. H. Silverman. The Arithmetic of Elliptic Curves. *vol. 106 of Graduate Text in Mathematics, Springer New York*, 1986.

[8] Y. Hasegawa. Q-curves over quadratic fields. *Manuscripta mathematica, vol. 94*, pages 347–364, Dec 1997.

[9] L. Defeo. Fast Algorithms for Towers of Finite Fields and Isogenies. *Ecole Polytechnique X*, 2010.

[10] D. A. Cox. Primes of the form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication. *John Wiley and Sons, Inc.*, 1989.

[11] Z. Kuijk. Modular functions of one variable I. *Springer Berlin Heidelberg*, 1973.

[12] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, pages 77–89, Sep 2012.

[13] J. Vélu. Isogénies entre courbes elliptiques. *C.R. Acad. Sc. Paris, Série A.*, page 238–241, 1971.

A Twist quadratique d'une courbe elliptique

Soit E une courbe elliptique définie sur $\mathbb{F}_q$ par $y^2 = x^3 + Ax + B$, avec $\#E(\mathbb{F}_q) = q + 1 - t$.

Proposition 30. $\#E(\mathbb{F}_{q^2}) = (q+1+t)(q+1-t)$.

Démonstration. Par les formules de Weil, on a $\#E(\mathbb{F}_{q^2}) = q^2 + 1 - (\alpha^2 + \bar{\alpha}^2)$ où $\alpha, \bar{\alpha}$ sont les deux solutions de $X^2 - tX + q = 0$. En calculant le discriminant du polynôme, on obtient $\Delta = t^2 - 4q$ et donc $\alpha = \frac{t+\sqrt{t^2-4q}}{2}$, $\bar{\alpha} = \frac{t-\sqrt{t^2-4q}}{2}$. Finalement, $\#E(\mathbb{F}_{q^2}) = q^2 + 1 - \frac{1}{4}(2t^2 + 2t^2 - 8q) = q^2 + 1 - t^2 + 2q = (q^2 + 2q + 1) - t^2 = (q+1)^2 - t^2$. □

Remarque 31. On peut, de manière similaire, calculer $\#E(\mathbb{F}_{q^3})$: $\alpha^3 + \bar{\alpha}^3 = \frac{1}{8}2t^3 + 6t(t^2 - 4q)) = t^3 - 3tq$ donc $\#E(\mathbb{F}_{q^3}) = q^3 + 1 - t^3 + 3tq^3$. De même, $\alpha^4 + \bar{\alpha}^4 = t^4 - 4t^2q + 2q^2$ donc $\#E(\mathbb{F}_{q^4}) = q^4 + 1 - t^4 + 4t^2q^4 - 2p^8$.

Définition 32 (twist quadratique)**.** *On définit le twist quadratique de E par $u \in \mathbb{F}_q - (\mathbb{F}_q)^2$ (ou encore par $\sqrt{u} \in \mathbb{F}_{p^2} - \mathbb{F}_p$) la courbe E' d'équation $y^2 = x^3 + u^2Ax + u^3B$.*

Remarque 33. E et E' sont définies sur $\mathbb{F}_q$, mais l'isomorphisme entre les deux est défini sur $\mathbb{F}_{p^2}$:

$$\begin{array}{rccc} \delta(\sqrt{u}) : & E & \longrightarrow & E' \\ & (x,y) & \longmapsto & (\sqrt{u}^2 x, \sqrt{u}^3 y) \end{array}$$

D'un point de vue géométrique, E et E' représentent les mêmes courbes. Pour construire E', on cherche u qui n'est pas un carré, puis on applique (à un changement de variables près) $(x,y) \mapsto (x, \sqrt{u}y)$.

Proposition 34. $\#E(\mathbb{F}_q) + \#E'(\mathbb{F}_q) = 2q + 2$.

Démonstration. Pour $x \in \mathbb{F}_q$, on considère $a = x^3 + Ax + B$. Si $a = 0$, $(x,0) \in E(\mathbb{F}_q) \cap E'(\mathbb{F}_q)$. Si $a \neq 0$, on a deux cas. Si a est un carré, il existe $y \in \mathbb{F}_q$ tel que $(x, \pm y) \in E$. Si a n'est pas un carré, on peut l'écrire uy^2 avec $y \in \mathbb{F}_q$, $u \in (\mathbb{F}_q)^2 - \mathbb{F}_q$. Dans ce cas, $(\sqrt{u}^2 x, \pm\sqrt{u}^3 y) \in E'$. Finalement, quelque soit la valeur de a, on obtient toujours deux points (soit un sur E et un sur E', soit deux sur E, soit deux sur E'). Finalement on a dénombré $2q$ points sur E et E'. Les deux derniers éléments $\mathcal{O}_E$ et $\mathcal{O}_{E'}$ permettent d'obtenir la formule. □

Remarque 35. On en déduit que $\#E'(\mathbb{F}_q) = q+1+t$, et que $\#E(\mathbb{F}_{q^2}) = \#E(\mathbb{F}_q) \times \#E'(\mathbb{F}_q)$.

Remarque 36. A partir d'un twist par un élément $u \in \mathbb{F}_q$ non-carré, on obtient un $\mathbb{F}_{q^2}$-isomorphisme entre $E(\mathbb{F}_q)$ et $E'(\mathbb{F}_q)$, donné par $\delta(\sqrt{u})$. L'isomorphisme inverse est $\delta(1/\sqrt{u})$. D'une isogénie f entre deux courbes, on obtient une isogénie f' entre les deux twists :

$$\begin{array}{ccc} E_1(\mathbb{F}_q) & \xleftarrow{\quad\delta(1/\sqrt{u})\quad} & E_1'(\mathbb{F}_q) \\ f \downarrow & & \downarrow f' \\ E_2(\mathbb{F}_q) & \xrightarrow[\quad\delta(\sqrt{v})\quad]{} & E_2'(\mathbb{F}_q) \end{array} \qquad \begin{array}{c} P \\ \downarrow \\ f'(P) = \delta(\sqrt{v}) \circ f \circ \delta(1/\sqrt{u})(P) \end{array}$$

Remarque 37. Le twist d'une courbe peut être utilisé pour casser le logarithme discret : lors d'une implémentation x-only, on peut faire une attaque par faute lors du calcul de $x^3 + Ax + B$ et calculer le logarithme discret sur le twist. On souhaite donc obtenir une courbe avec un cardinal du twist avec des gros facteurs.

B Isogénies et formules de Vélu

A partir d'une courbe elliptique E et d'un sous-groupe G de E, Vélu donne dans [13] les formules permettant d'expliciter une isogénie $\rho : E \longrightarrow E/G$ (de noyau G). Cette isogénie est définie de la manière suivante (où $G^* = G - \mathcal{O}$) :

$$\begin{aligned}\rho : \quad P \quad &\longmapsto \quad \left(x_P + \sum_{Q\in G^*}(x_{P+Q} - x_Q), y_P + \sum_{Q\in G^*}(y_{P+Q} - y_Q)\right) \\ &= \left(\sum_{Q\in G}(x_{P+Q} - x_Q), 1 + \sum_{Q\in G}(y_{P+Q} - y_Q)\right)\end{aligned}$$

Si E a pour équation $y^2 = x^3 + Ax + B$, on note $f(P) = f(x_P) \stackrel{\text{not}}{=} x_P^3 + Ax_P + B$ et on montre alors grâce aux formules d'addition (voir [9, page 122] pour plus de détails) que

$$\rho_1(x,y) = x + \sum_{Q\in G^*} \frac{f'(Q)}{x - x_Q} + \frac{2f(Q)}{(x-x_Q)^2} \qquad \rho_2(x,y) = y - \sum_{Q\in G^*} \frac{yf'(Q)}{(x-x_Q)^2} + \frac{4yf(Q)}{(x-x_Q)^3}$$

Le noyau de ρ est exactement G. On reconnaît que $\rho_2(P) = y\rho_1'(P)$. De plus, on peut écrire $\rho_1(P) = \frac{g(x)}{h(x)}$ avec $h(x) = \prod_{Q\in G^*}(x - x_Q)$: en effet, si $Q \in G^*$ est de 2-torsion, $f(Q) = 0$, et sinon, $f(Q)$ et $f(-Q)$ ont même abscisse, si bien qu'aucun terme en $\frac{1}{(x-x_Q)^2}$ n'apparaît. Pour résumer :

$$\rho(x,y) = \left(\frac{g(x)}{h(x)}, y\left(\frac{g(x)}{h(x)}\right)'\right)$$

On peut généraliser cette expression pour une isogénie quelconque entre deux courbes sous forme de Weierstrass, en utilisant le fait que les morphismes $(x,y) \mapsto (u^2x, u^3y)$ permettent de passer d'une courbe sous forme de Weierstrass à une autre. On obtient qu'une isogénie entre deux courbes elliptiques (sous forme de Weierstrass) peut toujours s'écrire

$$\varphi : (x,y) \longmapsto \left(\frac{g(x)}{h(x)}, \frac{y}{\Lambda(\varphi)}\left(\frac{g(x)}{h(x)}\right)'\right)$$

et on appelle $\Lambda(\varphi)$ le facteur de normalisation. L'isogénie de Vélu est toujours sous forme normalisée ($\Lambda(\varphi) = 1$).

Exemple d'isogénie de degré 2

Dans le cas d'un sous-groupe à deux éléments $G = \{\mathcal{O}, (\alpha, 0)\}$, les formules sont assez simples : $\rho_1(P) = x + x_{P+(\alpha,0)} - \alpha = x - \alpha + \left(\frac{y}{x-\alpha}\right)^2 - x - \alpha = -2\alpha + \frac{x^3+Ax+B}{(x-\alpha)^2}$. Comme $\alpha^3 + A\alpha + B = 0$, on peut réécrire $\rho_1(P) = x + \frac{3\alpha^2+A}{x-\alpha}$ et on a donc

$$\rho(x,y) = \left(x + \frac{3\alpha^2 + A}{x-\alpha}, y\left(1 - \frac{3\alpha^2+A}{(x-\alpha)^2}\right)\right)$$

On obtient les coefficients de E/G en calculant $\rho_2(x,y)^2$ et $\rho_1(x,y)^3$:

$$E/G : y^2 = x^3 + (-4A - 15\alpha^2)x + (B - 7\alpha(3\alpha^2 + A))$$

Pour les formules dans le cas d'un sous-groupe quelconque, on pourra lire [9, page 122].

C Génération des $\mathbb{Q}$-courbes sur Sage

Construction des tables 1 et 2 de [4]

```
KK.<x> = PolynomialRing(QQ)

def separate_square_factors(r) :
    p = r.numerator()
    q = r.denominator()
    s = 1
    if p*q>0 :
        D = 1
    else :
        D = -1
    for l in p.factor() :
        if l[1] >1:
            s = s * l[0]^(l[1]//2)
            D = D * l[0]^(l[1]%2)
        else :
            D = D * l[0]
    for m in q.factor() :
        if m[1]>1:
            s = s/(m[0]^(m[1]//2))
            D = D*(m[0]^(m[1]%2))
            s = s/(m[0]^(m[1]%2))
        else :
            s = s / m[0]
            D = D * m[0]
    return [abs(s),D]

def roots_d2(P) :
    if P.degree() >2 :
        print "too difficult to find the roots of ", P
    if P.degree() == -1 :
        print "Zero polynomial, every element is a root"
    if P.degree() == 0 :
        print "Non-zero constant polynomial, no root"
    if P.degree() == 1 :
        return [-P[0]/P[1]]
    if P.degree() == 2 :
        a = P[2]
        b = P[1]
        Delta = P.discriminant()
        [s,D] = separate_square_factors(Delta)
        K = NumberField(x^2 - D, 'r')
        return [(-b+s*K.gen())/(2*a),(-b-s*K.gen())/(2*a)]

def construct_CM_j_roots(list_D0, list_f) :
```

```
    T = []
    for i in range(len(list_D0)) :
        D = -list_D0[i]*list_f[i]^2
        T.append([list_D0[i],list_f[i],
        roots_d2(KK(sage.schemes.elliptic_curves
        .cm.hilbert_class_polynomial(D,algorithm=None)))])
    return T

list_D0_h1 = [3,3,3,4,4,7,7,8,11,19,43,67,163]
list_f_h1 = [1,2,3,1,2,1,2,1,1,1,1,1,1]
list_D0_h2 = [3,3,3,4,4,4,7,8,8,11,15,15,20,24,35,
40,51,52,88,91,115,123,148,187,232,235,267,403,427]
list_f_h2 = [4,5,7,3,4,5,4,2,3,3,1,2,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1]
```

Construction des tables de [4, théorème 6]

```
def get_hasegawa_j_inv(d) :
    Q0.<s> = PolynomialRing(QQ)
    Q1.<D> = PolynomialRing(Q0)
    Q2.<sqrtD> = PolynomialRing(Q1)
    if d == 5 :
        Q3.<sqrtD> = Q2.quo(sqrtD^2 +1)
    else :
        Q3.<sqrtD> = Q2.quo(sqrtD^2 - D)

    if d == 2 :
        C = 9*(1+s*sqrtD)
        sC = 9*(1-s*sqrtD)
        jnum = -12^3*(C-24)^3
        jden = (C^2*sC)

    if d == 3 :
        C = 2*(1+s*sqrtD)
        sC = 2*(1-s*sqrtD)
        jnum = 2^8*3^3*(2*C+1)^3
        jden = (C*sC^3)

    if d == 5 :
        jnum = -64*(3*(6*s^2+6*s-1)-20*(s^2-s)*sqrtD)^3
        jden = (1+s^2)*(1+s*sqrtD)^4

    if d == 7 :
        C = 7*(27+s^2*sqrtD^2)
        sC = C
        jnum = (27+s^2*sqrtD^2)*(85+96*s*sqrtD+15*s^2*sqrtD^2)^3
        jden = (1-s^2*sqrtD^2)*(1-s*sqrtD)^6
```

```
    return [Q3(jnum),Q3(jden)]

def get_hasegawa_coefficients(degree,s,Delta) :
    K.<d> = NumberField(x^2 - Delta)
    if degree == 2:
        C = 9*(1+s*d)
        A = 2*(C-24)
        B = -8*(C-16)
    if degree == 3 :
        C = 2*(1+s*d)
        A = -3*(2*C+1)
        B = C^2 + 10 * C - 2
    if degree == 5 :
        A = -27*s*(11*s - 2) * (3*(6*s^2+6*s-1) - 20 * s * (s-1) * d)
        B = 54*s^2*(11*s-2)^2 * ((11*s^2+59*s - 9)
        - 2 * (s-1) * (20*s+9) * d)
    if degree == 7:
        C = 7*(27+s^2*d^2)
        A = -3*C*(85+96*s*d+15*s^2*d^2)
        B = 14*C*(9*(3*s^4*d^4 + 130 *s^2*d^2 + 171)
        + 16*(9*s^2*d^2 + 163)*s*d)
    return [A,B,d]

def get_hasegawa_reduction_coefficients(p, d, s, Delta) :
    K.<x> = PolynomialRing(GF(p))
    if p % 4 == 3 :
        #we write sqrt(Delta) with sqrt(-1)
        #Creation of FF_{p^2}
        L.<X> = GF(p^2, modulus=x^2 + 1)
        #Expression of sqrt(Delta) with sqrt(-1)   (noted Dbar)
        S.<y> = PolynomialRing(L)
        Dbar = S(y^2 - Delta).roots()[1][0]
    if p % 4 == 1 :
        #Creation of FF_{p^2}
        if p == 2^255 - 19:
            L.<X> = GF(p^2, name='X', modulus=x^2-2)
        else :
            L.<X> = GF(p^2, name='X')
        #Expression of sqrt(Delta) with L.gen()   (noted Dbar)
        S.<y> = PolynomialRing(L)
        Dbar = S(y^2 - Delta).roots()[1][0]
    s = L(GF(p)(s.numerator())) * L(GF(p)(s.denominator()))^(-1)
    if d == 2 :
        C = 9*(1+s*Dbar)
        A = 2*(C-24)
        B = -8*(C-16)

    if d == 3 :
```

```
        C = 2*(1+s*Dbar)
        A = -3*(2*C+1)
        B = C^2 + 10 * C - 2

    if d == 5 :
        A = -27*s*(11*s - 2) * (3*(6*s^2+6*s-1)
        - 20 * s * (s-1) * Dbar)
        B = 54*s^2*(11*s-2)^2 * ((11*s^2+59*s - 9)
        - 2 * (s-1) * (20*s+9) * Dbar)

    if d == 7 :
        C = 7*(27+s^2*Dbar^2)
        A = -3*C*(85+96*s*Dbar+15*s^2*Dbar^2)
        B = 14*C*(9*(3*s^4*Dbar^4 + 130 *s^2*Dbar^2 + 171)
        + 16*(9*s^2*Dbar^2 + 163)*s*Dbar)

    return [A,B,L.modulus(),Dbar]

def get_Qfamily_equation_from_j(d,j0) :
    [jnum,jden] = get_hasegawa_j_inv(d)
    L.<r> = PolynomialRing(jnum.parent())
    if j0 in QQ :
        newj0 = j0[0]
    else :
        newj0 = j0[0] + r * j0[1]
    return L(newj0*jden - jnum)

def compute_table_thm(d, list_D0_h1, list_f_h1, list_D0_h2, list_f_h2) :
    T1 = construct_CM_j_roots(list_D0_h1, list_f_h1)
    T2 = construct_CM_j_roots(list_D0_h2, list_f_h2)
    table_thm = []
    for t in T1:
        Disc = - t[0] * t[1]^2
        Q = get_Qfamily_equation_from_j(d,t[2][0])
        #Q is in QQ[sqrtD][s][r] but degree 0 so in QQ[s*sqrtD]
        Q = Q[0]
        GCD = gcd(Q[0],Q[1])
        if GCD != 1:
            #GCD is in QQ[s][D]
            if GCD.degree() >1 :
                print 'Problem !', GCD
            if GCD.degree() == 1 :
                if GCD[1].degree()!= 2 or GCD[0].degree()>=1 :
                    print 'ALERT : GCD is not in the form a * s^2*D + b'
                else :
                    table_thm.append([t[0], t[1]]
                    + separate_square_factors(-GCD[0][0] / GCD[1][2]))
            else :
```

```
            if GCD.degree() == 0 :
                for s in GCD[0].roots():
                    if s[0] == 0 :
                        table_thm.append([t[0], t[1], s[0],0])
                    else :
                        table_thm.append([t[0], t[1], s[0],-1])
    for t in T2:
        Q = get_Qfamily_equation_from_j(d,t[2][0])

        #Recherche d'une solution avec r et sqrtD algébriquement
        #indépendants (aucune)
        GCD = gcd(gcd(Q[0][0],Q[0][1]),gcd(Q[1][0],Q[1][1]))
        if GCD != 1 :
            print 'there is a root to find !'
            print Disc ,';',GCD
        #Recherche d'une solution avec r et sqrtD algébriquement
        #dépendants, thm => r = sqrtD
        #On remplace sqrtD par r dans Q
        #puis on cherche les racines rationnelles
        r = t[2][0].parent().gen()
        R = (Q[0][0](D = r^2) + r*Q[0][1](D= r^2)) + r * (Q[1][0](D = r^2)
        + r*Q[1][1](D= r^2))
        for root in R.roots():
            if root[0] in QQ :
                table_thm.append([t[0], t[1], abs(root[0]), ZZ(r^2)])
    return table_thm
```

Construction des courbes de Smith

```
def DefineReductionQCurve(p,d,s,Delta) :
    [A,B,pol_gen,Dbar] =
    get_hasegawa_reduction_coefficients(p, d, s, Delta)
    if 4*A^3 + 27 *B^2 != 0 :
        return [EllipticCurve(A.parent(), [A,B]), A, B,pol_gen,Dbar]
    return ['singular curve', A, B,pol_gen,Dbar]

p = 2^255-19
d = 2
s = 70/99
D = 2
[E,A,B,pol_gen,Dbar] = DefineReductionQCurve(p, d, s, D)
```

D Algorithme de Stark

```
 def Stark(E,CM_number, ord) :
    K.<x> = (E.base_ring())[]
    a_i_result = []
    P = E.weierstrass_p(ord)
    LaurentSeriesRing = P.parent()

    #création de alpha0 = P(sqrt(CM_number)*z)
    alpha = LaurentSeriesRing(0)
    for i in range(-2, ord) :
        if i%2 == 0 :
            alpha = alpha + LaurentSeriesRing.gen()^i * P[i]
            * (CM_number)^(i/2)

    for i in range(abs(CM_number)) :
        soustrac = alpha - alpha[-2] * P
        if soustrac != 0 :
            a_i_result.append(K(alpha[-2] * x + soustrac[0]))
            soustrac = soustrac - soustrac[0]
        else :
            a_i_result.append(K(alpha[-2] * x))
        if soustrac != 0 :
            alpha = 1/soustrac
        else :
            return a_i_result
    return a_i_result


def list_to_pol(L) :
    P = 0
    for i in range(len(L)) :
        j = len(L) - 1 - i
        P = 1/(P +L[j])
    return 1/P

L = Stark(E, -22, 100)

F = list_to_pol(L)
```

E Endomorphisme $[\sqrt{-11}]$

```
#Définition de la courbe
p = 2^255 - 19
KK.<a> = GF(p^2, modulus=x^2-2)
s = 70/99
C = 9*(1+GF(p)(s)*a)
A4Q = 2*(C-24)
B4Q = -8*(C-16)
E4Q = EllipticCurve(KK, [A4Q,B4Q])

#Isogénie de Vélu
P_11 = E4Q.division_polynomial(11)
factor_P_11 = P_11.factor()[0][0]
f = EllipticCurveIsogeny(E4Q, factor_P_11)
#Image curve
A_f = 57896044618658097711785492504343953926634992332820282\
8201972879200395656481840 9*a+57896044618658097711785492504\
343953926634992332820282019728792003956564816319
B_f = 67760*a+74536
E_f = EllipticCurve(KK, [A_f, B_f])

print E_f.j_invariant() == E4Q.j_invariant().frobenius()

Esigma = EllipticCurve(KK, [A4Q^p, B4Q^p])

u2 = sqrt(A4Q^p/A_f)
if(u2^3 * B_f != B4Q^p):
u2 = -u2

u = sqrt(u2)
#maybe sqrt chose the bad root...

def isom(P):
    return Esigma(P[0] * u^2, P[1] * u^3)

def Frob(P):
    return E4Q(P[0]^p , P[1]^p)

P = E4Q.random_element()
print Frob(isom(f(P)))
```