Mémoire de fin d'étude

Recherche et implémentation de courbes elliptiques à couplages

Master *CRYPTIS*
Mention *Mathématiques*

Auteur
Azogagh Sofiane

Superviseur
Clarisse Rémi

30 août 2021

Table des matières

Introduction

Présentation de l'entreprise

Orange est une entreprise française de télécommunication. Avec un chiffre d'affaire en 2020 de 42 milliard d'euros, l'entreprise est leader ou second opérateur dans la plupart des pays où elle est implantée.

En 2010, Orange consacre $1,9\%$ de son chiffre d'affaire au financement de la recherche et du développement. Depuis janvier 2007, Orange a unifié ses laboratoires de recherches au sein du réseau Orange Labs. Le 21 mai 2021, Mickael Trabbia, directeur de la division Technology & Global Innovation (TGI) annonce le changement de nom d'Orange Labs pour Orange Innovation.

Ces laboratoires de recherches sont répartis sur plusieurs ville en France (Paris, Rennes, Caen, Lannion, *etc*) mais également à l'étranger (Londres, Madrid, Le Caire *etc*). Orange Innovation compte 8000 salariés dédiés à la recherche dont 62% en France et 38% hors de France.

Les défis numériques d'Orange Innovation d'aujourd'hui et de demain sont

1. Garantir une innovation durable, inclusive et responsable ;
2. Préparer les réseaux d'Orange aux ruptures technologiques à venir ;
3. Accompagner la transformation digitale en Afrique et Moyen-Orient ;
4. Imaginer les solutions IT et télécommunication du marchés B2B de demain ;
5. Développer de nouveaux services financiers ;
6. Placer la Data et l'IA au cœur de son modèle d'innovation.

Le département Sécurité d'Orange Innovation accompagne les projets du Groupe dans la prise en compte de la sécurité et de la gestion de leurs risques, leur fournit des référentiels, méthodes et outils et contribue activement à la recherche pour proposer des solutions innovantes de protection des réseaux et services d'Orange ainsi que les données des clients. Plusieurs métiers y sont représentés (voir Figure 1).

Présentation du stage

J'ai effectué mon stage au sein de l'équipe NCP (Network Cybersecurity and Privacy) qui travaille à la sécurisation des réseaux du futur. Cette équipe est composée de profils divers : développeurs et chercheurs couvrant de vastes domaines, du *machine learning* aux protocoles cryptographiques de protection de la vie privée. Mon stage s'inscrit dans

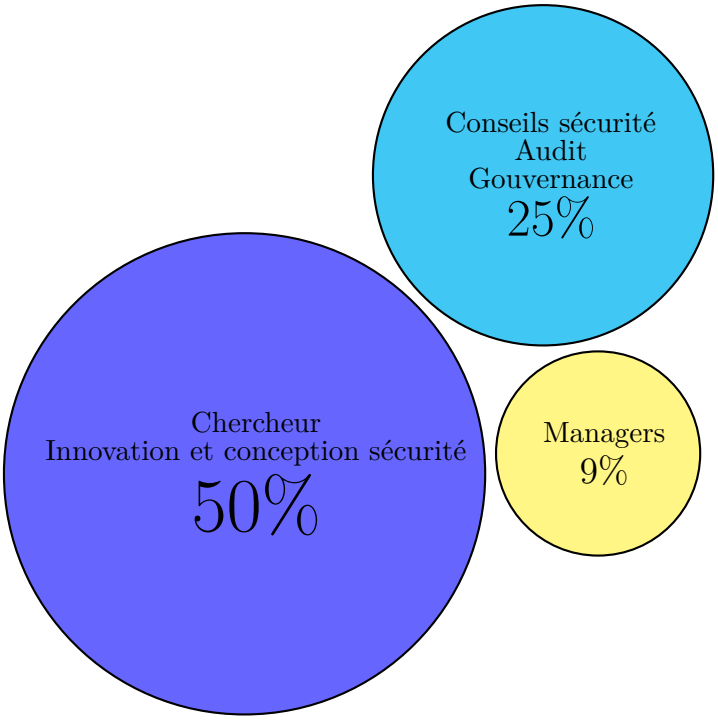

FIGURE 1 – Les différents métiers représentés au sein du département Sécurité d'Orange Innovation

la thèse industriel (CIFRE) de Rémi Clarisse qui s'intitule « Conception de courbes elliptiques et applications ».

Les courbes elliptiques sont massivement déployées à l'heure actuelle pour garantir la sécurité de nombreuses communications, principalement sur internet. Certaines de ces courbes supportent l'opération de couplage permettant d'instancier des primitives cryptographiques comme le chiffrement basé sur l'identité ou les signatures de groupe. Cependant, il faut construire ces courbes en fonction des objectifs à remplir car tirer une courbe au hasard n'est pas faisable en pratique.

C'est dans cette approche que s'inscrit mon stage sur la recherche et l'implémentation de courbes elliptiques à couplages.

CHAPITRE 1

Outils Mathématiques

La cryptographie moderne est basée sur des résultats mathématiques et plus particulièrement sur ceux de la théorie des nombres, il est essentiel de définir et d'expliquer certains de ces résultats pour mieux comprendre le sujet de ce stage. Nous commencerons d'abord par présenter les différentes structures algébriques qui se cachent derrière les courbes elliptiques à couplage. Ensuite nous parlerons brièvement des notions relatives à la complexité. Et enfin nous définirons le couplage et les intéressantes propriétés qu'il procure.

1.1 Structures Algébriques

1.1.1 Groupes

Une structure algébrique essentielle et très utilisée en cryptographie est celle de groupe. Cette section a pour but de les définir.

Définition (Groupe)**.** Un groupe $(\mathbb{G}, \cdot)$ est un ensemble $\mathbb{G}$ muni d'une loi de composition interne vérifiant les propriétés suivantes :

Associativité : Quels que soient x, y, z dans $\mathbb{G}$ alors $(x \cdot y) \cdot z = x \cdot (y \cdot z)$;

Neutre : Il existe un élément e dans $\mathbb{G}$ tel que pour tout x dans $\mathbb{G}$, $x \cdot e = e \cdot x = x$;

Inverse : Quel que soit x dans $\mathbb{G}$, il existe y dans $\mathbb{G}$ tel que $x \cdot y = y \cdot x = e$.

De plus, si $x \cdot y = y \cdot x$ pour tout couple (x, y) dans $\mathbb{G}^2$, on dit que le groupe est *commutatif* ou *abélien*.

Notation. L'inverse d'un élément x est noté x^{-1} lorsque le groupe $\mathbb{G}$ est dit multiplicatif et $-x$ lorsque le groupe est dit additif. Aussi, on note l'élément neutre $1_{\mathbb{G}}$ lorsque $\mathbb{G}$ est multiplicatif et $0_{\mathbb{G}}$ lorsque qu'il est additif.

Définition (Sous-groupe)**.** Soit $(\mathbb{G}, \cdot)$ un groupe et $\mathbb{H}$ un sous-ensemble de $\mathbb{G}$. On dit que $(\mathbb{H}, \cdot)$ est un sous-groupe de $(\mathbb{G}, \cdot)$ si les deux conditions suivantes sont satisfaites :

- $1_{\mathbb{G}} \in \mathbb{H}$;
- $\forall (x, y) \in \mathbb{H}^2$, $x \cdot y^{-1} \in \mathbb{H}$.

Notation. Pour simplifier l'écriture, on note le groupe $(\mathbb{G}, \cdot)$ simplement $\mathbb{G}$ et la loi $\cdot$ est considérée multiplicative.

Définition (Sous-groupe engendré par un élément)**.** Soit $x \in \mathbb{G}$, on appelle le sous-groupe engendré par x l'ensemble suivant :

$$\{x^n, n \in \mathbb{N}\}$$

que l'on note $\langle x \rangle$. On dit que $\mathbb{G}$ est *cyclique* s'il existe un élément $x \in \mathbb{G}$ tel que $\langle x \rangle = \mathbb{G}$.

Définition (Ordre d'un groupe)**.** L'ordre $\#\mathbb{G}$ est le cardinal du groupe $\mathbb{G}$. L'ordre d'un élément $x \in \mathbb{G}$ est exactement l'ordre du sous-groupe $\langle x \rangle$ engendré par x. Autrement dit, l'ordre de x est le plus petit entier t tel que $x^t = 1_{\mathbb{G}}$.

Le théorème suivant indique d'une part que l'ordre d'un élément divise l'ordre du groupe mais aussi que tout groupe d'ordre premier est cyclique et que chacun de ses éléments est un générateur, sauf le neutre.

Théorème (Lagrange)**.** *Soient $\mathbb{G}$ un groupe fini et $\mathbb{H}$ un sous-groupe de $\mathbb{G}$. Alors $\#\mathbb{H}$ divise $\#\mathbb{G}$.*

Théorème (Classification des groupe abéliens finis)**.** *Soit $\mathbb{G}$ un groupe abélien fini. Il existe une suite d'entiers $(n_1, n_2, \ldots, n_k)$ telle que $\mathbb{G}$ soit isomorphe au produit de groupes cycliques suivant :*

$$\mathbb{G} \simeq (\mathbb{Z}/n_1\mathbb{Z}) \times (\mathbb{Z}/n_2\mathbb{Z}) \times \cdots \times (\mathbb{Z}/n_k\mathbb{Z})$$

où a_{i+1} divise a_i, pour tout $i \in \{1, \ldots, k-1\}$. On appelle les éléments de cette suite les facteurs invariants de $\mathbb{G}$.

Définition (Morphisme de groupes)**.** Soient $(\mathbb{G}_1, \star)$ et $(\mathbb{G}_2, *)$ deux groupes. Un *morphisme de groupes* est une application $\psi : \mathbb{G}_1 \to \mathbb{G}_2$ telle que

$$\forall (x, y) \in \mathbb{G}_1^2, \quad \psi(x \star y) = \psi(x) * \psi(y).$$

1.1.2 Anneaux

Définition (Anneau)**.** Un *anneau* $(\mathbb{A}, +, \times)$ est un ensemble $\mathbb{A}$ muni de deux opérations $+$ et $\times$ telles que :

Neutre additif : il existe un élément $0_{\mathbb{A}} \in \mathbb{A}$ tel que $(\mathbb{A}, +)$ soit un groupe abélien ;

Associativité : $\times$ est associative, *i.e* $\forall (x, y, z) \in \mathbb{A}^3, (x \times y) \times z = x \times (y \times z)$;

Distributivité : $\times$ est distributive sur la loi $+$, *i.e*

$$\forall (x, y, z) \in \mathbb{A}^3, x \times (y + z) = x \times y + x \times z.$$

On dit de plus que l'anneau est *unitaire* s'il existe un élément $1_{\mathbb{A}} \in \mathbb{A}$ neutre pour la loi $\times$.

Nous ne considérons dans la suite que des anneaux unitaires et commutatifs.

Définition (Morphisme d'anneaux)**.** Soient $(\mathbb{A}_1, +_{\mathbb{A}_1}, \times_{\mathbb{A}_1})$ et $(\mathbb{A}_2, +_{\mathbb{A}_2}, \times_{\mathbb{A}_2})$ deux anneaux. Un *morphisme d'anneaux* est une application $\psi : \mathbb{A}_1 \to \mathbb{A}_2$ telle que $\psi(1_{\mathbb{A}_1}) = 1_{\mathbb{A}_2}$ et

$$\forall (x, y) \in \mathbb{A}_1^2, \quad \psi(x +_{\mathbb{A}_1} y) = \psi(x) +_{\mathbb{A}_2} \psi(y),$$

$$\forall (x, y) \in \mathbb{A}_1^2, \quad \psi(x \times_{\mathbb{A}_1} y) = \psi(x) \times_{\mathbb{A}_2} \psi(y).$$

De plus, le *noyau* de ψ est l'ensemble :

$$\mathrm{Ker}(\psi) = \{x \in \mathbb{A}_1 : \psi(x) = 0_{\mathbb{A}_2}\} \subset \mathbb{A}_1,$$

et l'*image* de ψ l'ensemble :

$$\mathrm{Im}(\psi) = \{y \in \mathbb{A}_2 : \exists x \in \mathbb{A}_1, y = \psi(x)\} \subset \mathbb{A}_2.$$

Notation. Pour simplifier l'écriture, on note l'anneau $(\mathbb{A}, +, \times)$ simplement $\mathbb{A}$.

Définition (Idéal)**.** Soit $\mathbb{A}$ un anneau. Un idéal I de $\mathbb{A}$ est un ensemble vérifiant les propriétés suivantes :

- $(I, +)$ est un sous groupe de $(\mathbb{A}, +)$;
- $\forall a \in \mathbb{A}, \forall x \in I, a \times x \in I$.

Cette dernière propriété est appelée *absorption*, autrement dit I est stable sous multiplication des éléments de l'anneau $\mathbb{A}$. Par exemple, le noyau d'un morphisme est un idéal.

Définition (Anneau quotient)**.** Soient $\mathbb{A}$ un anneau et I un idéal de $\mathbb{A}$. La *classe d'équivalence* de x est l'ensemble :

$$\overline{x} = \{y \in \mathbb{A} : x - y \in \mathrm{I}\}$$

L'ensemble des classes d'équivalence est appelé anneau quotient, il est noté $\mathbb{A}/I$.

Théorème (Théorème d'isomorphisme)**.** *Soient $\mathbb{A}_1$ et $\mathbb{A}_2$ deux anneaux. Soit $\psi : \mathbb{A}_1 \to \mathbb{A}_2$ un morphisme d'anneaux. Alors*

$$\mathbb{A}/\,\mathrm{Ker}(\psi) \simeq \mathrm{Im}(\psi).$$

Ce théorème d'isomorphisme permet la construction d'anneaux particuliers que l'on appelle *corps finis*.

1.1.3 Corps finis

Définition (Corps fini)**.** Un corps fini est un ensemble fini que l'on note $\mathbb{F}$ et qui satisfait les propriétés suivantes :

- $(\mathbb{F}, +)$ est un groupe additif abélien dont le neutre est noté 0 ;
- $(\mathbb{F} \setminus \{0\}, \cdot)$ est un groupe multiplicatif dont le neutre est noté 1 ;
- la distributivité est respectée, *i.e.*

$$\forall a, b, c \in \mathbb{F}, \quad (a + b) \cdot c = a \cdot c + b \cdot c.$$

Définition (Ordre d'un corps fini)**.** L'ordre d'un corps fini est le nombre d'éléments de ce corps. Il existe un corps fini $\mathbb{F}_q$ d'ordre q si et seulement si q est une puissance d'un nombre premier, *i.e* $q = p^m$ avec p un nombre premier, que l'on appelle la caractéristique du corps $\mathbb{F}_q$, et m un entier positif. Il existe, à isomorphisme prés, un unique corps fini d'ordre q.

Définition (Extension de corps fini)**.** Soit p un nombre premier et $m \geq 2$. Soit $\mathbb{F}_p[X]$ l'ensemble des polynômes en l'indéterminée X à coefficients dans $\mathbb{F}_p$. Soit $f(X) \in \mathbb{F}_p[X]$ un polynôme irréductible de degré m. L'extension de $\mathbb{F}_p$ de degré m qu'on note $\mathbb{F}_{p^m}$ est défini, via le théorème d'isomorphisme, comme l'anneau quotient

$$\mathbb{F}_{p^m} \simeq \mathbb{F}_p[X]/\langle f(X)\rangle$$

où $\langle f(X)\rangle$ est l'idéal engendré par $f(X)$.

Définition (Corps algébriquement clos)**.** Un corps $\mathbb{K}$ est *algébriquement clos* lorsque tout polynôme non constant à coefficient dans $\mathbb{K}$ admet au moins une racine dans $\mathbb{K}$.

Définition (Clôture algébrique de $\mathbb{F}_p$)**.** Soit $\mathbb{F}_p$ un corps fini à p éléments. La *clôture algébrique* de $\mathbb{F}_p$ est l'extension $\overline{\mathbb{F}}_p$ de $\mathbb{F}_p$, qui est algébriquement close.

Définition (Sous-corps d'un corps fini)**.** Un sous-ensemble k d'un corps $\mathbb{K}$ est appelé un *sous corps* si k est lui-même un corps contenu dans $\mathbb{K}$. Un corps fini $\mathbb{F}_{p^m}$ n'a qu'un seul sous-corps d'ordre p^ℓ pour chaque diviseur ℓ de m. Les éléments du sous-corps $\mathbb{F}_{p^\ell}$ sont les éléments $a \in \mathbb{F}_{p^m}$ tels que $a^{p^\ell} = a$.

Définition (Corps premier)**.** Soit p un nombre premier. Dans le cas des corps finis on appelle *corps premier* le corps $\mathbb{F}_p$ qui ne contient aucun sous-corps strict. Ce corps est isomorphe à l'anneau des entiers modulo p, que l'on note $\mathbb{Z}/p\mathbb{Z}$.

On peut illustrer les extensions de corps et les sous-corps d'un corps fini avec une *tour d'extension*. Un exemple de tour d'extension est donné à la Figure 3.1.

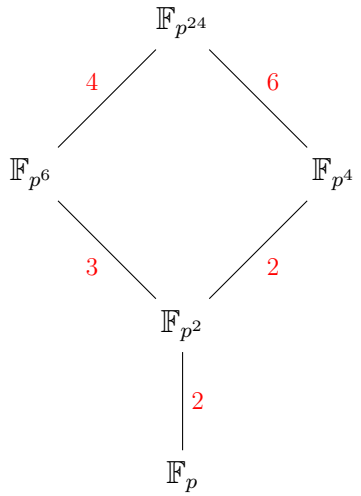

FIGURE 1.1 – Tour d'extension de corps fini avec les degrés d'extension en rouge

Définition (Base d'un corps fini)**.** Lorsque l'on considère l'ensemble $\mathbb{F}_{p^n}$ comme un espace vectoriel sur $\mathbb{F}_p$, une *base* du corps fini est simplement une base de cet espace vectoriel. Si $\mathcal{B} = \{b_1, \ldots, b_n\}$ est une base alors $a \in \mathbb{F}_p^n$ peut être représenté de façon unique par les coordonnées $(a_1, \ldots, a_n)$ dans $\mathbb{F}_p$:

$$a = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n.$$

Définition (Groupe multiplicatif d'un corps fini)**.** Soit $\mathbb{F}_q$ un corps fini. L'ensemble $\mathbb{F}_q \backslash \{0_{\mathbb{F}_q}\}$, que l'on note $\mathbb{F}_q^*$, muni de la loi $\times$, est un groupe cyclique appelé *groupe multiplicatif* de $\mathbb{F}_q$. Il existe donc au moins un élément $g \in \mathbb{F}_q^*$, appelé *élément primitif*, tel que

$$\mathbb{F}_q^* = \{g^i : 0 \leq i \leq q-2\}.$$

1.1.4 Corps de nombres

La théorie des courbes elliptiques et plus particulièrement la construction de courbes elliptiques à couplage utilise la notion de corps de nombres. Le but de cette section est de définir cette notion ainsi que certains résultats qui en découlent.

Définition (Corps de nombres)**.** Un *corps de nombres* K est un sous-corps de $\mathbb{C}$ dont le degré d'extension sur $\mathbb{Q}$ est fini. Ces corps sont engendrés par ce que l'on appelle des *nombres algébriques*, c'est-à-dire des éléments de $\mathbb{C}$ racines de polynômes dans $\mathbb{Q}[X]$.

Notation. Soit un ensemble de nombres algébriques $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. Le corps de nombre K engendré par ces éléments est

$$K = \mathbb{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n).$$

Exemple (Corps de nombres quadratiques). Soit $D \in \mathbb{Z}$ un entier sans facteur carré. Il existe deux types de corps de nombres quadratiques, *i.e.* des extensions de degré 2 sur $\mathbb{Q}$:
— les *corps quadratiques réels* sont les corps $\mathbb{Q}(\sqrt{D})$ avec $D > 0$;
— les *corps quadratiques imaginaires* sont les corps $\mathbb{Q}(\sqrt{D})$ avec $D < 0$.
Dans les deux cas

$$\mathbb{Q}(\sqrt{D}) = \{a + b\sqrt{D} : (a, b) \in \mathbb{Q}^2\}.$$

Dans la suite, on considère $D > 0$ et on écrit $\mathbb{Q}(\sqrt{-D})$ lorsqu'il s'agit de corps de nombres quadratiques imaginaires.

Définition (Anneau des entiers algébriques)**.** Soit K un corps de nombres quadratique. L'ensemble $\mathcal{O}_K$ formé par les éléments de K qui sont racines d'un polynôme unitaire de $\mathbb{Z}[X]$ est un anneau de K, appelé *anneau des entiers algébriques* de K. Il est de la forme

$$\mathcal{O}_K = \mathbb{Z} + \delta\mathbb{Z}$$

Définition (Ordre dans $\mathbb{Q}(\sqrt{-D})$)**.** Un ordre $\mathcal{O}$ dans $K = \mathbb{Q}(\sqrt{-D})$ est un sous-anneau de $\mathcal{O}_K$. C'est un groupe abélien de type fini de la forme

$$\mathcal{O} = \mathbb{Z} + f\delta\mathbb{Z}$$

où f et δ sont définis de la façon suivante

$$f = [\mathcal{O}_K : \mathcal{O}] \quad \text{et} \quad \delta = \begin{cases} \frac{1+\sqrt{-D}}{2} & \text{si } D \equiv 3 \mod 4, \\ \sqrt{-D} & \text{si } D \equiv 1, 2 \mod 4. \end{cases}$$

Enfin le *discriminant* de $\mathcal{O}$ est la quantité

$$D_{\mathcal{O}} = \begin{cases} -f^2 D & \text{si } D \equiv 3 \mod 4, \\ -4f^2 D & \text{si } D \equiv 1, 2 \mod 4. \end{cases}$$

Définition (Polynôme cyclotomique)**.** Soit k un entier positif. Notons $\zeta_k \in \mathbb{C}$ une racine primitive k-ème de l'unité, c'est-à-dire un nombre algébrique tel que $(\zeta_k)^k = 1$ et $(\zeta_k)^\ell \neq 1$ pour tout entier positif $\ell < k$. Le polynôme minimal de ζ_k est appelé *polynôme cyclotomique*, noté $\Phi_k(X) \in \mathbb{Z}[X]$. On peut définir ce polynôme récursivement en posant $\Phi_1(X) = X - 1$ et utiliser la formule :

$$X^k - 1 = \prod_{d|k} \Phi_d(X), \quad \text{pour } k > 1.$$

Le degré de Φ_k est l'indicatrice d'Euler évaluée en k :

$$\deg(\Phi_k) = \varphi(k) = \#\{e \in \mathbb{N}^* : e \leq k, \text{pgcd}(e, k) = 1\}.$$

Exemple (Corps cyclotomique). Soit ζ_k une racine primitive k-ème de l'unité. On appelle k-ème corps cyclotomique le corps de nombre engendré par ζ_k noté $\mathbb{Q}(\zeta_k)$. Une façon de construire cette extension de $\mathbb{Q}$ est de « quotienter » l'anneau $\mathbb{Q}[X]$ par l'idéal engendré par le polynôme cyclotomique $\Phi_k(X)$, ainsi

$$\mathbb{Q}(\zeta_k) \simeq \mathbb{Q}[X]/\langle \Phi_k(X) \rangle.$$

De la même façon qu'avec les corps finis, nous pouvons illustrer ces extensions à l'aide de tour d'extensions.

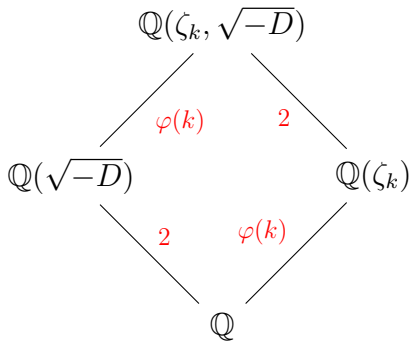

FIGURE 1.2 – Tour d'extension de corps de nombres quadratiques et cyclotomiques

Remarque. On suppose dans cette tour d'extension que $\sqrt{-D} \notin \mathbb{Q}(\zeta_k)$. Si $\sqrt{-D} \in \mathbb{Q}(\zeta_k)$, comme c'est le cas pour $k = 3$ et $D = -1$, alors le degré d'extension de $\mathbb{Q}(\sqrt{-D}, \zeta_k)/\mathbb{Q}$ est $\varphi(k)$.

1.1.5 Courbes elliptiques et cryptographie

L'objectif de cette section est de présenter l'arithmétique des courbes elliptiques et leur utilisation en cryptographie.

Forme de Weierstrass

Définition (Weierstrass longue)**.** Une courbe elliptique E sur un corps $\mathbb{K}$ est définie par l'équation suivante :

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

où $a_i \in \mathbb{K}$ pour tout i dans $\{1, 2, 3, 4, 6\}$. On appelle cette équation de courbe la forme *longue de Weierstrass*. De plus, si $\mathbb{L}$ est une extension de corps de $\mathbb{K}$ alors l'ensemble des *points $\mathbb{L}$-rationnels* est

$$E(\mathbb{L}) = \{(x, y) \in \mathbb{L} \times \mathbb{L} \ : \ y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0\} \cup \{\infty\}$$

où ∞ est appelé le point à l'infini.

Définition (Weierstrass courte)**.** En fonction de la caractéristique du corps $\mathbb{K}$, par un simple changement de variable, on peut simplifier l'équation de Weierstrass longue. Si la caractéristique du corps $\mathbb{K}$ est différente de 2 et 3, en effectuant le changement de variables [HMV04]

$$(x, y) \mapsto \left(\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1x}{216} - \frac{a_1^3 + 4a_1a_2 - 12a_3}{24} \right)$$

on obtient l'équation :

$$E : y^2 = x^3 + ax + b,$$

avec $a, b \in \mathbb{K}$.

Une illustration graphique d'une transformation permettant de passer de la forme longue à la forme courte de Weierstrass sur $\mathbb{R}$ est donnée dans la Figure 1.3.

Dans la suite de ce manuscrit, nous ne considérerons que des corps de caractéristique différente de 2 et 3, car les courbes elliptiques définies sur des corps binaires et ternaires n'offrent pas une sécurité suffisante dans notre contexte. Par conséquent, toutes les courbes elliptiques que nous utiliserons adopteront la forme courte de Weierstrass :

$$y^2 = x^3 + ax + b. \tag{1.1}$$

Étant donné un corps $\mathbb{K}$, on peut se demander si tous les couples $(a, b) \in \mathbb{K}^2$ donnent naissance à une courbe elliptique d'équation (1.1). Une courbe cubique est dite elliptique si et seulement si elle ne possède pas de point singulier. En effet, tout point (x, y) de la courbe annule la fonction $f(x, y) = y^2 - (x^3 + ax + b)$ et donc s'il existe un point $P = (x_P, y_P)$ tel que $\frac{\partial f}{\partial x}(x_P, y_P) = 0$ et $\frac{\partial f}{\partial y}(x_P, y_P) = 0$ simultanément alors P est un point singulier. De plus, une courbe elliptique d'équation (1.1) admet des points singuliers si et seulement si $4a^3 + 27b^2 = 0$.

En cryptographie, on utilise principalement des courbes elliptiques définies sur des corps finis, mais pour mieux illustrer les propriétés de ces courbes on utilise des représentations graphiques de courbes elliptiques définies sur $\mathbb{R}$.

Donnons comme exemples différentes courbes cubiques sur $\mathbb{R}$:

Groupe sur une courbe elliptique

Définition (Loi de groupe)**.** Soit E une courbe elliptique définie sur un corps $\mathbb{K}$. Il existe une opération d'addition sur l'ensemble des points de cette courbe appelée *règle de la corde et de la tangente* (chord-and-tangent rule en anglais). L'ensemble des points sur la courbe $E/\mathbb{K}$, adjoint de ∞, muni de cette opération d'addition + forme un groupe abélien. Intuitivement sur $\mathbb{R}$, cette loi d'addition utilise le fait qu'une courbe linéaire intersecte toujours trois fois une courbe cubique. La multiplicité d'un point est le nombre de fois que cette courbe linéaire intersecte la courbe en ce même point.

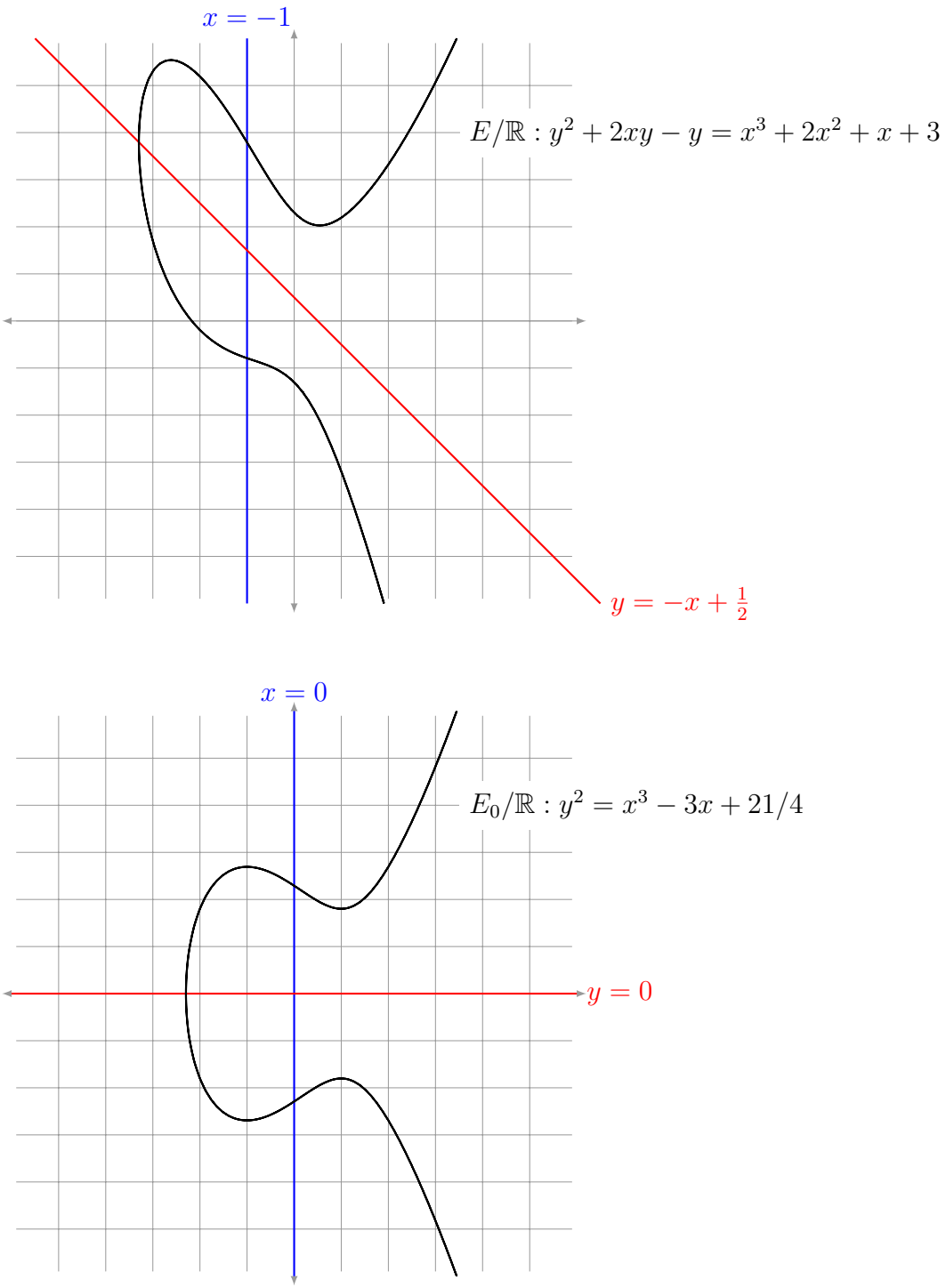

FIGURE 1.3 – Transformation de la courbe d'équation $y^2 + 2xy - y = x^3 + 2x^2 + x + 3$ en la courbe d'équation $y^2 = x^3 - 3x + 21/4$

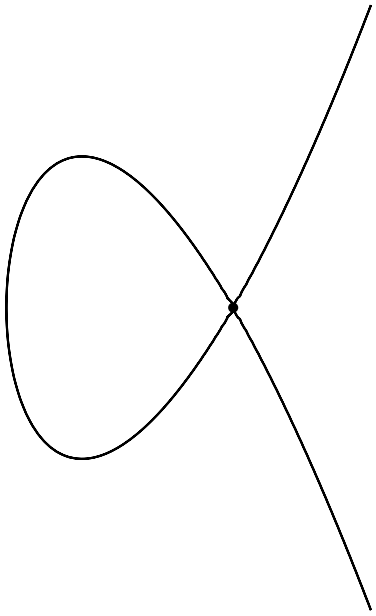

FIGURE 1.4 – Courbe d'équation $y^2 = x^3 - 3x + 2$ avec un point singulier de multiplicité 2

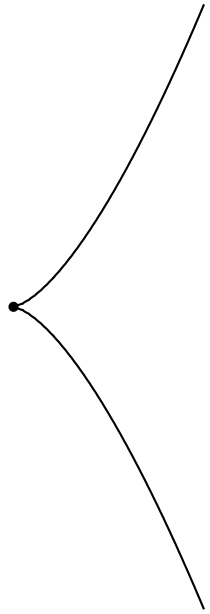

FIGURE 1.5 – Courbe d'équation $y^2 = x^3$ avec un point singulier de rebroussement

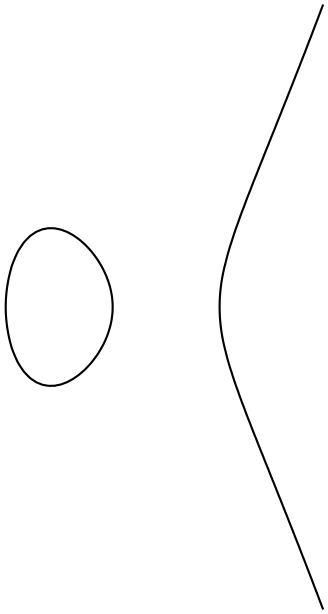

FIGURE 1.6 – Courbe d'équation $y^2 = x^3 - 2x$ avec deux composantes connexes

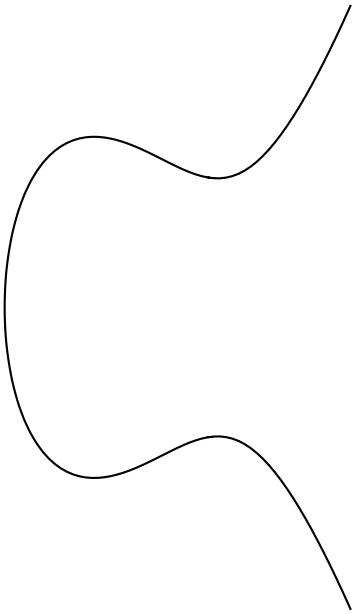

FIGURE 1.7 – Courbe d'équation $y^2 = x^3 - 2x + 2$

Addition. Soient P et Q deux points distincts d'une courbe elliptique E. Pour calculer le point $R = P + Q$, on trace la droite ℓ passant par P et Q. Cette droite ℓ coupe la courbe E en un troisième point $-R$. Le point R est alors le point symétrique de $-R$ par rapport à l'axe des abscisses. Voir la figure 1.8.

Formule d'addition. Déterminons les formules explicites pour le calcul de l'addition de deux points $P = (x_P, y_P)$ et $Q = (x_Q, y_Q)$ d'une courbe elliptique E. Soit la ligne

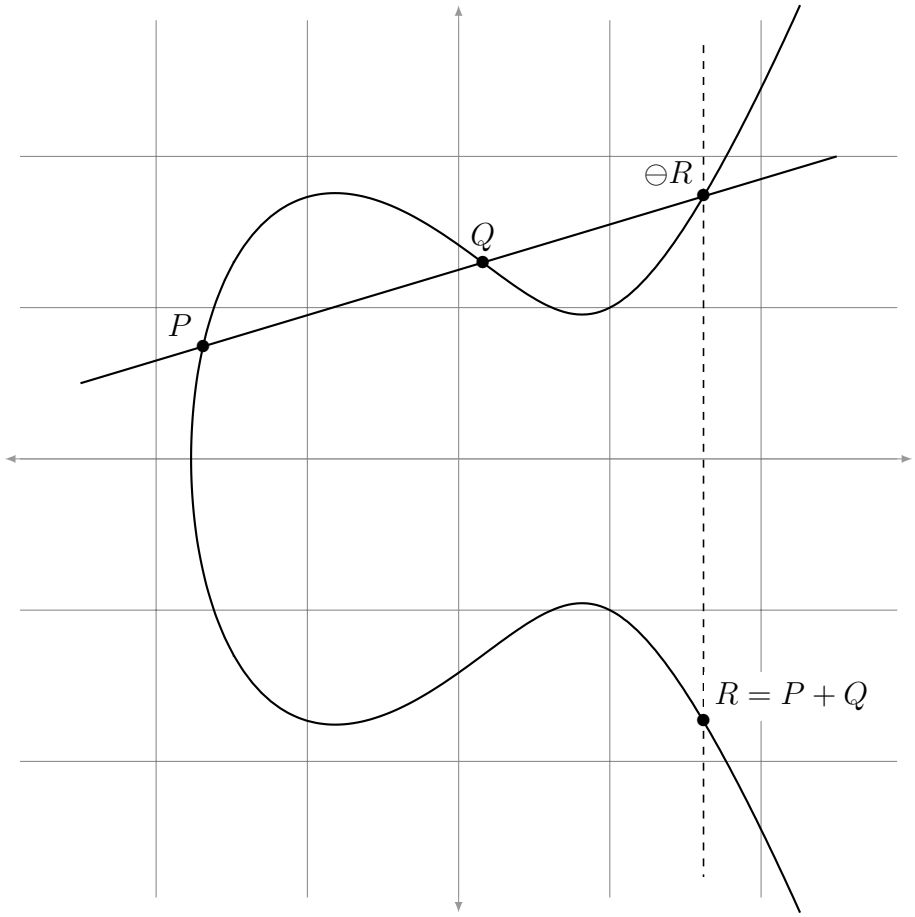

FIGURE 1.8 – Addition des points P et Q de la courbe d'équation $y^2 = x^3 - 2x + 2$ sur $\mathbb{R}$

$\ell : y = \lambda x + \mu$ qui intersecte le courbe E en P et Q. Ainsi, le coefficient directeur de ℓ est

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$$

et il s'ensuit que

$$\mu = \frac{y_Q x_P - y_P x_Q}{x_P - x_Q}.$$

De plus, on sait que ℓ intersecte E en un troisième point $\ominus R = (x_R, -y_R)$. Les abscisses de ces points sont donc racines d'un polynôme de degré 3, correspondant à la substitution de l'équation de la droite dans l'équation de la courbe :

$$(x - x_P)(x - x_Q)(x - x_R) = (x^3 + ax + b) - (\lambda x + \mu)^2.$$

En développant le membre de droite, on obtient

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\mu)x + (b - \mu^2)$$

ce qui nous permet d'identifier les coefficients devant x^2

$$\lambda^2 = x_P + x_Q + x_R.$$

Finalement, le point $R = (x_R, y_R)$ est tel que

$$x_R = \lambda^2 - x_P - x_R \quad \text{et} \quad y_R = -(\lambda x_R + \mu).$$

Expliquons maintenant le cas où $P = Q$ qui est l'opération de doublement.

Doublement. Soit $P = (x, y)$ un point d'une courbe elliptique E. Pour calculer le point $R = P \oplus P$, on trace la tangente de E au point P, cette tangente coupe alors la courbe en un deuxième point $\ominus R$. On obtient ainsi le point R en prenant le symétrique de $\ominus R$ par rapport à l'axe des abscisses.

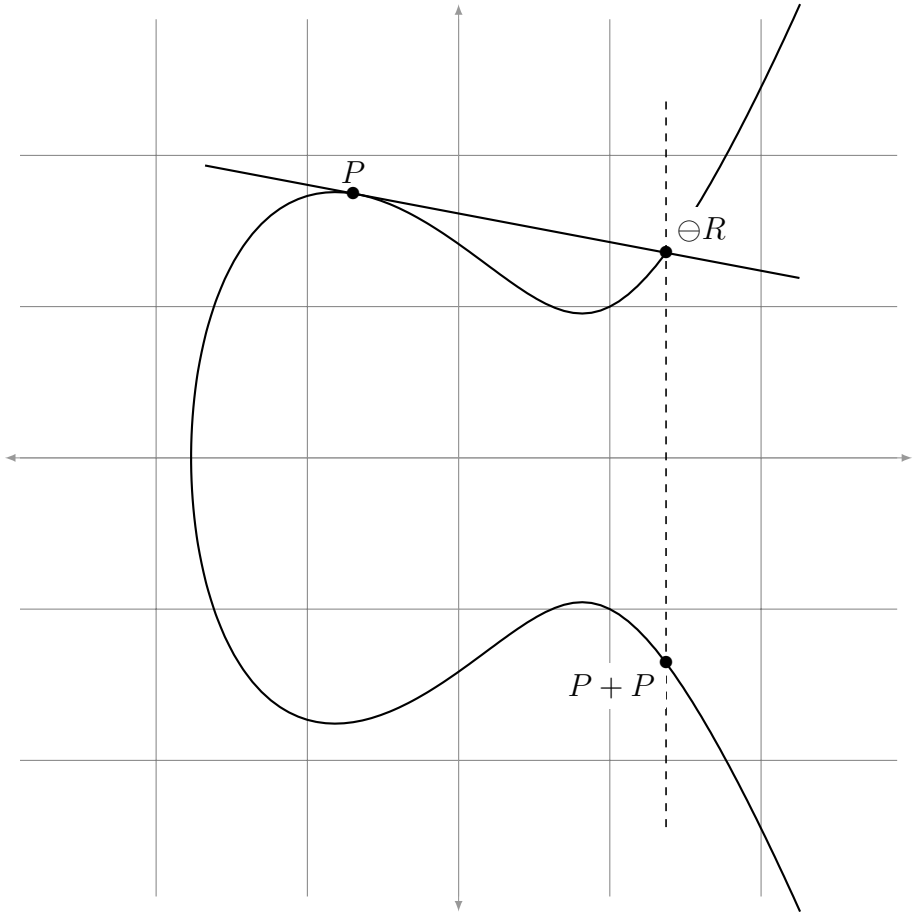

FIGURE 1.9 – Doublement du point P de la courbe d'équation $y^2 = x^3 - 2x + 2$ sur $\mathbb{R}$

Formule de doublement. Dans le cas d'un doublement de point, $\ell : y = \lambda x + \mu$ est la tangente de E au point P. Donc le coefficient directeur de cette tangente λ est la fonction dérivée dy/dx évaluée au point P. Posons $z = f(x, y) = y^2 - x^3 - ax - b$, f est différentiable et la différentielle totale s'écrit :

$$dz = \frac{\partial f}{\partial x}(x, y) \times dx + \frac{\partial f}{\partial y}(x, y) \times dy$$

On en tire la dérivée totale de z par rapport à x

$$\frac{dz}{dx} = \frac{\partial f}{\partial x}(x, y) + \frac{\partial f}{\partial y} \times \frac{dy}{dx}$$

Or comme $f(x, y) = 0$ au point P on a $\frac{dz}{dx} = 0$, d'où

$$\frac{dy}{dx} = -\frac{\dfrac{\partial f}{\partial x}}{\dfrac{\partial f}{\partial y}}$$

Ainsi, comme $\dfrac{\partial f}{\partial x} = -3x^2 - a$ et $\dfrac{\partial f}{\partial y} = 2y$ on a bien

$$\lambda = \frac{3x_P^2 + a}{2y_P}.$$

Ce qui nous donne *in fine*

$$x_R = \lambda^2 - 2x_P \quad \text{et} \quad y_R = -(\lambda x_R + \mu).$$

Neutre. Le point ∞ joue le rôle de l'élément neutre du groupe. C'est le point d'intersection de la courbe elliptique avec toutes les droites verticales, illustré par la Figure 1.10.

Définition (Ordre de $E(\mathbb{F}_q)$)**.** Soit E une courbe elliptique définie sur un corps fini $\mathbb{F}_q$. L'ordre $\#E(\mathbb{F}_q)$ est le nombre de points $\mathbb{F}_q$-rationnels. De plus, comme l'équation de Weierstrass admet au plus deux solutions pour tout $x \in \mathbb{F}_q$ alors $\#E(\mathbb{F}_q) \in [1, 2q+1]$. Ce n'est qu'en 1936 que Helmut Hasse énonce un encadrement plus précis :

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

Ces bornes sont appelées *bornes de Hasse* et elles peuvent être reformulées :

$$\#E(\mathbb{F}_q) = q + 1 - t, \quad \text{avec } |t| \leq 2\sqrt{q}, \tag{1.2}$$

où t est appelé *la trace du Frobenius* de $E/\mathbb{F}_q$, qui est définie plus loin. De plus, lorsque $\text{pgcd}(t, q) = 1$, la courbe E est dite *ordinaire* sinon elle est dite *supersingulière.*

De la même façon, si E est définie sur $\mathbb{F}_q$ alors il est possible de considérer les points de E sur une extension $\mathbb{F}_{q^n}$ de $\mathbb{F}_q$. On sait alors que le groupe $E(\mathbb{F}_q)$ est un sous groupe de $E(\mathbb{F}_{q^n})$, par conséquent $\#E(F_q)$ divise $\#E(F_{q^n})$. Le théorème suivant donne une formule explicite pour calculer l'ordre de $E(\mathbb{F}_{q^n})$ sur n'importe quelle extension $\mathbb{F}_{q^n}$ de $\mathbb{F}_q$.

Théorème (Ordre de $E(\mathbb{F}_{q^n})$)**.** *Soit E une courbe elliptique définie sur $\mathbb{F}_q$ et telle que $\#E(\mathbb{F}_q) = q + 1 - t$. Alors*

$$\#E(\mathbb{F}_{q^n}) = q^n + 1 - t_n$$

où $(t_n)_{n\in\mathbb{N}}$ est la suite définie par la récurrence [HMV04, Théorème 3.11] :

$$\begin{cases} (t_0, t_1) &= (2, t), \\ t_n &= t_1 t_{n-1} - q t_{n-2}. \end{cases}$$

Structure de groupe

Théorème (Structure de groupe sur courbe elliptique)**.** *Soit E une courbe elliptique définie sur $\mathbb{F}_q$. Alors le groupe $E(\mathbb{F}_q)$ est isomorphe à $\mathbb{Z}/n_1\mathbb{Z} \times \mathbb{Z}/n_2\mathbb{Z}$ avec n_1 et n_2 deux entiers uniques tels que n_2 divise n_1 et $q - 1$.*

Définition (Sous-groupe de torsion)**.** Soit $E(\mathbb{F}_q)$ une courbe elliptique définie sur $\mathbb{F}_q$ et d'ordre $\#E(\mathbb{F}_q)$. Le théorème de Lagrange indique que l'ordre de chaque point de $E(\mathbb{F}_q)$ divise $\#E(\mathbb{F}_q)$. Pour chaque diviseur ℓ de $\#E(\mathbb{F}_q)$, le *sous-groupe de ℓ-torsion* sur $\mathbb{F}_q$ est l'ensemble suivant

$$E(\mathbb{F}_q)[\ell] = \{P = (x_P, y_P) \in E(\mathbb{F}_q) : \ell P = \infty\}$$

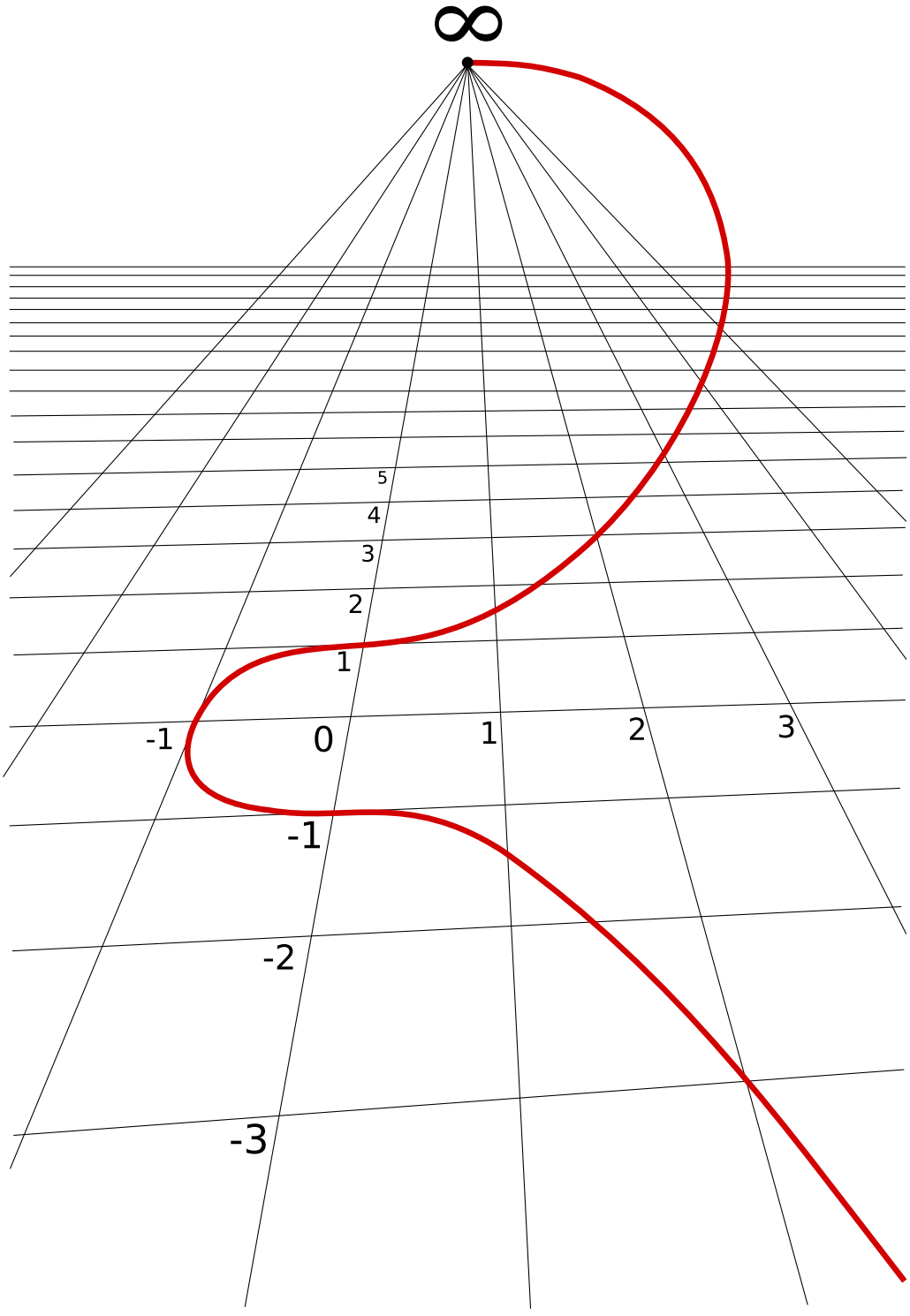

FIGURE 1.10 – Illustration d'une courbe sur $\mathbb{R}$ d'équation $y^2 = x^3 + 1$ avec le point à l'infinie

Notation. Le groupe de ℓ-torsion sur la clôture algébrique $\overline{\mathbb{F}}_q$ de $\mathbb{F}_q$ est noté $E[\ell]$.

Théorème. *Soit p la caractéristique de $\mathbb{F}_q$. Si p ne divise pas ℓ alors on a,*

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

Ce qui indique que $\#E[\ell] = \ell^2$ et que la ℓ-torsion est constituée de $\ell+1$ groupes cycliques. Or, comme $E[\ell]$ est fini, ses éléments sont définis sur une certaine extension de $\mathbb{F}_q$.

Définition (Degré de plongement)**.** Soit une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$. Soit ℓ un premier ne divisant pas q. On appelle le degré de plongement de q par rapport à ℓ, le plus petit entier k tel que

$$E[\ell] \subset E(\mathbb{F}_{q^k})$$

C'est aussi l'ordre de q dans $\mathbb{Z}/\ell\mathbb{Z}$. De plus si ℓ ne divise $\#E(\mathbb{F}_q)$ qu'une seul fois alors il n'y a qu'un seul sous-groupe d'ordre ℓ dans $E(\mathbb{F}_q)$. C'est donc grâce à cet entier k qu'on peut obtenir d'autres sous-groupes cycliques d'ordre ℓ.

Ainsi lorsque une courbe définie par la même équation est considérée sur le corps $\mathbb{F}_{q^k}$, on assiste à une sorte « d'éclosion » de la structure de groupe.

Anneau d'endomorphisme

Définition (Anneau d'endomorphisme)**.** Soit E une courbe elliptique définie sur un corps fini $\mathbb{F}_q$. L'ensemble des morphismes de groupes de $E(\mathbb{F}_q)$ dans $E(\mathbb{F}_q)$ muni des lois d'addition $+$ et de composition $\circ$, forme un anneau qu'on appelle *anneau d'endomorphismes* et qu'on note $\mathrm{End}(E)$, à savoir pour tous $\psi_1, \psi_2 \in \mathrm{End}(E)$:

$$\begin{aligned}(\psi_1 + \psi_2)(P) &= \psi_1(P) + \psi_2(P),\\ (\psi_1 \circ \psi_2)(P) &= \psi_1(\psi_2(P)),\end{aligned}$$

quel que soit $P \in E$.

Les paragraphes suivants définissent quelques exemples d'endomorphismes d'anneaux que l'on utilisera pour construire des courbes elliptiques à couplages.

Exemple (Multiplication scalaire). Soit E une courbe elliptique définie sur $\mathbb{F}_q$. Soit $m \in \mathbb{Z}$, la multiplication-par-m définie par

$$\begin{aligned}[m] : E &\longrightarrow E\\ P &\longmapsto mP := \underbrace{P + \cdots + P}_{m \text{ fois}}\end{aligned}$$

est un endomorphisme de E. Remarquons également que le noyau de l'application $[m]$ est le groupe de m-torsion.

Exemple (Endomorphisme de Frobenius). Soit E une courbe elliptique définie sur $\mathbb{F}_q$. L'endomorphisme de Frobenius est défini par

$$\begin{aligned}\pi : E(\overline{\mathbb{F}_q}) &\longrightarrow E(\overline{\mathbb{F}_q})\\ (x, y) &\longmapsto (x^q, y^q)\end{aligned}$$

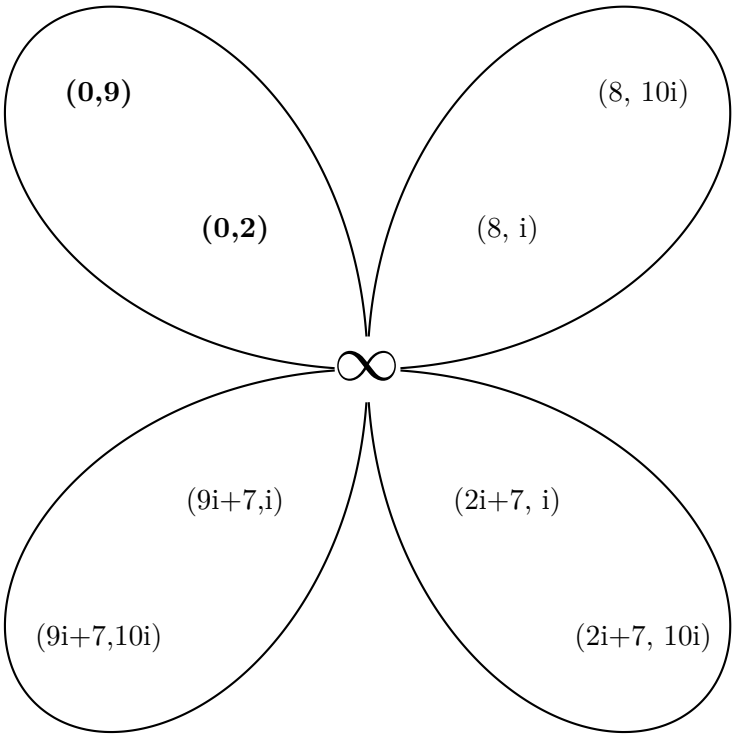

FIGURE 1.11 – Groupe de 3-torsion E[3] sur $\mathbb{F}_{q^2} = \mathbb{F}_q(i)$ avec $q = 11$ et la courbe E d'équation $y^2 = x^3 + 4$

est un endomorphisme de E qui laisse fixe les points de $E(\mathbb{F}_q)$. Autrement dit, quel que soit $P \in E(\mathbb{F}_q)$ on a,

$$\pi(P) = P.$$

Il s'ensuit,

$$\#E(\mathbb{F}_q) = \#\operatorname{Ker}(\pi - [1]).$$

De plus, pour tout $P \in E(\overline{\mathbb{F}_q})$,

$$\pi^2(P) = [t] \circ \pi(P) - [q]P$$

où t est appelé la trace du Frobenius. Cela signifie en d'autres termes que le polynôme minimal de π est

$$X^2 - tX + q.$$

Notons que l'endomorphisme π a deux valeurs propres : 1 et q. Par conséquent, les sous-espaces propres associés $\operatorname{Ker}(\pi - [1])$ et $\operatorname{Ker}(\pi - [q])$ sont non vides.

Exemple (Trace et anti-trace). Soient E une courbe elliptique définie sur $\mathbb{F}_q$ et $k \in \mathbb{Z}$. La trace de $P \in E(\mathbb{F}_{q^k})$ est définie par :

$$\begin{aligned} \mathrm{Tr} : E(\mathbb{F}_{q^k}) &\longrightarrow E(\mathbb{F}_q) \\ P &\longmapsto \sum_{i=0}^{k-1} \pi^i(P) \end{aligned}$$

Soit ℓ un entier premier ne divisant pas q. Supposons que k est le degré de plongement de q par rapport à ℓ. Alors

$$\forall P \in E[\ell], \quad \mathrm{Tr}(P) \in E[\ell] \cap \mathrm{Ker}(\pi - [1])$$

On appelle $E[\ell] \cap \mathrm{Ker}(\pi - [1])$ le *sous-groupe du corps de base*. De plus, on montre assez facilement que,

$$\forall P \in E[\ell] \cap \mathrm{Ker}(\pi - [q]), \quad \mathrm{Tr}(P) = \infty$$

On appelle alors $E[\ell] \cap \mathrm{Ker}(\pi - [q])$ le *sous-groupe de trace nulle*. L'anti-trace de $P \in E[\ell]$ est définie par :

$$\begin{aligned} \mathrm{aTr} : E[\ell] &\longrightarrow E[\ell] \cap \mathrm{Ker}(\pi - [q]) \\ P &\longmapsto [k]P - \mathrm{Tr}(P) \end{aligned}$$

Exemple (Endomorphisme GLV). En 2001, Gallant, Lambert et Vanstone ont montré qu'il était possible utiliser des endomorphismes pour accélérer la multiplication scalaire dans $E(\mathbb{F}_q)[\ell]$. L'idée est simple : trouver un endomorphisme $\Phi \in \mathrm{End}(E)$ tel que :

$$\forall P \in E(\mathbb{F}_q)[\ell], \quad \Phi(P) = \lambda P$$

où λ est une racine du polynôme minimal de Φ modulo ℓ. Ainsi, si on veut calculer aP pour un entier $a \in \mathbb{Z}/\ell\mathbb{Z}$ pris aléatoirement, il suffit de calculer le quotient a_1 et le reste a_0 de la division euclidienne de a par λ pour calculer aP. En effet, si $a = a_0 + \lambda a_1$ alors on a

$$\begin{aligned} aP &= (a_0 + a_1\lambda)P \\ &= a_0P + a_1(\lambda P) \\ &= a_0P + a_1\Phi(P) \end{aligned}$$

L'avantage d'utiliser cet endomorphisme Φ est de réduire de moitié le coût de la multiplication dans $E(\mathbb{F}_q)[\ell]$ puisque l'évaluation de Φ est bien moins chère que la multiplication par λ (elle est même gratuite si $\Phi(P)$ est pré-calculé).

Twist de courbes elliptiques

Définition (Twist)**.** Soient E et E' deux courbes elliptiques définies sur $\mathbb{F}_q$. On dit que E' est la twist de degré d de E si il existe un isomorphisme $\Psi_d : E' \to E$ sur $\mathbb{F}_{q^d}$ et si d est minimal.

Les seuls degrés de twist possible sont $d \in \{2, 3, 4, 6\}$ [Sil09, Prop. X.5.4]. Généralement, si $E : y^2 = x^3 + ax + b$ alors,

$$E' : y^2 = x^3 + a\omega^4 x + b\omega^6$$

Twist	E	E'
Quadratic	$y^2 = x^3 + ax + b$	$y^2 = x^3 + a\omega^4 x + b\omega^6$
Cubic	$y^2 = x^3 + b$	$y^2 = x^3 + b\omega^6$
Quartic	$y^2 = x^3 + ax$	$y^2 = x^3 + a\omega^4 x$
Sextic	$y^2 = x^3 + b$	$y^2 = x^3 + a\omega^4 x + b\omega^6$

avec ω appartenant à une certaine extension de $\mathbb{F}_q$ et l'isomorphisme entre E et E' est

$$\begin{aligned} \Psi_d : E' &\longrightarrow E \\ (x', y') &\longmapsto \left(\frac{x'}{\omega^2}, \frac{y'}{\omega^3}\right) \end{aligned}$$

On a donc différents types de twist résumés dans ce tableau :

Théorème (Ordre de E'). *Soit E une courbe elliptique définie sur $\mathbb{F}_q$ avec $\#E(\mathbb{F}_q) = q + 1 - t$ (voir 1.2). Supposons que cette courbe E admette une twist E' de degré d, alors en fonction de $d \in \{3, 4, 6\}$ les possibles ordres de E' sur $\mathbb{F}_q$ sont :*

d	$\#E'(\mathbb{F}_q)$	$t^2 - 4q$
3	$q + 1 - \frac{\pm 3f - t}{2}$	$-3f^2$
4	$q + 1 \pm f$	$-f^2$
6	$q + 1 - \frac{\pm 3f - t}{2}$	$-3f^2$

TABLE 1.1 – Ordre de la twist E'

Remarque. Toutes les courbes elliptiques admettent une twist d'ordre 2 [Sil09].

Diviseur

Définition (Diviseur). Soit E une courbe elliptique définie sur $\mathbb{F}_q$. Un diviseur D sur la courbe E est une somme formelle finie de points de E. À savoir,

$$D = \sum_{P \in E} a_P(P)$$

où un nombre fini des $a_P \in \mathbb{Z}$ sont non nuls.

Définition (Groupe des diviseur). Soit E une courbe elliptique définie sur $\mathbb{F}_q$. L'ensemble des diviseurs sur la courbe E noté $\mathrm{Div}_{\overline{\mathbb{F}_q}}(E)$, muni de la loi d'addition forme un groupe abélien. L'élément neutre de ce groupe est le diviseur dont tous les a_P sont nuls. La loi de d'addition se fait naturellement en sommant les coefficients entre eux.

Définition (Degré et support d'un diviseur)**.** Soit E une courbe elliptique définie sur $\mathbb{F}_q$. Soit $D \in \mathrm{Div}_{\overline{\mathbb{F}_q}}(E)$ un diviseur de cette courbe. Le degré de D est la somme

$$\deg(D) = \sum_{P \in E} a_P$$

et le support de D est l'ensemble

$$\mathrm{Supp}(D) = \{P \in E(\overline{\mathbb{F}_q}\} : a_P \neq 0\}$$

Définition (Diviseur principal)**.** Soient E une courbe elliptique definie sur $\mathbb{F}_q$ et f une fonction sur E. On note $\mathrm{ord}_P(f)$ la multiplicité de f au point P avec

$\mathrm{ord}_P(f) < 0$ si P est un pôle de f
$\mathrm{ord}_P(f) > 0$ si P est un zero de f
$\mathrm{ord}_P(f) = 0$ si P n'est ni un zero ni un pole de f

On définit le diviseur de f comme

$$(f) = \sum_{P \in E} \mathrm{ord}_P(f)(P)$$

On dit que D est un diviseur principal s'il existe une fonction f telle que $D = (f)$. En particulier, on a l'équivalence suivante [Men97, Théorème 2.5]

$$D \text{ est principal } \iff \deg(D) = 0 \text{ et } \sum_{P \in E} a_P P = \infty$$

Définition (Diviseur équivalent)**.** Soient D_1 et D_2 deux diviseurs d'une courbe elliptique E définie sur $\mathbb{F}_q$. On dit que D_1 et D_2 sont équivalents, $D_1 \sim D_2$, si leur différence, $D_1 - D_2$, est un diviseur principal.

1.2 Complexité

En cryptographie, la sécurité des protocoles repose sur des hypothèses de sécurité. Ces hypothèses consistent à supposer que la probabilité de succès qu'un *algorithme probabiliste polynomial* a de résoudre un certain problème donné est majorée par une *fonction négligeable*. On dira alors que le problème est difficile. Le but de cette section est de définir ces termes.

1.2.1 Préliminaires

Définition (Algorithme probabiliste polynomial)**.** On dit d'un algorithme A qu'il est polynomial lorsqu'il existe un polynôme $p(x)$ tel que pour toute entrée $x \in \{0,1\}^*$ (mot binaire de longueur arbitraire), le temps d'exécution de A est majorée par $p(|x|)$. De plus, si A a accès à une source d'aléa générant de manière indépendante des bits uniformément distribués sur $\{0,1\}$ on dit alors qu'il est probabiliste.

Définition (Fonction négligeable)**.** Une fonction f est dite négligeable lorsque pour tout polynôme p, il existe un entier N tel que,

$$\forall n \geq N, f(n) < \frac{1}{p(n)}$$

Complexité. La complexité des attaques est exprimés en nombre d'opérations nécessaires pour résoudre les problèmes sur lesquels se basent la sécurité des protocoles cryptographiques considérés. Les études de complexité portent dans la majorité des cas sur le comportement asymptotique, *i.e* lorsque la taille des entrées tend vers l'infini et on utilise pour cela la notation de Landau.

Définition (Notation de Landau)**.** On introduit ici la notation de Landau permettant d'exprimer des relations asymptotiques :

Grand O On note $v(n) = O(f(n))$ s'il existe une constante positive c telle que pour n suffisamment grand $|v(n)| \leq cf(n)$. On dit qu'une attaque a une complexité en $O(f(n))$ s'il existe une constante c tel que pour toute entrée de taille n le nombre d'étapes pour achever l'attaque est inférieur à $c \times f(n)$.

Petit o On note $v(n) = o(u(n))$ si,

$$\lim_{n\to\infty} \frac{v(n)}{u(n)} = 0.$$

On dit aussi que v est *négligeable* devant u.

On peut classer ces complexités en plusieurs classe à l'aide de la notation de Landau :

Complexité	Temps de calcul
Constant	$O(1)$
Logarithmique	$O(\log_2(n))$
Polynomial	$2^{O(\log_2(n))}$
Exponentiel	$2^{O(n)}$

Définition (Notation L_Q)**.** Il existe des algorithmes dont la complexité ne peut s'exprimer avec la notation de Landau. On introduit alors une autre notation qu'on appelle la *notation-L* est qui est définie par :

$$L_Q(\alpha, c) = \exp((c + o(1)) \ln^{\alpha}(Q) \ln^{1-\alpha}(\ln(Q)))$$

avec $\alpha \in [0, 1]$ et $c > 0$. Intuitivement, plus α se rapproche de 1 et plus $L_Q(\alpha, c)$ est exponentielle en la taille de $\ln Q$ et plus α est proche de 0 plus $L_Q(\alpha, c)$ est polynomial en la taille de $\ln Q$. La constante c joue un rôle important dans le domaine non asymptotique.

1.2.2 Problème du logarithme discret

Un problème sur lequel se reposent beaucoup de protocoles cryptographiques est le problème du logarithme discret. De ce fait, ce problème est très étudié en cryptanalyse comme nous allons le voir. Mais avant cela, définissons-le.

Définition (Problème du logarithme discret)**.** Soit $\mathbb{G}$ un groupe cyclique d'ordre n. Soient un générateur g du groupe $\mathbb{G}$ et un élément aléatoire g_a du groupe $\mathbb{G}$. Le problème du logarithme discret (DLP) consiste à,

$$\text{calculer } a \in \mathbb{Z}/n\mathbb{Z} \text{ tel que } g_a = g^a.$$

Suivant le choix de $\mathbb{G}$ ce problème est supposé difficile, c'est à dire qu'il n'existe pas d'algorithme polynomial résolvant ce problème. Par exemple, le groupe $\mathbb{G}$ peut être le groupe multiplicatif $\mathbb{F}_q^*$.

Attaques sur le DLP. Pour assurer la sécurité d'un protocole cryptographique basée sur DLP, on étudie la complexité en temps et en mémoire des meilleurs attaques connues afin de dimensionner les groupes en fonction de ces attaques. Listons quelques attaques connues :

Baby-Step Giant-Step & ρ de Pollard sont des attaques génériques qui calculent le logarithme discret dans un groupe $\mathbb{G}$ d'ordre n en temps $O(\sqrt{n})$. Si nous souhaitons une sécurité de s bits il faut donc que $\mathbb{G}$ soit d'ordre 2^{2s}. En effet, l'attaque calcule le log discret en $O(\sqrt{2^{2s}}) = O(2^s)$.

Pohlig-Hellman utilise le fait que l'ordre du groupe $\mathbb{G}$ soit composé. En effet, si on écrit la décomposition de l'ordre n de $\mathbb{G}$

$$n = p_1^{e_1} \cdot p_2^{e_2} \dots p_k^{e_k}$$

alors l'algorithme de Pohlig-Hellman décompose le logarithme discret dans chacun des sous-groupes premiers de $\mathbb{G}$. L'attaque a donc pour complexité $O(\sum_{i=1}^{k} e_i \log(n + \sqrt{p_i}))$.

Number Field Sieve (NFS) est une attaque spécifique, de type calcul d'indices, qui est particulièrement efficace pour certains corps finis $\mathbb{F}_q$ avec $q = p^k$ et tels que p est asymptotiquement prépondérant devant $L(\frac{2}{3}, 1)$, dans ce cas les algorithmes NFS ont une complexité de $L_q(\frac{1}{3}, \sqrt[3]{\frac{64}{9}})$ [JL11]. Par conséquent, une attaque NFS est *a priori* asymptotiquement plus efficace qu'une attaque générique, plus précisément ces attaques dépendent de la taille total du corps fini et non pas de la taille du groupe multiplicatif.

ECDLP : DLP dans des groupes de courbe elliptique L'un des avantages des courbes elliptiques est que le DLP dans des groupes de courbe elliptique, est considéré comme plus dur que dans des sous-groupes de $\mathbb{F}_q^*$. Cela nous permet d'utiliser, pour un même niveau de sécurité des paramètres de plus petite taille par rapport au DLP dans des sous-groupes de $\mathbb{F}_q^*$. Concrètement, pour une sécurité de 128 bits, on peut utiliser des groupes sur courbes elliptiques de taille 256 bits contre au moins 3072 bits pour les sous-groupes de $\mathbb{F}_q^*$.

1.3 Couplages

1.3.1 Couplages et groupe bilinéaire

Il existe une application entre les groupes de points d'une courbe elliptique et les groupes multiplicatifs d'un corps fini, cette application s'appelle le couplage. Historiquement, les

couplages de points sur une courbe ont été introduits par André Weil à la fin des années 40. Et c'est en 1991 que Menezes, Vanstone et Okamoto [MVO91] les ont utilisés en cryptanalyse pour calculer des logarithmes discrets sur des courbes elliptiques. Après que les couplages aient été considérés comme un outil destructeur, Joux donna en 2000 [Jou00] un exemple que les couplages peuvent être utilisés aussi de façon constructive, notamment en élaborant un protocole d'échange de clé tripartite, ce qui lui a valu le prix Gödel en 2013. D'autres protocoles ont été élaborés ensuite comme le chiffrement basée sur l'identité [BF01], les signatures courtes [BLS01] *etc.*

Le but de cette section est de définir les couplages ainsi que leurs propriétés.

Définition (Couplage)**.** Soient $\mathbb{G}_1$, $\mathbb{G}_2$ et $\mathbb{G}_T$ trois groupes d'ordre premier ℓ. Un couplage est une application

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$$

qui vérifie les propriétés suivantes :

Bilinéarité : $\forall g_1 \in \mathbb{G}_1, \forall g_2 \in \mathbb{G}_2, \forall a, b \in \mathbb{Z}/\ell\mathbb{Z}$,

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} = e(g_1^b, g_2^a)$$

Non dégénérescence : $\forall g_1 \in \mathbb{G}_1, \forall g_2 \in \mathbb{G}_2$,

$$e(g_1, g_2) = 1_{\mathbb{G}_T} \Rightarrow (g_1 = 1_{G_1} \text{ ou } g_2 = 1_{\mathbb{G}_2}).$$

Dans le contexte cryptographique, on peut ajouter les propriétés de *efficacement calculable* (c'est-à-dire calculable par un algorithme polynomial) et *difficilement inversible*.

Depuis le début de l'utilisation des couplages, un bon nombre d'entre eux ont été conçus, les plus connus étant ceux de Weil et Tate.

Différents types de couplage. En pratique, les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ sont construits à partir d'une courbe elliptique E alors que $\mathbb{G}_T$ est un sous-groupe d'un corps fini. On note les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ additivement et le groupe $\mathbb{G}_T$ multiplicativement. Pour certaines propriétés de $\mathbb{G}_1$, $\mathbb{G}_2$ et $\mathbb{G}_T$, le choix de la courbe est trés important. Ce choix détermine notamment l'existence d'isomorphisme (facilement calculable) entre ces groupes. Il existe 3 types de couplage qui déterminent les groupes bilinéaires $\mathbb{G}_1, \mathbb{G}_2$ et $\mathbb{G}_T$ [GPS08]

Type 1 **:** Les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ sont construits à partir d'une courbe elliptique supersingulière et il existe deux isomorphismes $\psi_1 : \mathbb{G}_1 \to \mathbb{G}_2$ et $\psi_2 : \mathbb{G}_2 \to \mathbb{G}_1$ calculables de manière efficace.

Soit E une courbe elliptique définie sur $\mathbb{F}_q$ et ℓ un premier ne divisant pas q. Pour les autres types de couplages, $\mathbb{G}_1$ est définie comme le sous groupe du corps de base.

Type 2 **:** Le groupe $\mathbb{G}_2$ est définie comme n'importe quel sous-groupe d'ordre ℓ de $E[\ell]$ sauf le sous groupe du corps de base et le sous-groupe de trace nulle. De plus, l'endomorphisme Tr permet d'envoyer efficacement un point de $\mathbb{G}_2$ dans $\mathbb{G}_1$.

Type 3 **:** Le groupe $\mathbb{G}_2$ est définie comme le sous-groupe de trace nulle et il n'existe aucun isomorphisme entre $\mathbb{G}_1$ et $\mathbb{G}_2$ calculable de manière efficace.

Le choix du type est important dans la construction de systèmes cryptographiques basés sur les couplages. En effet, l'existence d'isomorphismes efficacement calculables entre $\mathbb{G}_1$ et $\mathbb{G}_2$ donne plus de possibilité à un adversaire d'attaquer ces constructions. Des

problèmes jugés difficiles dans les groupes utilisés par les couplages de types 2 ou 3 se révèlent faciles dans les groupes utilisés par les couplages de type 1 [Cos12]. Comme la plupart des protocoles cryptographiques basée sur les couplages et pour des raisons d'efficacité [GPS08], on ne considérera dans la suite que des couplages de type 3, ainsi on définit $\mathbb{G}_1$ et $\mathbb{G}_2$ comme :

$$\mathbb{G}_1 = E[\ell] \cap Ker(\pi - [1])$$
$$\mathbb{G}_2 = E[\ell] \cap Ker(\pi - [q])$$

Utilisation des twists Notons que les éléments de $\mathbb{G}_2$ sont des éléments de $E(\mathbb{F}_{q^k})[\ell]$. Par conséquent, les calculs dans $\mathbb{G}_2$ utilisent des opérations dans le corps $\mathbb{F}_{q^k}$. Supposons que E admette une twist de degré d et posons $m = pgcd(k, d)$ et $e = \frac{k}{m}$. Puisque k est le plus petit entier tel que ℓ divise $q^k - 1$, on sait que $\#E(\mathbb{F}_{q^e})$ est divisible par ℓ mais pas par ℓ^2. On sait donc qu'il existe une unique twist E' de degré m tel que $E'(\mathbb{F}_{q^e})$ est divisible par ℓ. (voir [HSV06] pour plus de détails). Ainsi, à l'aide de twist, on peut représenter les éléments de $\mathbb{G}_2$ sur un sous-corps de $\mathbb{F}_{q^k}$, à savoir $\mathbb{F}_{q^e}$. Les calculs sur $\mathbb{G}_2$ sont donc moins coûteux en utilisant l'isomorphisme Ψ entre E et E'. De plus, cet isomorphisme envoie tous les éléments du sous-groupe de trace nulle défini sur $\mathbb{F}_{q^k}$ vers le sous-groupe du corps de base défini sur $\mathbb{F}_{q^e}$. Pour résumé, si $\Psi : E' \to E$ alors

$$\Psi^{-1}(E(\mathbb{F}_{q^k})[\ell] \cap \mathrm{Ker}(\pi - [q])) = E(\mathbb{F}_{q^e})[\ell] \cap Ker(\pi - [1])$$

Pour ce qui est de $\mathbb{G}_T$ il est pris comme étant le groupe des racines ℓ-ème de l'unité dans $\mathbb{F}_{q^k}$. Autrement dit,

$$\mathbb{G}_T = \mu_\ell = \{\zeta \in \mathbb{F}_{q^k} : \zeta^\ell = 1\}$$

1.3.2 Calcul du couplage

Après avoir fait ce pourtour des groupes bilinéaires que les couplages utilisent, il reste à présenter les fonctions bilinéaires, non-dégénérée qui feront le calcul du couplage.

Définition (Couplage de Weil)**.** Soient E une courbe elliptique definie sur $\mathbb{F}_q$ et ℓ un entier premier ne divisant pas q et divisant $\#E(\mathbb{F}_q.)$ Soit k le degré de plongement de q par rapport à ℓ. Soient P et Q deux point de $E(\mathbb{F}_{q^k})[\ell]$. Soient D_P et D_Q deux diviseurs de support disjoint tels que,

$$D_P \sim (P) - (\infty) \text{ et } D_Q \sim (Q) - (\infty)$$

On sait alors que sous ses conditions, il existe des fonctions f et g telles que,

$$(f) = \ell D_P \text{ et } (g) = \ell D_Q$$

On définit alors le couplage de *Weil* comme étant l'application bilinéaire et non dégénérée suivante :

$$\begin{aligned} w_\ell : E(\mathbb{F}_{q^k})[\ell] \times E(\mathbb{F}_{q^k})[\ell] &\longrightarrow \mu_\ell \subset \mathbb{F}_{q^k} \\ (P, Q) &\longmapsto \frac{f(D_Q)}{g(D_P)} \end{aligned}$$

C'est le théorème suivant qui fait fonctionner la bilinéarité de ce couplage.

Théorème (Réciprocité de Weil)**.** *Soient f et g deux fonctions sur une courbe telles que (f) et (g) soient disjoints. Alors*

$$f((g)) = g((f))$$

Définition (Couplage de Tate)**.** Soient E une courbe elliptique definie sur $\mathbb{F}_q$ et ℓ un entier premier divisant $\#E(\mathbb{F}_q)$. Soit k le degré de plongement de q par rapport à ℓ. Soient P et Q deux points de la courbe tels que $P \in E(\mathbb{F}_{q^k})[\ell]$ et $Q \in E(\mathbb{F}_{q^k})$. Soient D_P et D_Q deux diviseurs de supports disjoint tels que,

$$D_P \sim (P) - (\infty) \text{ et } D_Q \sim (Q) - (\infty)$$

Soit f une fonction sur E de diviseur $(f) = \ell D_P$ disjoint de D_Q. Le couplage de *Tate* d'ordre ℓ est l'application suivante :

$$\begin{aligned} t_\ell : E(\mathbb{F}_{q^k}[\ell] \times E(\mathbb{F}_{q^k})[\ell] &\longrightarrow \mu_\ell \subset \mathbb{F}_{q^k} \\ (P, Q) &\longmapsto f(D_Q)^{\frac{q^k-1}{\ell}} \end{aligned}$$

Définition (Fonction de Miller)**.** Soit E une courbe elliptique définie sur $\mathbb{F}_q$. Soient P un point de cette courbe elliptique et i un entier. La *fonction de Miller* est une fonction $f_{i,P}$ telle que

$$(f_{i,P}) = i(P) - ([i]P) - (i-1)(\infty)$$

Pour calculer les fonctions f de diviseur ℓD_P dans les couplages de Weil et Tate, on utilise la *boucle de Miller* décrit dans l'algorithme 1. Cette algorithme utilise le fait que la fonction de Miller satisfait

$$f_{a+b,P}(Q) = f_{a,P}(Q) \cdot f_{b,P}(Q) \cdot \frac{g_{[a]P,[b]P}(Q)}{g_{[a+b]P}(Q)}$$

où les fonctions $g_{[a]P,[b]P}$ et $g_{[a+b]P}$ satisfont

$$\begin{aligned} (g_{[a]P,[b]P}) &= ([a]P) + ([b]P) + (-[a+b]P) - 3(\infty) \\ (g_{[a+b]P}) &= ([a+b]P) + (-[a+b]P) - 2(\infty) \end{aligned}$$

En particulier,
— la fonction $g_{[a]P,[b]P}$ est la droite qui passe par les points $[a]P$ et $[b]P$ (la tangente lorsque $a = b$)
— la fonction $g_{[a+b]P}$ est la droite verticale passant par le point $[a+b]P$

Algorithme 1 Boucle de Miller

Entrée : $\ell = (\ell_{n-1} \dots \ell_1 \ell_0)_2$ avec $\ell_{n-1} = 1$, $P \in E(\mathbb{F}_{q^k})[\ell]$, $D_Q \sim (Q) - (\infty)$ avec $Q \in E(\mathbb{F}_{q^k})[\ell]$

Sortie : $f = f_{\ell,P}(D_Q) \in \mathbb{F}_{q^k}$

$R \leftarrow P$ **et** $f \leftarrow 1$

pour i **allant de** $n-2$ **à** 0 **faire**

Calculer les droites $g_{R,R}$ et $g_{[2]R}$

$R \leftarrow [2]R$

$f \leftarrow f^2 \frac{g_{R,R}}{g_{[2]R}}(D_Q)$

si $\ell_i = 1$ **alors**

Calculer les droites $g_{P,R}$ et g_{P+R}

$R \leftarrow P + R$

$f \leftarrow f \frac{g_{P,R}}{g_{P+R}}(D_Q)$

fin si

fin pour retourner $f_{\ell,P}(D_Q)$

CHAPITRE 2

Construction de courbe elliptique à couplage

Les courbes elliptiques à couplages sont spéciales car elles visent des objectifs bien particuliers. C'est pour cela que toute une théorie s'est formée autour de leurs constructions. Ce chapitre a pour but de faire un rapide tour d'horizon des différentes constructions existantes pour présenter celle étudiée et utilisée durant ce stage.

2.1 Courbes elliptiques pairing-friendly

On a vu dans la section précédente qu'étant donnée une courbe elliptique E définie sur $\mathbb{F}_q$ les couplages ont comme domaine de définition $E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell]$ et comme domaine d'arrivée $\mathbb{F}_{q^k}*$. Par conséquent pour qu'un schéma cryptographique utilisant des couplages soit sécurisé, le problème du logarithme discret (voir 1.2.2) dans ces groupes doit être calculatoirement infaisable.

D'un autre côté, pour que le calcul du couplage soit efficace, il faut aussi que le degré de plongement k soit suffisamment petit. Il a été démontré que si nous prenons une courbe elliptique aléatoire E alors $k \sim \ell$ avec une forte probabilité [BK98].

Ainsi, notre problème est donc de trouver une courbe elliptique qui a un sous-groupe d'ordre premier ℓ suffisamment grand et de degré de plongement k assez petit. Les courbes qui satisfont ces conditions sont appelées des courbes appropriées aux couplages ou encore *pairing-friendly*. Le but de cette section est de présenter brièvement ces courbes. Tous les résultats de cette section sont pris de la taxonomie de Freeman, Scott et Teske [FST10].

Définition (Courbe pairing-friendly). Soit E une courbe elliptique définie sur un corps fini $\mathbb{F}_q$. On dit que E est une courbe *pairing-friendly* si elle satisfait les deux conditions suivantes :

1. il existe un premier $\ell \geq \sqrt{q}$ qui divise $\#E(\mathbb{F}_q)$
2. le dégrée de plongement k de E par rapport à ℓ est plus petit que $\log_2(\ell)/8$.

Nous introduisons la valeur-ρ souvent utilisée et définie par

$$\rho := \frac{\log_2(q)}{\log_2(\ell)}$$

ou

$$\rho := \frac{\deg(q)}{\deg(\ell)}$$

lorsque q et ℓ sont des polynômes de $\mathbb{Q}[X]$. La première condition dans la définition des courbes pairing-friendly est alors équivalente à $\rho \leq 2$. Mais cette valeur-ρ indique également l'efficacité de la représentation des éléments de $E(\mathbb{F}_q)[\ell]$ et le cas idéal est le cas où $\rho = 1$ car cela signifie que les éléments de $E(\mathbb{F}_q)[\ell]$ sont représentés en mémoire en $O(\log_2(\ell))$ bits.

2.1.1 Sécurité des courbes pairing-friendly

On a vu dans la Section 1.2.2 que l'un des avantages des courbes elliptiques était que le problème du logarithme discret (ECDLP) y est considéré comme le plus difficile. En effet, nous ne connaissons que des algorithmes dit génériques pour résoudre le problème du logarithme discret dans ces groupes [Sho97] tels que l'attaque du ρ de Pollard qui calcule le logarithme discret en au plus $\sqrt{\frac{\pi\ell}{4}}$ étapes [BLS11] De ce fait, si on veut une sécurité de λ bits, il suffit de prendre ℓ avec 2λ bits. Déterminer la taille de $\mathbb{G}_1$ et $\mathbb{G}_2$ est alors facile, mais qu'en est-il de $\mathbb{G}_T$, défini lui dans $\mathbb{F}_{q^k}$?

Pendant très longtemps, la difficulté du problème du logarithme discret dans $\mathbb{F}_{q^k}$ dépendait seulement de la taille de ce corps, à savoir $k\log_2(q)$. En se basant sur cette hypothèse, il était facile de choisir la taille de ℓ, q et k si nous désirions un niveau de sécurité λ. Par exemple, dans le cas où $\lambda = 128$, il est recommandé par le NIST [Bar20, Table 2, page 54] de prendre $k\log_2(q) = 3072$. Ainsi, les paramètres suivants semblaient être optimaux :

$$q \sim \ell \sim 2^{256} \text{ et } k = 12.$$

Malheureusement, l'attaque de Kim et Barbulescu en 2016 [KB16], qui est une variante NFS, a confirmé qu'il ne fallait plus se baser sur la taille de $\mathbb{F}_{q^k}$ pour estimer la difficulté de résoudre le logarithme discret dans ce corps. De ce fait, l'efficacité des calculs dans les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ est aussi impactés. Concrètement, il est maintenant nécessaire d'augmenter la taille de q pour rester compatible avec certaines valeurs k permettant un calcul du couplage efficace. De façon plus précise et dans le cas de figure où le corps $\mathbb{F}_{q^k}$ contient l'image d'un couplage, ces algorithmes NFS ont tous pour complexité

$$L_{q^k}\left[\frac{1}{3}, c\right] \text{ avec } c \in \left[\sqrt[3]{\frac{32}{9}}, \sqrt[3]{\frac{96}{9}}\right]$$

Ce n'est alors qu'en 2019 que Barbulescu et Duquesne [BD19] ont appliqué ces nouvelles attaques aux couplages. Ils ont ainsi proposé de nouvelles graines pour des courbes à couplages déjà standardisées permettant à ces dernières d'être résistantes à ces attaques. Typiquement, pour une sécurité de 128 bits, suivant la variante NFS utilisée (*e.g.* exTNFS, SexTNFS) pour calculer les logarithmes discrets, les recommandations de Barbulescu et Duquesne sont résumées dans le tableau suivant.

Variante NFS.	$k \log_2(q)$
NFS	2930
exTNFS	3618
SexTNFS	5004

2.2 Multiplication Complexe

Soit E une courbe elliptique. Lorsque l'anneau d'endomorphisme $\mathrm{End}(E)$ est strictement plus grand que $\mathbb{Z}$ alors on dit que E est à *multiplication complexe* (CM en anglais). De plus, lorsque E est une courbe elliptique ordinaire définie sur un corps de caractéristique non nulle, cet anneau est isomorphe à un ordre dans un corps de nombres quadratiques imaginaires. Donc, pour un certain corps de nombres quadratiques imaginaires $K = \mathbb{Q}(\sqrt{-D})$ avec $D > 0$ sans facteur carré, il existe un ordre $\mathcal{O}$ de K tel que

$$\mathrm{End}(E) \simeq \mathcal{O}.$$

On sait que l'endomorphisme de Frobenius π est dans $\mathrm{End}(E)$ et $\mathcal{O}$ est un sous-anneau de $\mathcal{O}_K = \mathbb{Z}[\delta]$ où

$$\delta = \begin{cases} \frac{1+\sqrt{-D}}{2} & \text{si } D \equiv 3 \mod 4 \\ \sqrt{-D} & \text{si } D \equiv 1, 2 \mod 4 \end{cases}$$

donc par minimalité de l'indice $[\mathcal{O}_K : \mathcal{O}]$, on peut trouver un entier f tel que π et $f\delta$ engendre le même ordre. Ainsi on a

$$\mathrm{End}(E) \simeq \mathbb{Z} + f\delta\mathbb{Z},$$

et donc le discriminant du polynôme minimal du Frobenius à savoir $t^2 - 4q$ est égal au discriminant de l'ordre à savoir $-f^2D$ ou $-4f^2D$. Soit

$$y = \begin{cases} f & \text{si } D \equiv 3 \mod 4, \\ 2f & \text{si } D \equiv 1, 2 \mod 4. \end{cases}$$

On définit *l'équation CM* comme étant

$$-y^2D = 4q - t^2. \tag{2.1}$$

2.3 Méthodes de construction

Dans la taxonomie [FST10], les auteurs ont classifié les courbes pairing-friendly en deux types selon la méthode de construction utilisée pour générer ces dernières :

Courbes individuelles : ces courbes sont générées par des méthodes qui donnent des entiers q, ℓ, t et k tels qu'il existe une courbe E définie sur $\mathbb{F}_q$ dont la trace du Frobenius est t, avec un sous-groupe d'ordre ℓ et un degré de plongement k par rapport à ℓ.

Familles paramétriques de courbes : ces courbes sont générées par des méthodes qui donnent des polynômes $q(x), \ell(x), t(x)$ et un entier k tels que si $q(x_0)$ est premier pour une certaine graine $x_0 \in \mathbb{Z}$ alors il existe une courbe E définie sur $\mathbb{F}_{q(x_0)}$ dont la trace du Frobenius est $t(x_0)$, avec un sous-groupe d'ordre $\ell(x_0)$ et un degré de plongement k par rapport à $\ell(x_0)$.

Les courbes *supersingulières* sont des courbes individuelles, l'inconvénient de ces courbes est que $k = 2$ et les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ sont des groupes de couplage de type 1. Il existe aussi deux autres constructions dans la littérature qui génèrent des courbes individuelles, à savoir les constructions *Cocks-Pinch* [CP01] et *Dupont-Enge-Morain* [DEM05].

Si le discriminant D est suffisamment petit alors certaines constructions utilisent l'équation CM 2.1 pour trouver l'équation de la courbe. C'est le cas de la plupart des familles paramétriques de courbes. Et comme ces courbes sont définies par des polynômes $q(x), \ell(x), t(x) \in \mathbb{Q}[x]$, l'équation CM devient alors,

$$y^2 D = 4q(x) - t(x)^2$$

Ainsi lorsqu'on souhaite résoudre cette équation on fait face à deux situations :
— Il existe un ensemble infini de couple (x, y) qui sont solutions de cette équation, on dit alors que cette famille de courbes est *creuse* ;
— Il est possible de paramétrer y en fonction de x, on dit alors que cette famille de courbes est *complète*.

Parmi les familles creuses, on trouve la construction *Miyaji-Nakabayashi-Takano* [MNT00] qui construit des courbes d'ordre premier et qui ont un degré de plongement $k \leq 10$. Pour les familles complètes, il est possible de construire des courbes de n'importe quel degré de plongement k et généralement avec une valeur-ρ strictement plus grande que 1. On distingue trois sous-familles de familles complètes :

Famille cyclotomique : ℓ est un polynôme cyclotomique et K est un corps cyclotomique qui contient D.

Famille sporadique : ℓ n'est pas cyclotomique et K est une extension finie d'un corps cyclotomique qui contient D.

Famille de Scott-Barreto : ℓ n'est pas cyclotomique et K est une extension finie d'un corps cyclotomique qui ne contient pas D.

Pour résumer toutes ces méthodes de construction, nous reprenons l'arbre de classification de [FST10] en Figure 2.1.

2.4 Construction étudiée

Nous avons vu dans la section précédente qu'il existe de nombreuses méthodes dans la littérature pour construire des courbes « pairing-friendly » et plus particulièrement celles qui nous intéressent, *i.e.* courbes ordinaires. L'idée derrière ces constructions est la même :
— Fixer k et calculer des entiers q, ℓ et t tels qu'il existe une courbe elliptique E définie sur $\mathbb{F}_q$ dont la trace du Frobenius est t, le nombre de points sur cette courbe est divisible par un premier ℓ et E est de degré de plongement k.
— Utiliser l'équation CM pour trouver l'équation de la courbe sur $\mathbb{F}_q$ à la manière d'Atkin et Morain [AM93].

Une courbe elliptique ordinaire peut être construite si et seulement si les conditions suivantes sont satisfaites [FST10] :

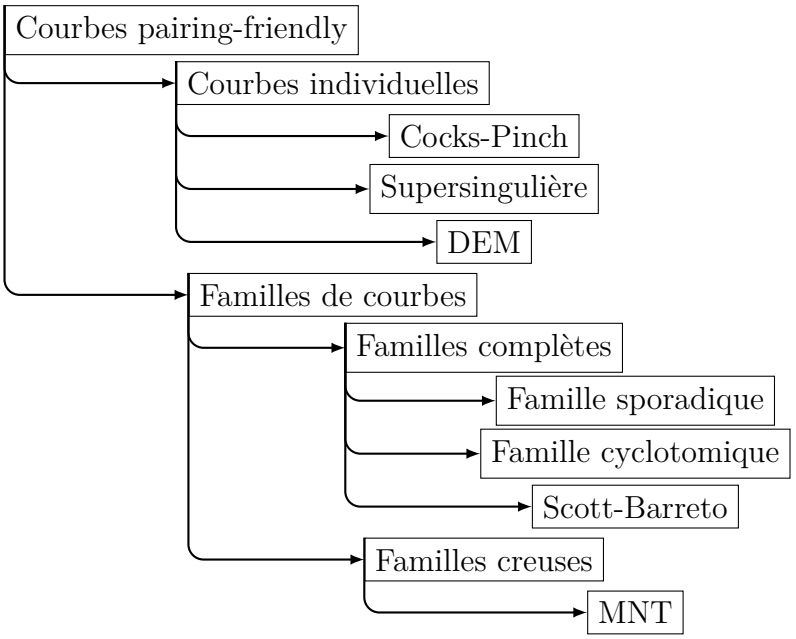

FIGURE 2.1 – Arbre de classification des courbes pairing-friendly

1. q est premier
2. ℓ est premier
3. $\text{pgcd}(t, q) = 1$
4. ℓ divise $q + 1 - t$
5. ℓ divise $q^k - 1$ et pas $q^i - 1$ pour $i < k$
6. $4q - t^2 = -Dy^2$ pour un certain entier positif D suffisamment petit et un certain entier y

En effet, la condition 1 assure l'existence d'un corps fini. Les conditions 2 et 4 combinées indiquent que $E(\mathbb{F}_q)$ a un sous-groupe d'ordre premier ℓ. La condition 3 assure que la courbe elliptique est ordinaire. La condition 5 implique que E a un degré de plongement k par rapport à ℓ. Et enfin, la condition 6 implique $|t| \leq 2\sqrt{q}$. Toutes ces conditions constituent le fil rouge des constructions générant les courbes qui nous intéressent et en particulier la construction de Brezing et Weng.

2.4.1 Construction de Brezing-Weng

Dans cette sous-section, nous étudions une construction particulière des familles cyclotomiques à savoir la construction Brezing-Weng. Elle permet de construire des familles de courbes elliptiques avec un degré de plongement choisi. Soit E une courbe elliptique ordinaire définie sur $\mathbb{F}_q$ de degré de plongement k par rapport à ℓ premier. Alors

$$\begin{cases} \#E(\mathbb{F}_q) = q + 1 - t & \equiv 0 \mod \ell, \\ q^k & \equiv 1 \mod \ell. \end{cases}$$

En combinant les deux équations, on obtient

$$(t-1)^k \equiv 1 \mod \ell.$$

Ainsi, par minimalité de k, $t-1$ est une racine primitive k-ème de l'unité modulo ℓ. La condition 5 présentée précédemment peut être donc remplacée par

$$\ell \text{ divise } \Phi_k(t-1).$$

Regardons maintenant l'anneau d'endomorphismes $\mathrm{End}(E)$. On a vu qu'il était isomorphe à un ordre $\mathcal{O}$ dans un corps de nombres quadratiques imaginaires $K = \mathbb{Q}(\sqrt{-D})$ avec D un entier positif sans facteur carré. On a vu également que le polynôme minimal du Frobenius π est $X^2 - tX + q$, donc π correspond à un élément ω tel que

$$\omega + \overline{\omega} = t \quad \text{et} \quad \omega\overline{\omega} = q,$$

où $\overline{\omega}$ est le conjugué de ω. On en déduit qu'il existe un entier b tel que

$$\omega = \frac{t + b\sqrt{-D}}{2}$$

Déterminons cet entier b, on sait que $q + 1 - t \equiv 0 \mod \ell$, donc en développant

$$\begin{aligned} \omega\overline{\omega} - t + 1 &= \left(\frac{t + b\sqrt{-D}}{2}\right)\left(\frac{t - b\sqrt{-D}}{2}\right) - t + 1, \\ &= \frac{(t-2)^2 + b^2 D}{4}, \end{aligned}$$

et en notant δ une racine carrée de $-D \mod \ell$, on obtient la congruence suivante

$$b \equiv \pm\frac{t-2}{\delta} \mod \ell$$

À partir de ces observations, on peut déduire un algorithme très simple attribué à Cocks-Pinch [CP01], pour trouver les entiers q, ℓ et t :

Algorithme 2 Algorithme Cocks-Pinch

Entrée : k le degré de plongement, D un entier positif sans facteur carré
Sortie : q, ℓ, t

Prendre ℓ satisfaisant :
- $\ell \equiv 1 \mod k$ ▷ Assure l'existence de ζ_k
- il existe δ tel que $\delta^2 \equiv -D \mod \ell$

$a \leftarrow \zeta_k + 1 \mod \ell$
$b \leftarrow \pm\dfrac{a-2}{\delta} \mod \ell$
$\omega \leftarrow \dfrac{a + b\sqrt{-D}}{2}$
$q \leftarrow \omega\overline{\omega}$
$t \leftarrow a$
si q est premier ou puissance d'un nombre premier **alors**
  retourner q, ℓ, t
fin si

On utilise alors l'équation CM pour déterminer l'équation de la courbe. La construction de Brezing-Weng [BW05] utilise la même approche mais en construisant les entiers q, ℓ et t comme images de polynômes de $\mathbb{Q}[X]$. Pour cela, on construit le corps de nombres

$$M = \mathbb{Q}(\zeta_n, \sqrt{-D})$$

avec $n \equiv 0 \mod k$. On suppose donc que

$$M \simeq \mathbb{Q}[X]/\langle f(X)\rangle$$

où $f(X) \in \mathbb{Q}[X]$ est un polynôme irréductible de degré $\varphi(n)$ si $\sqrt{-D} \in \mathbb{Q}(\zeta_n)$ et $2\varphi(n)$ sinon. Nous savons qu'il existe $\varphi(k)$ racines k-ème primitives de l'unité dans M, notons alors $g_i(X)$ pour $1 \leq i \leq \varphi(k)$ la classe d'équivalence de chacune de ces racines dans M et h_j pour $1 \leq j \leq 2$ la classe d'équivalence des racines carrées de $-D$ dans M. Supposons ainsi qu'il existe $i \in \{1, \ldots, \varphi(k)\}$ et $j \in \{1, 2\}$ tels que $g_i(X)$ et $h_j(X)$ soient à coefficients entiers. De la même façon, nous définissons les polynômes $a(X)$ et $b(X)$ représentant l'élément

$$\omega = \frac{a + b\sqrt{-D}}{2}$$

Si nous suivons la structure de l'Algorithme 2, nous poserions

$$\begin{aligned} a(X) = &\quad g_i(X) + 1 \mod f(X) \\ b(X) = &\quad (a(X) - 2)h_j^{-1}(X) \mod f(X) \end{aligned}$$

Or, pour que $b(X)$ soit à coefficients entiers il faut que l'inverse de $h_j(X) \mod f(X)$ soit également à coefficients entiers, ce qu'on ne sait pas *a priori*. Pour remédier à cela, l'astuce est de prendre à la place de $b(X)$, le polynôme représentant $(a-2)\delta$, à savoir

$$\hat{b}(X) = (a(X) - 2)h_j(X) \mod f(X)$$

ce qui est possible car $Db^2 = \dfrac{\hat{b}^2}{D}$ dans le calcul de la norme de ω que nous notons $q(X)$

$$q(X) = \frac{1}{4}\left(a(X)^2 + \frac{\hat{b}(X)^2}{D}\right)$$

On cherche ensuite une classe d'équivalence $x_0 \in \mathbb{Z}/D\mathbb{Z}$ telle que

$$\begin{cases} \hat{b}(x_0) \equiv 0 \mod D \\ f(DX + x_0) \text{ irréductible} \end{cases}$$

En effet, supposons que $q(X) \in \mathbb{Q}[X]$ soit irréductible. Pour que ce polynôme prenne des valeurs premières il faut au minimum que le quotient $\hat{b}(X)^2/D$ prenne des valeurs entières, ce qui est le cas pour tout entier divisible par D, d'où la première condition. De plus, les polynômes $a(X)$ et $\hat{b}(X)$ étant définis modulo $f(X)$, on souhaite que le premier ℓ soit image de $f(X)$, pour cela il faut que $f(DX + x_0) \in \mathbb{Z}[X]$ soit irréductible car tout entier dans la classe d'équivalence x_0 s'écrit $Dm + x_0$ avec $m \in \mathbb{Z}$, d'où la deuxième condition. On cherche alors un entier $x_1 \equiv x_0 \mod D$ tel que $q(x_1)$ et $f(x_1)$ soient premiers. Si on trouve une telle *graine* x_1, il existe alors une courbe elliptique sur $\mathbb{F}_{q(x_1)}$ dont l'anneau d'endomorphismes est isomorphe à un ordre $\mathcal{O}$ de $\mathbb{Q}(\sqrt{-D})$. Le Frobenius correspond alors à l'élément

$$\omega_{x_1} = \frac{1}{2}\left(a(x_1) \pm \frac{\hat{b}(x_1)}{D}\sqrt{-D}\right)$$

dont la trace est $a(x_1)$ et par construction

$$\#E(\mathbb{F}_{q(x_1)}) = \omega_{x_1}\overline{\omega_{x_1}} - a(x_1) + 1 = \frac{1}{4}\left((a(x_1)-2)^2 + \frac{\hat{b}(x_1)}{D}\right) \equiv 0 \mod f(x_1)$$

De plus, les degrés de $a(X)$ et $b(X)$ étant inférieurs ou égaux à $\deg(f(X))-1$, la valeur-ρ tend vers $2 - \frac{2}{\deg(f(X))}$ lorsque ℓ tend vers l'infinie.

Le succès de cette construction dépend énormément du choix du corps de nombres M car il faut qu'il contienne ζ_k et $\sqrt{-D}$. Un résultat de la théorie des nombres présent dans [MF05] indique que :

Proposition. *Soit K le n-ème corps cyclotomique $\mathbb{Q}(\zeta_n)$ avec ζ_n une racine n-ème de l'unité. On a les implications suivantes*

- *si K contient $\sqrt{-1}$ alors $4|n$*
- *si K contient $\sqrt{-2}$ alors $8|n$*
- *K contient $\sqrt{sp}$ pour tout premier impair $p|n$ et $s = (-1)^{\frac{p-1}{2}}$*

Ainsi, dans la suite $D \in \{1,2,3\}$ et nous prenons n en fonction de D, *i.e.*

- si $D = 1$ alors $n = \dfrac{4k}{\text{pgcd}(4,k)}$
- si $D = 2$ alors $n = \dfrac{8k}{\text{pgcd}(8,k)}$
- si $D = 3$ alors $n = \dfrac{3k}{\text{pgcd}(3,k)}$

Remarque (Courbes Barreto-Lynn-Scott). Parmi les courbes que peut générer la méthode de Brezing-Weng, il y a les courbes de Barreto-Lynn-Scott avec comme particularité que le degré de plongement k est de la forme $2^i 3^j$.

CHAPITRE 3

Réalisation et contribution

Ce stage s'est déroulé en plusieurs étapes, la première étant l'appropriation des connaissances nécessaires à la compréhension des courbes elliptiques à couplage ainsi que de leurs constructions présentes dans la taxonomie de Freeman, Scott et Teske [FST10]. C'était la partie théorique du stage. Le prochain objectif est de chercher les paramètres permettant de construire une de ces courbes, de l'implémenter et enfin de la comparer à d'autres courbes déjà standardisées.

3.1 Recherche de courbe

Dans le cadre de ce stage, le point de vue qui est adopté est différent de celui adopté dans la plupart de la littérature : on part des besoins de protocoles cryptographiques et on propose une courbe plus adaptée que celles déjà standardisées. Par exemple dans de nombreux schémas de signature de groupe utilisant les couplages, l'exponentiation dans $\mathbb{G}_1$ est plus utilisée que les autres opérations. De ce fait, Clarisse, Duquesne et Sanders ont proposé, dans [CDS20], une nouvelle courbe « optimisée » pour l'exponentiation dans $\mathbb{G}_1$ au détriment des autres opérations. C'est donc dans ce cadre là qu'ils ont suggéré de dégager d'autres familles de protocoles cryptographiques qui bénéficieraient de ce changement de point de vue. Notre objectif durant ce stage est donc d'obtenir une courbe elliptique à couplage d'une sécurité de 128 bits et équilibrant au mieux l'exponentiation dans $\mathbb{G}_1$ et $\mathbb{G}_2$, comparé aux courbes de la littérature.

Les méthodes de construction de courbes elliptiques à couplage avec une valeur-ρ proche de 1 sont les méthodes donnant l'ordre du corps fini q comme image d'un polynôme, on dit alors que q est *spécial* (si le degré de ce polynôme est au moins 3). Sachant cela, nous nous sommes naturellement tournés vers les constructions de familles paramétriques et plus particulièrement une reprise de la méthode de Brezing-Weng faite par Guillevic [Gui20, Algorithme 3.1]. Dans cet algorithme, Guillevic ajoute un paramètre e qui sert à choisir la racine k-ième de l'unité. Nous rappelons cet algorithme ici :

Algorithme 3 Construction Brezing-Weng

Entrée : k le degré de plongement, D le discriminant, e le choix de la racine
Sortie : $q(X), \ell(X), t(X)$

si $D = 1$ **alors**
  $m \leftarrow 4/\operatorname{pgcd}(4, k)$
sinon si $D = 2$ **alors**
  $m \leftarrow 8/\operatorname{pgcd}(8, k)$
sinon si $D = 3$ **alors**
  $m \leftarrow 3/\operatorname{pgcd}(3, k)$
sinon $m \leftarrow 1$
fin si
$\ell(X) \leftarrow \Phi_{km}(X)$
$K \leftarrow \mathbb{Q}[X]/(\ell(X))$
$\zeta_{km} \leftarrow$ une racine de $\ell(X)$ dans K
si $-D$ n'est pas un carré dans K **alors**
  retourner $\perp$
fin si
si $\operatorname{pgcd}(e, k) \neq 1$ **alors**
  retourner $\perp$
fin si
$t(X) \leftarrow X^{me} + 1 \mod \ell(X)$
$b(X) \leftarrow$ le polynôme représentant $t(\zeta_{km}) - 2)\sqrt{-D}/D$ dans K
$q(X) \leftarrow \dfrac{t(X)^2 + Db(X)^2}{4}$
si $q(X)$ n'est pas irréductible **alors**
  retourner $\perp$
fin si
retourner $q(X), \ell(X), t(X)$

De plus, la courbe construite doit être résistante aux nouveaux algorithmes de types NFS. C'est donc dans ce contexte que nous utiliserons l'algorithme fait par Guillevic [GS19] pour évaluer la sécurité de nos courbes, disponible à l'adresse

`https://gitlab.inria.fr/tnfs-alpha/alpha`

Il ne manque plus qu'à choisir les bons paramètres k, D et e pour que l'algorithme 3 nous génère des familles de courbes elliptiques.

3.1.1 Choix des paramètres

Afin de réduire le coût de l'exponentiation dans $\mathbb{G}_2$ nous nous focalisons sur les degrés de plongement composé. Plus précisément, nous souhaitons que la courbe trouvée E admette une twist E' de degré d pour avoir les points de $\mathbb{G}_2$ définis sur $\mathbb{F}_{p^s}$ avec $s = \frac{k}{d} \in \{4, 5, \ldots, 20\}$. De plus, les degrés d possibles des twists pour les courbes elliptiques ordinaires sont $d \in 3, 4, 6$. On se restreint donc aux k tels que

$$12 \leq k \leq 60 \text{ et } k \text{ divisible par } 3, 4 \text{ ou } 6.$$

On souhaite également que notre courbe soit définie sur un corps premier dont les éléments tiennent au plus sur 5 mots machines dans une architecture de 64 bits, donc

$$\log_2(q) \leq 320.$$

De plus, puisque nous souhaitons avoir un niveau de sécurité de 128 bits, l'ordre premier ℓ des groupes $\mathbb{G}_1, \mathbb{G}_2$ et $\mathbb{G}_T$ doit satisfaire

$$\log_2(\ell) \geq 256.$$

Les *graines* $x_0 \in \mathbb{Z}$ qui nous donneront nos entiers $q = q(x_0)$ et $\ell = \ell(x_0)$, doivent donc vérifier

$$\begin{cases} \log_2(q(x_0)) & \leq 320 \\ \log_2(\ell(x_0)) & \geq 256 \end{cases}$$

Supposons que les polynômes $q(X)$ et $\ell(X)$ sont équivalents à leurs monômes de plus au degré (ce qui est pour un antécédent suffisamment grand), *i.e.*

$$q(X) \sim \mathrm{lc}(q)X^{\deg(q)}$$
$$\ell(X) \sim \mathrm{lc}(\ell)X^{\deg(q)}$$

où $\mathrm{lc}(\cdot)$ désigne le coefficient dominant du polynôme considéré. Nous obtenons, *in fine* l'encadrement suivant

$$\frac{256 - \mathrm{lc}(\ell)}{\deg(\ell)} \leq x_0 \leq \frac{320 - \mathrm{lc}(q)}{\deg(q)} \tag{3.1}$$

Remarque. Nous ne cherchons les graines x_0 que lorsque la borne de gauche est inférieure à la borne de droite.

Après avoir effectué un script `SageMath`, nous avons obtenu 29 triplets (k, D, e) permettant la construction de familles de courbes avec la méthode de Brezing-Weng. Chaque triplet donne lieu aux polynômes $q(X), \ell(X)$ et $t(X)$ grâce à l'algorithme 3. Nous avons alors implémenté une fonction en `Pari-GP` pour chercher les graines x_0 telles que $q(x_0)$ soit premier. Or $q(X)$ est à coefficients dans $\mathbb{Q}$ donc pour que $q(x_0)$ soit premier il faut déjà qu'il soit entier. L'astuce alors est de multiplier $q(X)$ par ν le ppcm des dénominateurs des coefficients de $q(X)$ afin d'obtenir le polynôme $\tilde{q}(X) = \nu q(X)$ qui lui est à coefficients entiers. Ainsi, pour toutes graines x_0 dans l'intervalle (3.1) on a l'implication suivante

$$\tilde{q}(x_0) \equiv 0 \mod \nu \Longrightarrow q(x_0) \in \mathbb{Z}$$

Il ne reste plus qu'à effectuer un test de primalité sur $q(x_0)$, si le test s'avère être positif alors on inscrit x_0 comme graines permettant la construction de notre courbe. Parmi les 29 triplets (k, D, e) possibles, seulement 10 d'entre eux admettent au moins une graine x_0 telle que $q(x_0)$ est un premier d'au plus 320 bits et $\ell(x_0)$ est un premier d'au moins 256 bits. Ces résultats sont résumés dans la table 3.1.

Remarque. Lorsque $k = 24$ et $k = 48$, la construction de Brezing-Weng donne des courbes BLS.

TABLE 3.1 – Courbes possibles

k	D	e	$\deg q$	$\deg \ell$	ρ	# de graines
24	3	1	10	8	1.25	> 25800
27	3	1	20	18	1.11	29
27	3	19	20	18	1.11	19
33	3	1	24	20	1.2	1
33	3	23	24	20	1.2	2
39	3	28	30	24	1.25	1
44	1	1	24	20	1.2	4
48	3	1	18	16	1.125	106
48	3	17	18	16	1.125	87
51	3	35	36	32	1.125	1

3.1.2 Courbe sélectionnée

Choix du triplet (k, D, e)

Afin de sélectionner la courbe la plus intéressantes parmi les 10 triplets (k, D, e) de la table 3.1, nous avons procéder par élimination. Pour $k = 33$, respectivement $k = 39$, $k = 44$ et $k = 51$, leur twist est définie sur une extension de $\mathbb{F}_q$ de degré 11, respectivement 13, 11 et 17, ce qui est assez grand. D'autant plus que ces degrés d'extensions sont tous des nombres premiers, donc l'efficacité de certaines opérations sur les corps finis en serait impactée. Pour $k = 48$ et $k = 27$, quand bien même pour certaines graines x_0, l'ordre $q(x_0)$ du corps fini tient sur 9 mots machines sur une architecture de 32 bits, leur twist est définie sur une extension de degré 8 et 9, respectivement. Nous avons donc préféré choisir la famille de courbe pour laquelle la twist est définie sur une extension de degré 4, à savoir la `BLS-24`.

Choix de la graine x_0 pour la `BLS-24`

La famille de courbes Barreto, Lynn et Scott avec $k = 24$ a été étudiée par Costello, Lauter et Naehrig [CLN11]. Afin d'augmenter l'efficacité et la sécurité de l'implémentation de ces courbes, les auteurs suggèrent des choix spécifiques pour ces courbes. Ils proposent également, à la fin de leur article, une liste de courbes BLS satisfaisant ces particularités et avec un niveau de sécurité allant de 192 bits à 320 bits.

Définition (Courbe Barreto-Lynn-Scott)**.** Les courbes de la famille BLS sont définies, lorsque le degré de plongement est 24, par les polynômes

$$\begin{aligned} q(X) &= \frac{1}{3}(X-1)^2(X^8 - X^4 + 1) + X \\ \ell(X) &= X^8 - X^4 + 1 \\ t(X) &= X + 1 \end{aligned}$$

Des équations $\#E(\mathbb{F}_q) = q + 1 - t$ et $3f^2 = 4q - t^2$, on déduit deux autres polynômes

$$n(X) = \frac{1}{3}(X-1)^2(X^8 - X^4 + 1)$$
$$f(X) = \frac{1}{3}(X-1)(2X^4 - 1)$$

Dans l'article [CLN11], les auteurs dégagent quatre sous-familles de `BLS-24`. Ils montrent que choisir E dans l'une de ces sous-familles impacte grandement l'efficacité de l'ensemble des opérations qu'implique un couplage. De plus, en choisissant une de ces sous-familles, les équations de la courbe E et de sa twist E' sont immédiatement déterminées. La twist E' est définie sur le corps $\mathbb{F}_{q^4}$ qu'on construit comme une extension de $\mathbb{F}_{q^2}$. Détaillons la construction de cette tour d'extension de corps avant de définir les quatre sous-familles.

Construction de $\mathbb{F}_{q^2}$: Pour construire $\mathbb{F}_{q^2}$, on « quotiente » l'anneau $\mathbb{F}_q[X]$ par l'idéal $\langle X^2+1\rangle$. Ce polynôme X^2+1 est irréductible si $q \equiv 3 \mod 4$. En notant u une des deux racines de ce polynôme :

$$\mathbb{F}_{q^2} = \mathbb{F}_q(u).$$

L'avantage de construire $\mathbb{F}_{q^2}$ ainsi est que les opérations dans $\mathbb{F}_q[X]/\langle X^2+1\rangle$ sont moins coûteuses que dans $\mathbb{F}_q[X]/\langle X^2 - \alpha\rangle$ pour $\alpha \neq \pm 1$ un résidu non-quadratique de $\mathbb{F}_q$.

Construction de $\mathbb{F}_{q^4}$: Considérons $\mathbb{F}_{q^2}$ construit comme précédemment. Nous construisons $\mathbb{F}_{q^4}$ comme une extension de degré 2 de $\mathbb{F}_{q^2}$. La manière la plus intuitive et la plus simple de le faire est de « quotienter » $\mathbb{F}_{q^2}[X]$ par l'idéal $\langle X^2 + u\rangle$. Or les polynômes de la forme $X^2 + su$ avec $s \in \mathbb{F}_q$ ne sont pas irréductibles dans $\mathbb{F}_{q^2}$. En effet, puisque -1 n'est pas un résidu quadratique dans $\mathbb{F}_q$, un seul des deux éléments $\pm s/2$ est un résidu quadratique dans $\mathbb{F}_q$. Si c'est $s/2$, il existe $a \in \mathbb{F}_q$ tel que $s = 2a^2$. Donc le polynôme $X^2 + su$ s'écrit

$$X^2 + 2a^2u = (X + au - a)(X - au + a).$$

Si c'est $-\frac{s}{2}$. Il existe alors $a \in \mathbb{F}_q$ tel que $s = -2a^2$. Donc le polynôme $X^2 + su$ s'écrit

$$X^2 - 2a^2u = (X - au - a)(X + au + a).$$

Donc dans les 2 cas, le polynôme X^2+su n'est pas irréductible sur $\mathbb{F}_{q^2}$. Il faut donc trouver un autre polynôme pour construire $\mathbb{F}_{p^4}$. Nous avons choisi le polynôme $X^2 - (u+1)$ qui, lui, est irréductible sur $\mathbb{F}_{q^2}$. En notant v une racine de ce polynôme, on a

$$\mathbb{F}_{q^4} = \mathbb{F}_{q^2}(v).$$

L'avantage de construire $\mathbb{F}_{q^4}$ ainsi est que les opérations sont plus efficace puisque la multiplication par $u+1$ dans $\mathbb{F}_{q^2}$ est peu coûteuse (voir [Dev+06])

Nous résumons la construction de nos corps $\mathbb{F}_{q^2}$ et $\mathbb{F}_{q^4}$ par la tour d'extension en Figure 3.1.

Le Tableau 3.2 définit les quatre sous-familles des courbes `BLS-24` proposées dans [CLN11].

Parmi toutes les graines que nous avions, seulement une poignée sont dans l'une de ces quatre sous-familles, voir le Tableau 3.3.

Pour effectuer notre sélection, nous avons ajouté deux critères :

$$\mathbb{F}_p \xrightarrow{u^2+1} \mathbb{F}_{p^2} \xrightarrow{v^2-(u+1)} \mathbb{F}_{p^4}$$

FIGURE 3.1 – Tour d'extension de corps fini avec les degrés d'extension en rouge

x_0 mod 72	$q(x_0)$ mod 72	$n(x_0)$ mod 72	E	E'
7	19	12	$y^2 = x^3 + 1$	$y^2 = x^3 \pm v^{-1}$
16	19	3	$y^2 = x^3 + 4$	$y^2 = x^3 \pm 4v$
31	43	12	$y^2 = x^3 + 1$	$y^2 = x^3 \pm v$
64	19	27	$y^2 = x^3 - 2$	$y^2 = x^3 \pm 2v^{-1}$

TABLE 3.2 – Les quatre sous-familles des courbes `BLS-24`

x_0	x_0 mod 72	$\#E(\mathbb{F}_q)$ mod 4	poids de $\ell(x_0)$	poids de x_0
4294971136	64	3	167	5
−4295102624	64	3	166	5
−4295495945	7	0	147	6
4295565856	64	3	177	6
4295623264	16	3	156	6
−4295753921	7	0	173	6
−4296016712	16	3	159	6
4296310792	16	3	170	5
−4296458240	64	3	159	6
4297328800	16	3	161	6

TABLE 3.3 – Nos graines et leurs critères

— pouvoir utiliser la forme d'Edwards sur $\mathbb{G}_2$ (voir [Ber+08] pour plus de détails). Pour remplir ce critère il suffit que l'ordre de la courbe soit divisible par 4.
— minimiser le nombre d'itérations dans la boucle de Miller (voir algorithme 1). Pour remplir ce critère il faut prendre ℓ et x_0 avec les plus petits poids de Hamming.

La graine choisie est donc

$$x_0 = -4295495945$$

Elle nous garantit d'avoir une courbe elliptique à couplage E définie sur un corps fini de 319 bits et un sous-groupe d'ordre ℓ de 257 bits.

3.2 Implémentation

Pour effectuer nos comparaisons entre notre courbe et d'autres courbes déjà standardisées, il faut l'implémenter. Pour cela, nous utilisons la librairie `relic-toolkit` [Ara+]

disponible à l'adresse : `https://github.com/relic-toolkit/relic`.

3.2.1 Librairie RELIC

La librairie `relic-toolkit` est une boîte à outils cryptographiques mettant l'accent sur l'efficacité et la flexibilité. Elle est écrite en C et peut être utilisée pour créer des outils cryptographiques efficaces et adaptables à des niveaux de sécurité différents. Les algorithmes implémentés dans cette librairie :

— Arithmétique d'entiers multi-précision ;
— Arithmétique des corps finis ;
— Courbes elliptiques définies sur les corps finis ;
— Couplages et extensions de corps qu'ils utilisent ;
— Protocoles cryptographique (RSA, Rabin, ECDSA, ECMQV, ECSS (Schnorr), ECIES, Échange de clé basée sur l'identité Sakai-Ohgishi-Kasahara , Signature courte de Boneh-Lynn-Schacham et Boneh-Boyen, Système de chiffrement homomorphe de Paillier et Benaloh).

La librairie `relic` est organisée en plusieurs dossiers et sous dossiers.

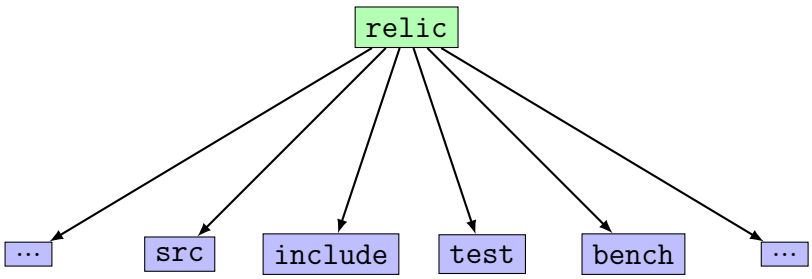

FIGURE 3.2 – Dossiers utilisés dans la librairie `relic`

Nous décrivons ci-dessous les différents répertoires utilisés pour implémenter notre courbe elliptique à couplage.

Répertoire `src`

Ce dossier contient l'ensemble des fichiers sources des algorithmes implémentés dans la librairie. Il se décompose en plusieurs sous-dossiers dont ceux qu'on utilise : `fp`, `fpx`, `ep`, `epx`. Détaillons chacun de ses sous-dossiers pour une meilleure compréhension de la librairie.

`fp` : `f` pour « finite field » et `p` pour « prime », ce sous-dossier contient l'ensemble des fichiers concernant les corps finis premiers et leur arithmétique. On y trouve, en particulier, le fichier `relic_fp_param.c` dans lequel figure la fonction `fp_param_get()` qui retourne la graine x_0. Cette fonction est utilisée conjointement avec la fonction `relic_fp_set_pairf()` présente dans le fichier `relic_fp_prime.c` et qui retourne le premier $q(x_0)$.

`fpx` : `f` pour « finite field », `p` pour « prime » et `x` pour « extension », ce sous-dossier contient l'ensemble des fichiers relatifs aux extensions de corps finis et leur arithmétique. À ce jour, seulement les extensions de degré $d \in \{2, 3, 4, 6, 8, 9, 12, 18, 24, 48, 54\}$ y sont codées.

`ep` : `e` pour « elliptic » et `p` pour « prime », ce sous-dossier contient l'ensemble des fichiers sources qui concerne l'arithmétique des courbes elliptiques définies sur des corps premiers. On peut y trouver, entre autres, le fichier `relic_ep_param.c` dans lequel figurent les paramètres définissant les courbes elliptiques implémentées dans la librairie.

`epx` : `e` pour « elliptic », `p` pour « prime » et `x` pour « extension », ce sous-dossier contient l'ensemble des fichiers sources qui concerne l'arithmétique des courbes elliptiques définies sur des extensions de corps premiers. Lorsque le degré d'extension vaut d alors les fichiers concernés commencent par `relic_ep`d. À ce jour seulement deux extensions sont implémentées : les quadratiques ($d = 2$) et les quartiques ($d = 4$).

Remarque. Avant ce stage, seulement les fichiers des courbes elliptiques définies sur des extensions quadratiques (`relic_ep2`) étaient présents. En effet, comme la twist de notre courbe est définie sur une extension quartique, nous avions pour projet d'ajouter les fichiers `relic_ep4` nous permettant d'implémenter la twist de notre courbe. C'est après avoir échangé quelques mails avec le principal auteur de la librairie, Diego F. Aranha, que celui-ci a accepté de partager son avancement personnel sur les fichiers `relic_ep4`.

Répertoire `include`

Ce dossier contient l'ensemble des *headers* de chaque sous-dossier. C'est le glossaire de la librairie. Sur `relic`, les corps finis et les courbes elliptiques sont identifiés grâce à des identifiants. Par exemple pour notre courbe, `B24_319`, défini dans le fichier `relic_fp.h`, désigne le corps $\mathbb{F}_{q(x_0)}$. Le `B` vient de BLS, `24` du degré de plongement et `319` de la taille en bits de $q(x_0)$. Également, `B24_P319` défini dans le fichier `relic_ep.h` désigne notre courbe E.

Répertoire `test`

Ce dossier contient l'ensemble des tests nécessaires pour valider l'implémentation d'un objet dans `relic`. Si tous les tests passent avec succès, l'objet est alors correctement implémenté. Pour notre implémentation nous n'utilisons que les fichiers

- `test_fp.c` pour tester l'arithmétique de notre corps $\mathbb{F}_{q(x_0)}$;
- `test_fpx.c` pour tester l'arithmétique de notre corps $\mathbb{F}_{q(x_0)^4}$;
- `test_ep.c` pour tester l'arithmétique de $E/\mathbb{F}_{q(x_0)}$;
- `test_epx.c` pour tester l'arithmétique de la twist $E'/\mathbb{F}_{q(x_0)^4}$.

Répertoire `bench`

Lorsque l'on implémente un objet dans `relic`, il peut arriver que nous voulions savoir en combien de temps toutes les fonctions utilisant cet objet s'exécutent. Le dossier `bench` contient les fichiers permettant de « chronométrer » les fonctions relatives aux objets de la librairie. Dans notre cas nous utilisons le fichier `bench_ep.c` (respectivement `bench_epx.c`) pour connaître le temps d'exécution de l'exponentiation dans $\mathbb{G}_1$ (respectivement $\mathbb{G}_2$).

3.2.2 Implémentation de la courbe `B24_P319`

L'implémentation d'une courbe elliptique à couplage se fait en plusieurs étapes.

La première est d'implémenter le corps fini `B24_319`. Pour cela, après l'avoir définie dans les headers, nous ajoutons la graine

$$x_0 = -(2^{32} + 2^{19} + 2^{12} + 2^8 + 9)$$

lorsque la constante de préprocesseur `FP_PRIME` vaut 319 dans la fonction `fp_param_set()`. Ceci étant fait, il reste à ajouter le polynôme

$$q(X) = (X-1)^2(X^8 - X^4 + 1)/3 + X.$$

dans la fonction `fp_prime_set_pairf()` lorsque le paramètre `pairf` vaut `EP_B24`.

Remarque. Pour la suite, rappelons qu'une courbe $E/\mathbb{F}_q$ avec $q \equiv 1 \mod 3$ d'équation $y^2 = x^3 + b$ admet un endomorphisme GLV $\Phi : E \to E$ défini par $(x, y) \mapsto (\beta x, y) = \lambda(x, y)$ avec $\beta \in \mathbb{F}_q$ un élément d'ordre 3 et λ racine du polynôme $X^2 + X + 1 \mod \ell$.

La seconde étape est d'implémenter la courbe elliptique `B24_P319`. Pour cela, il faut définir certaines constantes. Commençons par le fichier `relic_ep_param.c` où se trouvent les paramètres permettant la définition de $\mathbb{G}_1$.

`B24_P319_A` : le paramètre a dans l'équation de la courbe ;

`B24_P319_B` : le paramètre b dans l'équation de la courbe ;

`B24_P319_X` : l'abscisse x d'un générateur du groupe $\mathbb{G}_1$;

`B24_P319_Y` : l'ordonnée y d'un générateur du groupe $\mathbb{G}_1$;

`B24_P319_R` : l'ordre ℓ du groupe $\mathbb{G}_1$;

`B24_P319_H` : le cofacteur h tel que $h\ell = \#E(\mathbb{F}_q)$;

`B24_P319_BETA` : un élément d'ordre 3 dans $\mathbb{F}_q$;

`B24_P319_LAMB` : le paramètre λ de l'endomorphisme GLV.

Il ne reste plus qu'à assigner le corps `B24_319` à la courbe `B24_P319` dans la fonction `ep_param_set()`. On procède de la même façon pour les paramètres permettant la définition de $\mathbb{G}_2$ dans le fichier `relic_ep4_curve.c` à la différence qu'un élément $a \in \mathbb{F}_{q^4}$ s'écrit avec quatre coordonnées $(a_0, a_1, a_2, a_3) \in \mathbb{F}_q^4$ satisfaisant

$$a = a_0 + a_1 u + (a_2 + a_3 u)v$$

où u et v sont les éléments permettant de construire notre tour d'extension (voir Figure 3.1). Ainsi pour implémenter une courbe elliptique le plus dur est de trouver les paramètres. Si certains sont immédiatement déterminés (comme a, b, ℓ), ce n'est pas toujours le cas des générateurs et de l'ordre de la courbe.

Chercher un générateur de $\mathbb{G}_1$

Rappelons que $\mathbb{G}_1 \simeq E(\mathbb{F}_q)[\ell]$. Pour trouver un générateur de $\mathbb{G}_1$, nous avons utilisé `SageMath` pour construire $E/\mathbb{F}_q$ à l'aide de la fonction `EllipticCurve(K,[a,b])` où `K` $= \mathbb{F}_q$. La méthode `.gens()[0]` du module des courbes elliptiques nous donne un générateur G de la courbe. Ainsi pour avoir un générateur de la ℓ-torsion sur $\mathbb{F}_q$ il suffit de multiplier G par le cofacteur $h = \#E/\ell$.

Chercher un générateur de $\mathbb{G}_2$

Rappelons que $\mathbb{G}_2 \simeq E'(\mathbb{F}_{q^4})[\ell]$. Comme `SageMath` ne construit pas le corps fini à q^4 éléments de la même manière que nous (Figure 3.1), on ne peut pas procéder comme pour $\mathbb{G}_1$ pour trouver un générateur de $\mathbb{G}_2$. Il faut donc construire $\mathbb{F}_{q^4}$ comme l'anneau quotient qui est reconnu comme tel par `SageMath`. L'idée pour trouver un générateur de $\mathbb{G}_2$ est de tirer aléatoirement un point $P \in E'(\mathbb{F}_{q^4})$ jusqu'à ce que $hP \neq \infty$ avec $h = \#E'/\ell$, ainsi le point $G = hP$ est un générateur de $\mathbb{G}_2$. Or pour cela il faut deux choses :
— une fonction qui calcule la racine carrée d'un élément de l'anneau quotient $\mathbb{F}_{q^4}$;
— calculer $\#E'(\mathbb{F}_{q^4})$.

Calcul de l'ordre de E' : Nous nous sommes inspirés de l'article [HSV06] pour calculer l'ordre de la twist de E. Sur $\mathbb{F}_q^k$, l'ordre de E est

$$p^k + 1 - \alpha^k - \beta^k$$

où α et β sont les racines complexes de $X^2 - tX + q$, le polynôme minimal du Frobenius. Or le discriminant de ce polynôme est donné par

$$\Delta = t^2 - 4q = -3f(x_0)^2$$

où le polynôme $f(X)$ est rappelé plus haut dans la définition des courbes BLS, nous en déduisons donc

$$\alpha = \frac{t + if\sqrt{3}}{2} \quad \text{et} \quad \beta = \frac{t - if\sqrt{3}}{2}.$$

On pose alors,

$$t_4 = \alpha^4 + \beta^4 \quad \text{et} \quad f_4 = \sqrt{\frac{t_4^2 - 4q^4}{-3}}.$$

La trace de la twist E' sur $\mathbb{F}_{q^4}$ est

$$t' = \frac{3f_4 + t_4}{2} \quad \text{ou} \quad t' = \frac{-3f_4 + t_4}{2}.$$

On déduit *in fine* l'ordre de la twist sur $\mathbb{F}_{q^4}$

$$\#E'(\mathbb{F}_{q^4}) = q^4 + 1 - t'.$$

Racine carré dans $\mathbb{F}_{q^4}$: Nous nous sommes inspirés de l'article [AR14] pour implémenter une fonction calculant la racine carrée dans $\mathbb{F}_{q^4}$ (voir Algo 4). L'idée de cet algorithme est d'utiliser le calcul de la racine carré dans $\mathbb{F}_{q^2}$ pour calculer celle dans $\mathbb{F}_{q^4}$. En effet soit a un résidu quadratique dans $\mathbb{F}_{q^4}$. Son symbole de Legendre est alors

$$\left(\frac{a}{q^4}\right) = a^{\frac{q^4-1}{2}} = 1$$

d'où

$$\begin{aligned} a &= a\left(a^{\frac{q^4-1}{2}}\right) \\ &= a\left(a^{\frac{q^2-1}{2}}\right)^{q^2+1} \\ &= \left(a^{\frac{q^2-1}{2}}\right)^{q^2} a^{\frac{q^2+1}{2}} \end{aligned}$$

De plus, puisque

$$\left(a^{\frac{q^2+1}{2}}\right)^{q^2-1} = 1$$

on en déduit $a^{(q^2+1)/2} \in \mathbb{F}_{q^2}$. On peut donc utiliser la racine carrée sur $\mathbb{F}_{q^2}$ pour calculer $\sqrt{a}$ sur $\mathbb{F}_{q^4}$

$$\sqrt{a} = \pm(a^{\frac{q^2-1}{4}})^{q^2}\sqrt{a^{\frac{q^2+1}{2}}}$$

Algorithme 4 Calcul de la racine carrée dans $\mathbb{F}_{q^4}$

Entrée : $a \in \mathbb{F}_{q^4}$
Sortie : x tel que $x^2 = a$ si a est un carré dans $\mathbb{F}_{q^4}$, $\perp$ sinon

```
c_0 ← 1
tant que c_0 = 1 faire
    c ←$ F_{q^4}
    c_0 = c^{(q^2-1)/2} mod q^2          ▷ Symbole de Legendre sur F_{q^4}
fin tant que
d ← c^{(q-1)/2}
e ← (dc)^{-1}
f ← (dc)^2
b ← a^{(q-1)/4}
a_0 ← (b^2)^q b^2
si a_0 = −1 alors
    retourner ⊥
fin si
si b^q b = 1 alors
    x_0 ← SQRT_{q^2}(b^2 a)
    x ← x_0 b^q
sinon
    x_0 ← SQRT_{q^2}(b^2 a f)
    x ← x_0 b^q e
fin si
retourner x
```

Pour tirer un point aléatoire de $E' : y^2 = x^3 + v^{-1}$ il faut tirer $x \in \mathbb{F}_{q^4}$ aléatoirement, calculer $y^2 = x^3 + v^{-1}$, vérifier si y^2 est un carré dans $\mathbb{F}_{q^4}$, si ce n'est pas le cas on recommence avec un nouveau $x \in \mathbb{F}_{q^4}$ jusqu'à ce que y^2 soit un carré. On calcule alors y une racine carrée de y^2 avec l'Algorithme 4. Ainsi, on a le point $P = (x, y)$ sur la courbe E'. On recommence cette opération jusqu'à ce que $hP \neq \infty$ pour obtenir un générateur de $\mathbb{G}_2$.

Compilation

La librairie `relic` utilise le générateur de système de construction *cmake* pour créer des *Makefiles* à partir du fichier *CMakeLists.txt*. C'est dans le fichier *CMakeCache.txt* que l'on choisit la valeur des constantes de préprocesseur et notamment la valeur de `FP_PRIME`.

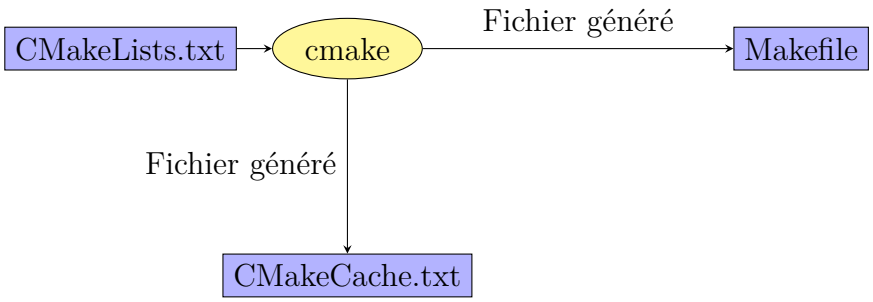

FIGURE 3.3 – Construction de système de compilation

Voici la commande à effectuer pour paramétrer le fichier CMakeCache.txt par rapport à notre courbe :

```
cmake -DWSIZE=64 -DRAND=UDEV -DSHLIB=OFF -DSTBIN=ON -DTIMER=HREAL -
    DCHECK=off -DVERBS=off -DARITH=easy -DFP_PRIME=319 -DFP_METHD="INTEG
    ;INTEG;INTEG;MONTY;LOWER;SLIDE" -DCFLAGS="-O3 -funroll-loops -fomit-
    frame-pointer -march=native -mtune=native" -DFP_PMERS=off -DFP_QNRES
    =on -DFPX_METHD="INTEG;INTEG;LAZYR" -DEP_PLAIN=off -DEP_SUPER=off -
    DPP_METHD="LAZYR;OATEP" -DWITH="ALL" $1
```

Listing 3.1 – Preset pour la courbe BLS_P319

3.3 Comparaison

Après avoir lancé tous les tests de notre courbe `B24_P319`, nous désirons la comparer avec d'autres courbes visant les 128 bits de sécurité, déjà implémentées dans `relic`. Les courbes choisies sont :

`BN_P446` : courbe de la famille Barreto-Naehrig [BN05] de degré de plongement 12 et définie sur un corps à 446 bits. Cette famille de courbes est mise à jour par Barbulescu et Duquesne dans [BD19].

`B12_P446` : courbe de la même famille que la nôtre mais avec un degré de plongement 12 et définie sur un corps fini à 446 bits.

`CP8_P544` : courbe de la famille Cocks-Pinch et définie sur un corps à 544 bits. Cette courbe est proposée par de récents travaux [GMT20] et vise une sécurité de 128 bits.

Toutes ces courbes bénéficient d'un endomorphisme GLV sur $\mathbb{G}_1$. Le tableau 3.4 résume les résultats obtenus sur un ordinateur équipé d'un processeur Intel Core i5-10310U à 1.70GHz. Le temps est donné en microsecondes et le nombre d'itérations effectuées est de 10^6. Rappelons que les éléments de $\mathbb{G}_2$ sont de taille au plus $e \log_2(q)$ ce qui explique le fait qu'avec notre courbe l'exponentiation sur $\mathbb{G}_2$ est plus lente qu'avec les courbes `B12_P446` et `BN_P446`, en effet les éléments de $\mathbb{G}_2$ pour ces courbes sont presque deux fois plus petits. De plus, ces résultats mettent en évidence l'inconvénient des courbes d'ordre premier Barreto-Naehrig : puisque $\#E(\mathbb{F}_q) = \ell$, les tests d'appartenance sur ces courbes sont inutiles, mais lorsqu'il s'agit de l'exponentiation sur $\mathbb{G}_1$ car on fait croître ℓ inutilement. Malheureusement, nous n'avons pas eu le temps de changer le modèle de la courbe en la mettant sous la forme d'Edwards, ce qui pourrait nettement ameliorer l'exponentiation dans les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$, voir [BCN13] pour plus de détails.

Courbe	B24_P319	B12_P446	BN_P446	CP8_P544
$\log_2(p)$	319	446(+40%)	446(+40%)	544(+70%)
$\log_2(\ell)$	257	299	446	256
mots machine	5	7	7	9
k	24	12	12	8
e	4	2	2	2
Exp. sur $\mathbb{G}_1$ (μs)	415	886(+113%)	1304(+214%)	1545(+272%)
Exp. sur $\mathbb{G}_2$ (μs)	4487	2329(−48%)	3533(−21%)	4924(+10%)

TABLE 3.4 – Comparaison entre la courbe B24_P319 et d'autres courbes standardisées visant un niveau de sécurité de 128 bits.

Conclusion

Ce stage était composé de deux parties : la recherche de paramètres qui m'a grandement aidé à m'approprier le sujet ainsi que les connaissances théoriques. Ensuite, la deuxième partie du stage était l'implémentation où j'ai pu découvrir le fonctionnement d'une librairie comme RELIC.

D'un point de vue mathématiques, ce stage m'a permis l'ancrage de mes acquis, notamment en théorie des nombres. J'ai aussi développé mes connaissances des courbes elliptiques et compris la plupart des mécanismes utilisés par celles-ci.

D'un point de vue informatique, ce stage m'aura permis de me perfectionner en développement, et plus particulièrement en découvrant de nouveaux outils mais aussi de nouvelles façons de penser son code.

Au niveau "vie professionnel", malgré la situation sanitaire que traverse le pays et plus généralement le monde, j'ai découvert au cours de ce stage de nouvelles façons de travailler qui me sont plus adaptées. J'ai aussi développé une certaine autonomie durant ce stage. Pour tout cela, je remercie mon encadrant Mr Clarisse Rémi, d'abord pour m'avoir choisi pour faire ce stage et ensuite pour sa disponibilité et son aide tout au long du stage, sa patience lorsqu'il fallait m'expliquer plusieurs fois le même concept ainsi que sa gentillesse d'avoir satisfait mes pics de curiosité.

Bibliographie

[AM93] A Oliver L Atkin et François Morain. "Elliptic curves and primality proving". In : *Mathematics of computation* 61.203 (1993), p. 29-68.

[AR14] Gora Adj et Francisco Rodriguez-Henriquez. "Square Root Computation over Even Extension Fields". In : *IEEE Trans. Computers* 63.11 (2014), p. 2829-2841. doi : 10.1109/TC.2013.145. url : https://doi.org/10.1109/TC.2013.145.

[Ara+] D. F. Aranha et al. *RELIC is an Efficient LIbrary for Cryptography*. https://github.com/relic-toolkit/relic.

[Bar20] Elaine Barker. *Recommendation for Key Management: Part 1 - General*. en. 2020-05-04 2020. doi : https://doi.org/10.6028/NIST.SP.800-57pt1r5.

[BCN13] Joppe W. Bos, Craig Costello et Michael Naehrig. "Exponentiating in Pairing Groups". In : *IACR Cryptol. ePrint Arch.* 2013 (2013), p. 458. url : http://eprint.iacr.org/2013/458.

[BD19] Razvan Barbulescu et Sylvain Duquesne. "Updating Key Size Estimations for Pairings". In : *J. Cryptol.* 32.4 (2019), p. 1298-1336. doi : 10.1007/s00145-018-9280-5. url : https://doi.org/10.1007/s00145-018-9280-5.

[Ber+08] Daniel J. Bernstein et al. "Twisted Edwards Curves". In : *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings*. Sous la dir. de Serge Vaudenay. T. 5023. Lecture Notes in Computer Science. Springer, 2008, p. 389-405. doi : 10.1007/978-3-540-68164-9_26. url : https://doi.org/10.1007/978-3-540-68164-9%5C_26.

[BF01] Dan Boneh et Matthew K. Franklin. "Identity-Based Encryption from the Weil Pairing". In : *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*. Sous la dir. de Joe Kilian. T. 2139. Lecture Notes in Computer Science. Springer, 2001, p. 213-229. doi : 10.1007/3-540-44647-8_13. url : https://doi.org/10.1007/3-540-44647-8%5C_13.

[BK98] R. Balasubramanian et Neal Koblitz. "The Improbability That an Elliptic Curve Has Subexponential Discrete Log Problem under the Menezes - Okamoto - Vanstone Algorithm". In : *J. Cryptol.* 11.2 (1998), p. 141-145. doi : 10.1007/s001459900040. url : https://doi.org/10.1007/s001459900040.

[BLS01] Dan Boneh, Ben Lynn et Hovav Shacham. “Short Signatures from the Weil Pairing”. In : *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*. Sous la dir. de Colin Boyd. T. 2248. Lecture Notes in Computer Science. Springer, 2001, p. 514-532. DOI : 10.1007/3-540-45682-1_30. URL : https://doi.org/10.1007/3-540-45682-1%5C_30.

[BLS11] Daniel J. Bernstein, Tanja Lange et Peter Schwabe. “On the Correct Use of the Negation Map in the Pollard rho Method”. In : *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*. Sous la dir. de Dario Catalano et al. T. 6571. Lecture Notes in Computer Science. Springer, 2011, p. 128-146. DOI : 10.1007/978-3-642-19379-8_8. URL : https://doi.org/10.1007/978-3-642-19379-8%5C_8.

[BN05] Paulo S. L. M. Barreto et Michael Naehrig. “Pairing-Friendly Elliptic Curves of Prime Order”. In : *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*. Sous la dir. de Bart Preneel et Stafford E. Tavares. T. 3897. Lecture Notes in Computer Science. Springer, 2005, p. 319-331. DOI : 10.1007/11693383_22. URL : https://doi.org/10.1007/11693383%5C_22.

[BW05] Friederike Brezing et Annegret Weng. “Elliptic Curves Suitable for Pairing Based Cryptography”. In : *Des. Codes Cryptogr.* 37.1 (2005), p. 133-141. DOI : 10.1007/s10623-004-3808-4. URL : https://doi.org/10.1007/s10623-004-3808-4.

[CDS20] Rémi Clarisse, Sylvain Duquesne et Olivier Sanders. “Curves with Fast Computations in the First Pairing Group”. In : *Cryptology and Network Security - 19th International Conference, CANS 2020, Vienna, Austria, December 14-16, 2020, Proceedings*. Sous la dir. de Stephan Krenn, Haya Shulman et Serge Vaudenay. T. 12579. Lecture Notes in Computer Science. Springer, 2020, p. 280-298. DOI : 10.1007/978-3-030-65411-5_14. URL : https://doi.org/10.1007/978-3-030-65411-5%5C_14.

[CLN11] Craig Costello, Kristin E. Lauter et Michael Naehrig. “Attractive Subfamilies of BLS Curves for Implementing High-Security Pairings”. In : *Progress in Cryptology - INDOCRYPT 2011 - 12th International Conference on Cryptology in India, Chennai, India, December 11-14, 2011. Proceedings*. Sous la dir. de Daniel J. Bernstein et Sanjit Chatterjee. T. 7107. Lecture Notes in Computer Science. Springer, 2011, p. 320-342. DOI : 10.1007/978-3-642-25578-6_23. URL : https://doi.org/10.1007/978-3-642-25578-6%5C_23.

[Cos12] Craig Costello. “Fast formulas for computing cryptographic pairings”. Thèse de doct. Queensland University of Technology, 2012.

[CP01] C. Cocks et R.G.E Pinch. “A Taxonomy of Pairing-Friendly Elliptic Curves”. In : *Unpublished manuscript* (2001).

[DEM05] Régis DUPONT, Andreas ENGE et François MORAIN. "Building Curves with Arbitrary Small MOV Degree over Finite Prime Fields". In : *J. Cryptol.* 18.2 (2005), p. 79-89. DOI : 10.1007/s00145-004-0219-7. URL : https://doi.org/10.1007/s00145-004-0219-7.

[Dev+06] Augusto Jun DEVEGILI et al. *Multiplication and Squaring on Pairing-Friendly Fields*. augusto@ic.unicamp.br 13564 received 13 Dec 2006, last revised 20 Feb 2007. 2006. URL : http://eprint.iacr.org/2006/471.

[FST10] David FREEMAN, Michael SCOTT et Edlyn TESKE. "A Taxonomy of Pairing-Friendly Elliptic Curves". In : *J. Cryptol.* 23.2 (2010), p. 224-280. DOI : 10.1007/s00145-009-9048-z. URL : https://doi.org/10.1007/s00145-009-9048-z.

[GMT20] Aurore GUILLEVIC, Simon MASSON et Emmanuel THOMÉ. "Cocks-Pinch curves of embedding degrees five to eight and optimal ate pairing computation". In : *Des. Codes Cryptogr.* 88.6 (2020), p. 1047-1081. DOI : 10.1007/s10623-020-00727-w. URL : https://doi.org/10.1007/s10623-020-00727-w.

[GPS08] Steven D. GALBRAITH, Kenneth G. PATERSON et Nigel P. SMART. "Pairings for cryptographers". In : *Discret. Appl. Math.* 156.16 (2008), p. 3113-3121. DOI : 10.1016/j.dam.2007.12.010. URL : https://doi.org/10.1016/j.dam.2007.12.010.

[GS19] Aurore GUILLEVIC et Shashank SINGH. "On the alpha value of polynomials in the tower number field sieve algorithm". In : *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 885. URL : https://eprint.iacr.org/2019/885.

[Gui20] Aurore GUILLEVIC. "A Short-List of Pairing-Friendly Curves Resistant to Special TNFS at the 128-Bit Security Level". In : *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*. Sous la dir. d'Aggelos KIAYIAS et al. T. 12111. Lecture Notes in Computer Science. Springer, 2020, p. 535-564. DOI : 10.1007/978-3-030-45388-6_19. URL : https://doi.org/10.1007/978-3-030-45388-6%5C_19.

[HMV04] Darrel HANKERSON, Alfred MENEZES et Scott VANSTONE. *Guide to elliptic curve cryptography*. New York : Springer, 2004. ISBN : 0–387–95273–X.

[HSV06] Florian HESS, Nigel P. SMART et Frederik VERCAUTEREN. "The Eta Pairing Revisited". In : *IEEE Trans. Inf. Theory* 52.10 (2006), p. 4595-4602. DOI : 10.1109/TIT.2006.881709. URL : https://doi.org/10.1109/TIT.2006.881709.

[JL11] Antoine JOUX et Reynald LERCIER. "Number Field Sieve for the DLP". In : *Encyclopedia of Cryptography and Security, 2nd Ed.* Sous la dir. d'Henk C. A. van TILBORG et Sushil JAJODIA. Springer, 2011, p. 867-873. DOI : 10.1007/978-1-4419-5906-5_834. URL : https://doi.org/10.1007/978-1-4419-5906-5%5C_834.

[Jou00] Antoine Joux. “A One Round Protocol for Tripartite Diffie-Hellman”. In : *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2-7, 2000, Proceedings*. Sous la dir. de Wieb Bosma. T. 1838. Lecture Notes in Computer Science. Springer, 2000, p. 385-394. doi : 10.1007/10722028_23. url : https://doi.org/10.1007/10722028%5C_23.

[KB16] Taechan Kim et Razvan Barbulescu. “Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case”. In : *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*. Sous la dir. de Matthew Robshaw et Jonathan Katz. T. 9814. Lecture Notes in Computer Science. Springer, 2016, p. 543-571. doi : 10.1007/978-3-662-53018-4_20. url : https://doi.org/10.1007/978-3-662-53018-4%5C_20.

[Men97] Alfred J. Menezes. *Elliptic curve public key cryptosystems*. T. 234. The Kluwer international series in engineering and computer science. Kluwer, 1997. isbn : 978-0-7923-9368-9.

[MF05] Angela Murphy et Noel Fitzpatrick. *Elliptic Curves for Pairing Applications*. Cryptology ePrint Archive, Report 2005/302. https://ia.cr/2005/302. 2005.

[MNT00] Atsuko Miyaji, Masaki Nakabayashi et Shunzo Takano. “Characterization of Elliptic Curve Traces under FR-Reduction”. In : *Information Security and Cryptology - ICISC 2000, Third International Conference, Seoul, Korea, December 8-9, 2000, Proceedings*. Sous la dir. de Dongho Won. T. 2015. Lecture Notes in Computer Science. Springer, 2000, p. 90-108. doi : 10.1007/3-540-45247-8_8. url : https://doi.org/10.1007/3-540-45247-8%5C_8.

[MVO91] Alfred Menezes, Scott A. Vanstone et Tatsuaki Okamoto. “Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field”. In : *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*. Sous la dir. de Cris Koutsougeras et Jeffrey Scott Vitter. ACM, 1991, p. 80-89. doi : 10.1145/103418.103434. url : https://doi.org/10.1145/103418.103434.

[Sho97] Victor Shoup. “Lower Bounds for Discrete Logarithms and Related Problems”. In : *Advances in Cryptology - EUROCRYPT ’97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*. Sous la dir. de Walter Fumy. T. 1233. Lecture Notes in Computer Science. Springer, 1997, p. 256-266. doi : 10.1007/3-540-69053-0_18. url : https://doi.org/10.1007/3-540-69053-0%5C_18.

[Sil09] Joseph H Silverman. *The Arithmetic of Elliptic Curves*. Graduate texts in mathematics. Dordrecht : Springer, 2009. doi : 10.1007/978-0-387-09494-6. url : https://cds.cern.ch/record/1338326.

Table des figures

Liste des tableaux