Recherche de primitives pour la cryptographie à base de couplage

Aminatou Pecha Njiahouo

UNIVERSITÉ PARIS 8

UNIVERSITÉ DE YAOUNDÉ 1

THÈSE

pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ PARIS 8 ET DE L'UNIVERSITÉ DE YAOUNDÉ 1

Discipline : **Mathématiques**

Écoles Doctorales : **Cognition, Langage, Interaction. Mathématiques, Informatique, Bioinformatique et Applications.**

présentée et soutenue publiquement par

PECHA NJIAHOUO Aminatou

08 Décembre 2017

Recherche de primitives pour la cryptographie à base de couplage

JURY

M. SCHMID Wolfgang	Directeur de thèse
M. NKUIMI-JUGNIA Célestin	Directeur de thèse (UY1)
Mme EL MRABET Nadia	Co-Directrice de thèse
M. GOUBIN Louis	Rapporteur
M. TIEUDJO Daniel	Rapporteur
M. YENGUI Ihsen	Examinateur

Table des matières

Remerciements

Je tiens tout d'abord à adresser mes remerciements les plus sincères à mes directeurs de recherche, Pr Wolfgang Schmid, Pr Célestin Nkuimi et Dr Nadia EL Mrabet, pour le soutien constant, la générosité et la grande disponibilité dont ils ont fait preuve à mon égard. Leurs précieux conseils et encouragements ont non seulement facilité l'élaboration de ce travail mais m'ont aussi permis d'enrichir l'horizon de mes réflexions.

J'exprime ma gratitude aux membres du jury qui ont accepté de participer à la soutenance de cette thèse.

Mes remerciements vont ensuite à mon tendre époux Dr Amadou Nsangou pour le soutien sans faille qu'il m'a exprimé à travers ses multiples encouragements et surtout pour les énormes sacrifices consentis pour que je puisse achever ce cycle de doctorat.

Je tiens également à remercier Dr Tony Ezome, le coordonnateur du projet PRMAIS, pour m'avoir intégré à ce projet qui a facilité mes séjours en France ; Dr Emmanuel Fouotsa, l'un de mes co-auteurs pour sa disponibilité permanente et son efficacité et Dr Emmanuel Pola à la fois pour ses relectures de ma thèse, son sens d'écoute très poussé et le cadre de travail serein qu'il a bien voulu mettre à ma disposition.

Un grand merci à ma famille et à mes amis pour leur assistance, leur patience et leurs encouragements infaillibles.

Introduction

Une primitive cryptographique est une brique élémentaire qui est utilisée dans la construction d'un protocole complet. Ce dernier est la description de l'ensemble des données nécessaires pour la mise en place d'un mécanisme de cryptographie : ensemble des messages clairs, des messages chiffrés, des clés possibles et des transformations. Le travail élaboré dans cette thèse consiste à trouver des briques élémentaires qui entrent dans la construction des cryptosystèmes basés sur le couplage. Le couplage comme notion mathématique introduite par André Weil en 1948 [Wei48], connaît un regain d'intérêt dans la communauté des cryptographes depuis le début des années 90. Cet outil mathématique vu sous le prisme de la cryptographie est, dans la pratique, défini sur les courbes elliptiques. En fait, les couplages sont des applications bilinéaires non-dégénérées définies sur le groupe de points rationnels d'une courbe elliptique ou hyper-elliptique [Was08]. Ainsi, en nous situant dans le contexte de la cryptographie à base de couplage et en nous appuyant sur la définition de primitive, notre principale préoccupation est de trouver des couplages ou des courbes elliptiques ordinaires adaptées au couplage. Il s'agit plus précisément de rechercher des courbes elliptiques ainsi que des couplages dits optimaux définis sur celles-ci.

Les couplages ont, dans un premier temps, été utilisés d'un point de vue destructif car ils ont permis de réduire des problèmes difficiles en des problèmes faciles. En effet, en 1993, Menezes, Okamoto et Vanstone [MOV93] ont montré que le couplage de Weil permet de réduire le problème du logarithme discret de la courbe elliptique en un problème du logarithme discret dans le groupe multiplicatif d'un corps fini. Ils ont aussi, dans un second temps, été utilisés d'un point de vue constructif car ils ont permis de construire des protocoles originaux tels que les protocoles cryptographiques basés sur l'identité [BF01, Coc01, LQ04] et de simplifier certains protocoles cryptographiques : l'échange tripartite de clé en seul tour d'Antoine Joux [Jou00] et les schémas de signature courte [BLS04b] en sont des exemples. Une revue de ces applications peut être trouvée dans [DBS04], [BSS05, Chap. X]. Ces applications justifient les travaux se focalisant sur la construction des couplages pour lesquels leurs calculs sont plus efficaces que ceux existants.

Généralement, si E est une courbe elliptique ordinaire définie sur un

corps fini $\mathbb{F}_q$ et r un grand diviseur premier de l'ordre du groupe $E(\mathbb{F}_q)$, le degré de plongement de E suivant r et q, est le plus petit entier k tel que r divise $q^k - 1$. Un couplage envoie un couple de points de la courbe elliptique E dans le sous groupe multiplicatif d'ordre r de $\mathbb{F}_{q^k}^*$. Calculer de manière effective un couplage revient à exécuter l'algorithme de Miller [Mil04]. Pour que ce calcul soit efficace, les ingrédients requis à cet effet sont l'utilisation des courbes elliptiques définies sur $\mathbb{F}_q$ adaptées au couplage [FST10], la possession d'une arithmétique efficace dans la tour d'extension de corps associés à $\mathbb{F}_{q^k}$ [KM05, GS10, Kar13, DSD07] et la réduction du nombre d'itérations durant l'algorithme de Miller. Sur ce dernier point, beaucoup de travaux élaborés ont conduit au concept de réseaux de couplage [Hes08] ou de couplage optimal décrit par Vercauteren qui peut être calculé avec le plus petit nombre d'itérations dans l'algorithme de Miller [Ver10]. Plus précisément, un couplage $b : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ avec $\mathbb{G}_1$, $\mathbb{G}_2$ des sous groupes de $E(\mathbb{F}_{q^k})$ et $\mathbb{G}_3$ un sous groupe de $\mathbb{F}_{q^k}^*$ tous d'ordre r, est dit **optimal** s'il peut être calculé approximativement en $\log_2 r/\varphi(k)$ itérations de l'algorithme de Miller, où φ représente la fonction indicatrice d'Euler. Ainsi, le calcul du couplage est plus efficace en terme d'itérations de l'algorithme de Miller pour des courbes telles que $\varphi(k)$ est grande. Il se dégage que l'efficacité de ce calcul fait entrer en balance un degré de plongement k pas trop grand pour que l'arithmétique dans $\mathbb{F}_{q^k}$ soit rapide, et k ayant $\varphi(k)$ grande. En plus de ces contraintes liées à l'efficacité, il y a une contrainte sur le rapport : $\rho = \frac{\log_2 q}{\log_2 r}$. Il donne le surcoût induit par l'arithmétique dans $\mathbb{F}_q$. Lorsque $\rho \approx 1$, les calculs sur la courbe elliptique et sur le corps fini sont beaucoup plus efficaces que pour des courbes ayant $\rho \approx 2$.

Concernant les courbes elliptiques adaptées à la cryptographie à base de couplage, elles sont assez rares. Ceci dû au fait que ces courbes elliptiques doivent posséder un degré de plongement assez petit de telle sorte que le calcul du couplage soit facile mais suffisamment grand pour que la résolution du problème du logarithme discret soit infaisable en un temps raisonnable dans le groupe multiplicatif $\mathbb{F}_{q^k}^*$. Car la sécurité d'un cryptosystème basé sur le couplage repose sur la difficulté à resoudre, en un temps raisonnable, le problème du logarithme discret à la fois dans le groupe multiplicatif de $\mathbb{F}_{q^k}$ et dans le groupe $E(\mathbb{F}_q)$. Freeman et *al.* [FST10] proposent une taxonomie de telles courbes qu'ils ont baptisé les courbes elliptiques bien couplées. Ces dernières sont toutes définies sur des corps premiers. Cependant, les corps premiers ne possèdent pas toujours une arithmétique qui se prête à une implémentation efficace. Ce qui nous amène à nous interroger sur la possibilité de construire des courbes elliptiques ordinaires adaptées au couplage définies sur les extensions de corps optimales. Car il est connu que ces corps tirent avantage de l'arithmétique rapide se trouvant dans le processeur pour une implémentation efficace. C'est en 1998 que Daniel Bailey et Christof Paar [BP98] ont introduit cette famille de corps finis appelée en anglais «Optimal

Extension Field, OEF». Ces extensions de corps optimales sont définies sur un corps de base de la taille d'un mot machine. Il s'agit là d'examiner la possibilité de construire des courbes elliptiques définies sur les corps OEF convenables à la cryptographie à base de couplage.

Partant du constat de l'insuffisance des travaux portant sur le calcul du couplage optimal Ate sur les courbes elliptiques de degré de plongement impair, il est apparu nécessaire d'explorer davantage ce champ de calcul. Il faut rappeler que le couplage de Tate réduit et ses versions améliorées (Ate réduit, twisted Ate réduit [HSV06], optimal Ate [Ver10]) sont les plus utilisés en cryptographie. Ces couplages prennent en argument deux points linéairement indépendants d'ordres r de sous groupes de $E(\mathbb{F}_{q^k})$ et renvoient un élément du sous-groupe des racines r-ièmes de l'unité dans le corps fini $\mathbb{F}_{q^k}$. Leurs calculs reviennent à exécuter l'algorithme de Miller et une exponentiation finale. Une attention particulière est portée sur l'étape de l'exponentiation finale car calculer efficacement cette dernière est devenu une tâche laborieuse.

La majorité des travaux proposés dans la littérature est concentrée sur l'amélioration de l'efficacité du calcul des couplages. C'est dans ce contexte d'étude des performances du calcul des couplages qu'en 2005 Koblitz et Menezes [KM05] ont examiné l'efficacité du couplage de Weil par rapport à celui du couplage de Tate. Cette étude a révélé qu'à des hauts niveaux de sécurité, le calcul du couplage de Weil est parfois plus rapide que celui du couplage de Tate. Peu après, en 2006, Granger et al [GPS06] ont réexaminé les techniques d'implémentation des couplages sur les courbes elliptiques ordinaires à divers niveaux de sécurité. Contrairement au précédent résultat, ils sont arrivés à la conclusion que le couplage de Tate est plus efficace que le couplage de Weil à tous niveaux de sécurité. Dès lors, s'ensuit une série de travaux [BGDM+10a, GF16, LMN10, Ver10, ZWL13] prouvant que le couplage de Tate et ses variantes (optimal Ate, twisted Ate...) offrent plus d'efficacité que le couplage de Weil. C'est en 2012 qu'un sursaut d'espoir est né par l'introduction d'un couplage optimal de type Weil convenable pour des exécutions en parallèle et appelé couplage β-Weil [AFCK+12]. Aranha et *al.* ont prouvé que ce couplage β-Weil peut être plus efficace que les couplages de type Tate.

Ce travail s'articule autour de 4 chapitres. Dans le premier chapitre, nous avons dans un premier temps introduit les notions de base nécessaires à la compréhension de la suite du travail. A cet effet, nous donnons les définitions formelles des corps finis, des courbes elliptiques définies sur des corps finis, des couplages ainsi que des notions liées au calcul des couplages. Dans un second temps, nous apportons des précisions sur toutes les contraintes mathématiques à prendre en compte pour assurer la fiabilité et la sécurité d'un cryptosystème à base de couplage à la lumière des récentes et considérables avancées observées dans la résolution du problème du logarithme discret sur

les corps finis [JL03, JP13, BGK15, KB16, MSS16].

Les résultats obtenus dans les chapitres 2, 3 et 4 constituent notre contribution à cette thèse. Au chapitre 2, nous présentons les problèmes rencontrés lors de la génération des courbes elliptiques définies sur les corps OEF convenables à la cryptographie à base de couplage. Nous ressortons également toutes les contraintes mathématiques liées à la génération de telles courbes.

Dans le chapitre 3, une bonne sélection des paramètres aux niveaux de sécurité respectifs 128, 192 et 256-bits, d'après la table 3.1 se trouvant au chapitre 3 p.36, nous a permis d'améliorer les coûts théoriques pour les étapes de l'algorithme de Miller et de l'exponentiation finale en utilisant la méthode basée sur les réseaux sur les courbes elliptiques ordinaires bien couplées de degré de plongement $k = 9, 15$ et 27 aux niveaux de sécurité de $128, 192$ et 256-bits respectivement comparativement aux résultats des travaux existants sur de telles courbes, admettant des tordues d'ordre 3, afin d'améliorer le coût théorique du calcul du couplage optimal Ate sur ces courbes sus-citées. Il est important de mentionner que les résultats présentés dans ce chapitre ont été obtenus au cours de la période pendant laquelle quelques avancées dans le calcul du logarithme discret sur les corps finis avaient été observées [JP13, BGK15]. Mais les algorithmes proposés à cet effet étaient considérés non-pratiques et les recommandations de tailles de paramètres n'avaient pas été changées. C'est pour cette raison que nous nous sommes appuyés sur les recommandations de tailles de paramètres de la table 3.1 proposées par Freeman et *al.* dans [FST10] pour mener notre étude. Nous donnons l'arithmétique et l'estimation des coûts des opérations dans la tour d'extensions de corps $\mathbb{F}_{p^9}$, $\mathbb{F}_{p^{15}}$ et $\mathbb{F}_{p^{27}}$. Cette étude a conduit à l'élaboration d'un article soumis en expertise à un journal avec pour co-auteurs Emmanuel Fouotsa de l'Université de Bamenda et de Nadia El Mrabet de l'équipe Systèmes Architectures Sécurisées du Centre Micro-électronique de Provence George Charpak (SAS-CMP) de Gardanne.

Dans le chapitre 4, nous avons proposé une extension du couplage β-Weil initialement introduit par Aranha et *al.* sur les courbes elliptiques ordinaires de degré de plongement pair à toutes courbes elliptiques ordinaires de degré de plongement quelconque. De ce résultat, nous avons identifié une famille de courbes elliptiques ordinaires bien couplées et avons proposé sur une telle famille un nouveau couplage optimal. Ce dernier est défini comme produit de fonctions rationnelles avec la même boucle de Miller. Ce qui nous a permis d'utiliser la technique du multi-pairing pour son calcul. Ce nouveau couplage nécessite une simple exponentiation finale en ce sens que son coût s'évalue en terme de coût des endomorphismes de Frobenius. Une comparaison est faite sur l'efficacité de ce nouveau couplage optimal avec le couplage optimal Ate. Nous proposons des exemples d'applications de ce couplage sur les courbes elliptiques Barreto-Lynn-Scott de degré de plongement 12

(BLS12) et sur les courbes elliptiques ordinaires bien couplées de degré de plongement 15. Nous avons rédigé et soumis en expertise à un journal un second article présentant la quintessence du développement de ces derniers résultats avec la collaboration de Nadia EL Mrabet et Emmanuel Fouotsa. Comme publication, nous avons un article intitulé «Optimal Weil pairing on Elliptic Curves with embedding degre 15» qui a été accepté et apparaîtra dans la revue internationale à comité de lecture : African Journal of Pure and Applied Mathematics IMHOTEP- Math. Proc. 4(2017).

Chapitre 1

État de l'art

Sommaire

Le but de ce chapitre extrait de la littérature ([Sil09],[Was08],...) est d'introduire les notions de base nécessaires à la compréhension de toute la suite du manuscrit.

1.1 Courbes elliptiques sur les corps finis

1.1.1 Corps finis

Les corps finis jouent un rôle essentiel et fondamental en cryptographie. Un corps est une structure algébrique munie de deux lois de composition

internes (ou opérations) dans laquelle on peut additionner, multiplier et inverser tout élément non nul. De manière formelle, on a la définition suivante :

Définition 1. *Soit $\mathbb{K}$ un ensemble non vide muni de deux opérations, addition et multiplication, notées respectivement $+$ et $\times$.*

On dit que $\mathbb{K}$ est un corps si

1. *$(\mathbb{K}, +)$ est un groupe abelien ; l'élément neutre pour $+$ est alors noté 0.*
2. *$(\mathbb{K} \setminus \{0\}, \times)$ est un groupe abelien ; l'élément neutre pour $\times$ est alors noté 1.*
3. *La multipication est distributive par rapport à l'addition ; i.e.*
   $(x + y) \times z = (x \times z) + (y \times z)$ pour tous $x, y, z \in \mathbb{K}$.

Si l'ensemble $\mathbb{K}$ est fini, alors on dit que $\mathbb{K}$ est un corps fini.

Théorème 1. *Pour chaque puissance p^n d'un nombre premier, il existe un corps fini, unique à isomorphisme près, de cardinal p^n. On le note $\mathbb{F}_{p^n}$.*

Remarque 1. *Les corps $\mathbb{F}_p$ sont appelés les corps premiers et les corps $\mathbb{F}_{p^n}$ sont appelés les extensions de corps finis qui sont construits à partir de corps premiers. Notons que le corps fini $\mathbb{F}_{p^n}$ est de caractéristique p. Il existe toujours au moins un polynôme f de degré n, irreductible sur $\mathbb{F}_p$, tel que $\mathbb{F}_{p^n} = \mathbb{F}_p[X]/(f)$. C'est aussi un espace vectoriel sur le corps $\mathbb{F}_p$, et la famille $\{1, X, ..., X^{n-1}\}$ en est une base.*

Théorème 2. *Pour chaque corps fini $\mathbb{F}_{p^n}$, le groupe multiplicatif $\mathbb{F}_{p^n}^*$ est cyclique.*

Une étude complète des corps finis se trouve dans le livre de Lidl et Niederreiter [LN97].

Tour d'extension de corps finis

Définition 2. *Soit $\mathbb{F}_{p^n}$ un corps fini. Si le degré de l'extension n est un entier composé et $n = \nu\zeta$ est une factorisation non triviale de n. Alors $\mathbb{F}_{p^n}$ admet une représentation en tour d'extension de corps $\mathbb{F}_{q^\zeta}$, où $q = p^\nu$.*

Ainsi l'arithmétique dans $\mathbb{F}_{p^n}$ peut se faire étape par étape dans chacune des petites extensions.

1.1.2 Définition d'une courbe elliptique sur un corps fini

Les notions abordées dans cette section émanent de [Sil09] et [Was08]. Dans un cadre algébrique plus général, en considérant un corps $\mathbb{K}$ de caractéristique quelconque et $\overline{\mathbb{K}}$ sa clôture algébrique i.e. une extension algébrique

de $\mathbb{K}$ qui est algébriquement close ; il est nécessaire, dans un premier temps, de rappeler la définition du plan projectif sur $\overline{\mathbb{K}}$ afin de donner une définition algébrique générale d'une courbe elliptique définie sur $\mathbb{K}$.

Définition 3. *Le plan projectif sur $\overline{\mathbb{K}}$, noté $\mathbb{P}^2(\overline{\mathbb{K}})$ ou $\mathbb{P}^2$, est l'ensemble quotient*

$$\overline{\mathbb{K}}^3 - \{(0,0,0)\}/\cong,$$

où $\cong$ est la relation d'équivalence telle que pour tous élélements (X,Y,Z) et (X',Y',Z') non nuls de $\overline{\mathbb{K}}^3$,

$$(X,Y,Z) \cong (X',Y',Z') \iff \text{il existe } \alpha \in \overline{\mathbb{K}}^* \text{ tel que } (X',Y',Z') = \alpha(X,Y,Z).$$

Le plan projectif $\mathbb{P}^2(\overline{\mathbb{K}})$ s'identifie à l'ensemble des droites vectorielles de $\overline{\mathbb{K}}^3$. Pour tout élément (X,Y,Z) non nul de $\overline{\mathbb{K}}^3$, $[X,Y,Z]$ désignera sa classe d'équivalence et P un point de $\mathbb{P}^2$ de coordonnées (X,Y,Z). Posons

$$V = \{[X,Y,Z] \in \mathbb{P}^2(\overline{\mathbb{K}}) | \, Z \neq 0\}.$$

Considérons l'application $\psi : V \to \overline{\mathbb{K}}^2$ définie par

$$\psi([X,Y,Z]) = \left(\frac{X}{Z}, \frac{Y}{Z}\right)$$

ψ est une bijection dont l'application réciproque est donnée par la formule

$$\psi^{-1}(x,y) = [x,y,1];$$

avec $x = \frac{X}{Z}$ et $y = \frac{Y}{Z}$. Les coordonnées non homogènes (x,y) sont appelées coordonnées affines du point P tandis que (X,Y,Z) sont des coordonnées homogènes qui sont aussi appelées coordonnées projectives du point P.

Définition 4. *Une courbe elliptique E définie sur un corps $\mathbb{K}$ de caractéristique quelconque est une courbe projective lisse d'équation de Weierstrass :*

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3, \tag{1.1.1}$$

où les a_i sont dans $\mathbb{K}$, à laquelle on adjoint un point spécial $\mathcal{O}$ appélé point à l'infini.

Remarque 2. *1. La lissicité de cette courbe se traduit par le fait qu'il n'existe pas de point $[X_0,Y_0,Z_0] \in \mathbb{P}^2$ tel que, en posant*

$$F(X,Y,Z) = Y^2Z + a_1XYZ + a_3YZ^2 - (X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3),$$

on ait

$$F(X_0,Y_0,Z_0) = \frac{\partial F}{\partial X}(X_0,Y_0,Z_0) = \frac{\partial F}{\partial Y}(X_0,Y_0,Z_0) = \frac{\partial F}{\partial Z}(X_0,Y_0,Z_0) = 0.$$

2. *L'ensemble des points* $[X, Y, Z] \in E$ *tels que* $Z = 0$ *est réduit au singleton* $\{\mathcal{O}\}$ *où*

$$\mathcal{O} = [0, 1, 0].$$

Il est le seul point à l'infini et il n'est pas singulier car $\frac{\partial F}{\partial Z}(0, 1, 0) = 1 \neq 0$.

3. *Lorsque la caractéristique de* $\mathbb{K}$ *est différente de* 2 *ou* 3, *la courbe* E *d'équation (1.1.1) est «isomorphe sur* $\mathbb{K}$*» à une courbe d'équation de Weierstrass réduite ([Sil09],Chap III §1) de la forme*

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{1.1.2}$$

où a *et* b *sont des éléments de* $\mathbb{K}$ *vérifiant*

$$4a^3 + 27b^2 \neq 0.$$

4. *Les coordonnées non homogènes* (x, y) *sont utilisées pour simplifier l'expression de l'équation (1.1.1) et de l'équation (1.1.2) respectivement :*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1.1.3}$$

$$y^2 = x^3 + ax + b \tag{1.1.4}$$

Exemple 1. *La courbe elliptique E définie sur $\mathbb{R}$ par l'équation de Weirstrass $y^2 = x^3 - x$ a pour représentation graphique la figure suivante :*

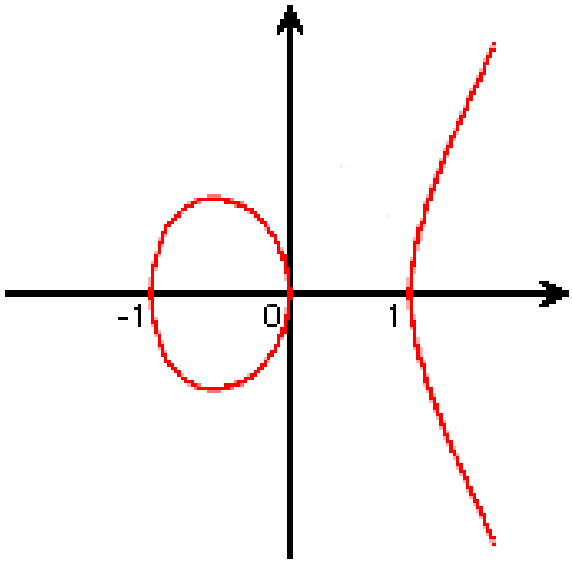

FIGURE 1.1 – Représentation graphique de E

A présent, la définition d'une courbe elliptique définie sur un corps fini est la suivante :

Définition 5. *Soit $\mathbb{F}_q$ un corps fini où $q = p^m$ avec $p \geq 5$ un nombre premier, $m \geq 1$ et soit $a,b \in \mathbb{F}_q$ tels que $4a^3 + 27b^2 \neq 0$. Une courbe elliptique E définie sur $\mathbb{F}_q$, notée $E(F_q)$ est l'ensemble des points $P = (x, y)$ de $\mathbb{F}_q^2$ vérifiant l'équation (1.1.4) plus le point à l'infini $\mathcal{O}$.*

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q \text{ tels que } y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}. \qquad (1.1.5)$$

Exemple 2. *La figure suivante illustre la courbe elliptique E définie sur le corps fini $\mathbb{F}_{127}$ et d'équation de Weierstrass $y^2 = x^3 - x + 3$.*

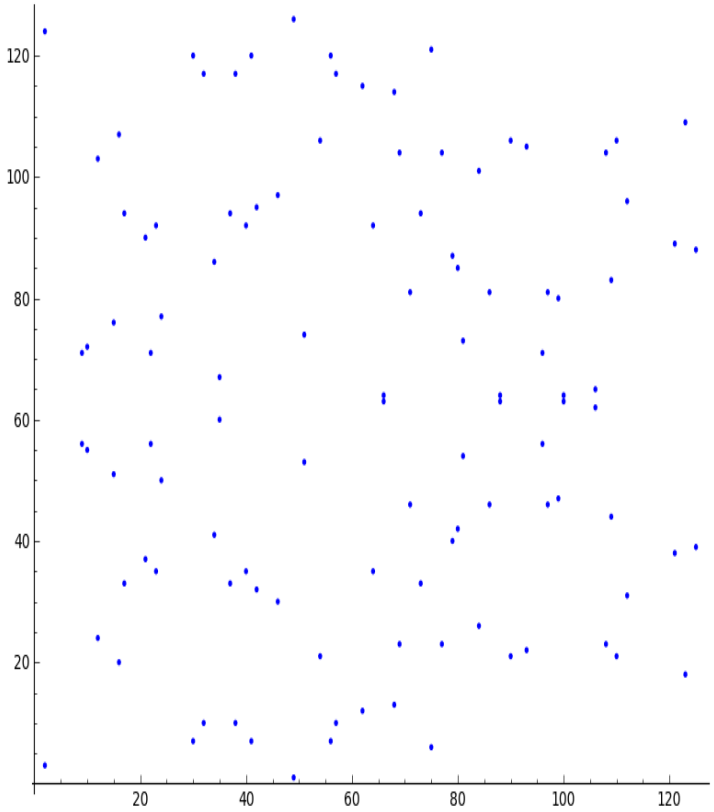

FIGURE 1.2 – $E(\mathbb{F}_{127})$ d'équation : $y^2 = x^3 - x + 3$

Dans le reste de ce manuscrit, toutes les courbes elliptiques utilisées seront définies sur des corps de caractéristique distincte de 2 et de 3. Il est important de rappeler qu'à une courbe elliptique est associée deux invariants qui sont le discriminant et le j-invariant.

Définition 6. *Soit E une courbe elliptique d'équation de Weierstrass $y^2 = x^3 + ax + b$. Il existe deux invariants, à savoir :*

$$\textit{le discriminant :}\ \ \Delta = -16(4a^3 + 27b^2)$$

$$\textit{le } j\textit{-invariant}\ \ j = -1728\frac{(4a^3)}{\Delta}.$$

Lorsque le discriminant de la courbe est non nul, alors la courbe est non singulière. En outre, le discriminant et le j-invariant permettent de classifier des courbes elliptiques : voir [Sil09, Chap III Prop. 1.4] pour de plus amples informations.

La structure de groupe d'une courbe elliptique

Soit E une courbe elliptique définie sur un corps $\mathbb{K}$. Sur E, une loi de composition interne va être définie à l'aide du théorème suivant connu sous le nom « règle de sécante tangente» :

Théorème 3. ***Règle de sécante tangente***
Soient E une courbe elliptique et L une droite, toutes deux définies sur un corps $\mathbb{K}$. Si L coupe E en deux points (comptés avec leur multiplicité) alors L coupe E en trois points (comptés avec leur multiplicité).

De ce théorème une loi d'addition, notée +, peut être définie sur l'ensemble des points de E de la manière suivante : Soit $P, Q \in E$ et L la droite passant par P et Q (si $P = Q$, L est la tangente de E au point P). L coupe E en un troisième point R' et le symétrique de R' par rapport à l'axe des abscisses est $P + Q$, comme l'illustre la figure ci-dessous lorsque $\mathbb{K} = \mathbb{R}$.

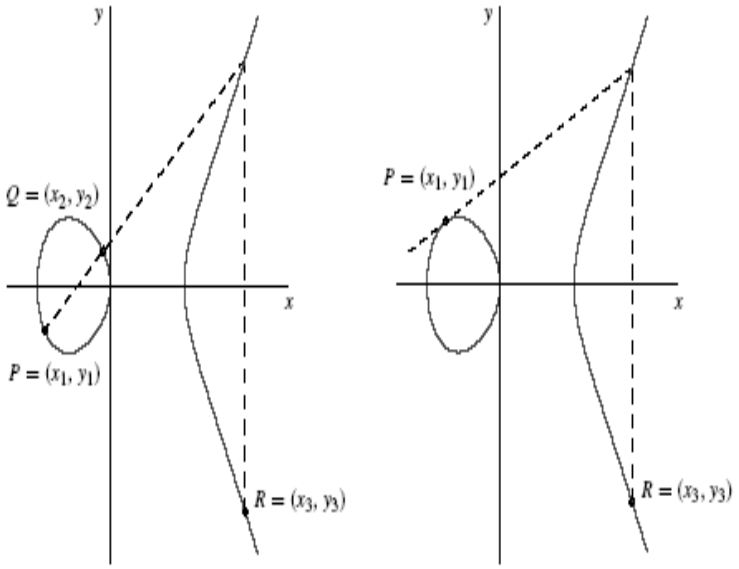

FIGURE 1.3 – Addition et doublement de points d'une courbe elliptique sur l'ensemble des nombres réels $\mathbb{R}$.

Proposition 1. *([Sil09], III.2 Prop 2.2)*
La loi + définie ci-dessus a les propriétés suivantes :

(a) Soient L une droite et P, Q et R les points d'intersection de L et E. Alors $(P + Q) + R = \mathcal{O}$.

(b) $P + Q = Q + P$, pour tout $P, Q \in E$.

(c) Pour tout $P \in E$, on a $P + \mathcal{O} = P$.

(d) Soit $P \in E$. Il existe un point dans E, noté $-P$, tel que $P+(-P) = \mathcal{O}$.

(e) Soient P, Q, $R \in E$. Alors, $(P+Q)+R = P+(Q+R)$.

$E(\mathbb{K})$ est un groupe abélien additif d'élément neutre $\mathcal{O}$.

A présent, il est nécessaire de donner les formules explicites permettant de calculer les coordonnées des points $P+Q$ et $-P$ pour tous points P et Q d'une courbe elliptique $E(\mathbb{K})$ donnée par une équation de Weierstrass.

Définition 7. *Soient $E(\mathbb{K})$ une courbe elliptique d'équation de Weierstrass $y^2 = x^3+ax+b$, P et Q deux points de $E(\mathbb{K})$ de coordonnées non homogènes (x_P, y_P) et (x_Q, y_Q) respectivement. En posant (x_R, y_R) les coordonnées non homogènes de $R = P+Q \in E(\mathbb{K})$, on a :*

— *Si $P \neq -Q$, alors on a :* $\begin{cases} \lambda = \frac{y_P - y_Q}{x_P - x_Q} \\ x_R = \lambda^2 - x_P - x_Q \\ y_R = \lambda(x_p - x_R) - y_P \end{cases}$

— *Si $P = Q$ alors $R = [2]P$ et* $\begin{cases} \lambda = \frac{3x_P^2 + a}{2y_P} \\ x_R = \lambda^2 - 2x_P \\ y_R = \lambda(x_p - x_R) - y_P \end{cases}$

— *Si $P = -Q$ alors* $\begin{cases} x_P = x_Q \\ y_P = -y_Q \end{cases}$

Notons que le coût de l'addition de deux points de la courbe elliptique $E(\mathbb{K})$ est de $1I + 2M + 1S$, où I (respectivement M et S) représente le coût d'une inversion (respectivement d'une multiplication et d'un carré) dans le corps $\mathbb{K}$. Et le coût d'un doublement est de $1I + 2M + 1S$.

Il est important de préciser quelques notations qui seront utilisées dans tout ce manuscrit.

Notations 1.

1. *Pour un entier i quelconque, M_i, S_i, I_i et A_i désigneront respectivement le coût d'une multiplication, d'un carré, d'une inversion et d'une addition dans le corps $\mathbb{F}_{p^i}$.*
2. *Si la longueur en bits de p est c, alors m_c, s_c, i_c, a_c désigneront respectivement le coût d'une multiplication, d'un carré , d'une inversion et d'une addition dans le corps $\mathbb{F}_p$.*

Dans toute la suite de ce manuscrit, p désignera toujours un entier premier. Les formules d'addition et de doublement de point présentées à la définition 7 restent valables sur le corps $\mathbb{F}_q$. Dans tout ce qui suit, les courbes elliptiques seront considérées uniquement définies sur $\mathbb{F}_q$, avec $q = p^m$, $p > 3$ et $m \geq 1$ un entier.

La définition 7 donne les formules d'addition et de doublement de point en coordonnées affines. Ce système de coordonnées fait apparaître des inversions dans le corps $\mathbb{F}_q$. Or, l'inversion est une opération coûteuse sur les

corps finis car elle est complexe. Par ailleurs, il existe d'autres systèmes de coordonnées tels que les coordonnées projectives ou jacobiennes présentant l'avantage qu'il n'y a pas d'inversion à faire lors d'une addition ou d'un doublement de point. Il est donc préférable dans certaines situations d'utiliser ces autres systèmes de coordonnées pour optimiser les calculs liés aux courbes elliptiques.

Dans ce manuscrit, nos travaux se sont focalisés essentiellement sur des courbes elliptiques définies par une équation de Weierstrass de la forme $y^2 = x^3 + b$. Ainsi les formules d'addition et de doublement de point données en coordonnées projectives utilisent de telles courbes. Pour déterminer les formules de l'addition et du doublement de point pour les coordonnées projectives, il suffit de remplacer dans les formules d'addition et doublement en coordonnées affines, les coordonnées x et y par leurs équivalents $\frac{X}{Z}$, $\frac{Y}{Z}$. Soit $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ deux points d'une courbe elliptique en coordonnées affines. Après remplacement et développement, les formules de doublement et d'addition sont les suivantes :

Doublement de point : $R = 2[P] = (x_R, y_R)$; λ devient une fraction en X_P, Y_P, Z_P en réduisant les dénominateurs :

$$\lambda = \frac{3X_P^2}{2Y_P Z_P}.$$

L'égalité $x_R = \lambda^2 - 2x_P$ devient :

$$\frac{X_R}{Z_R} = \frac{2X_P Y_P^3 - 18bX_P Y_P Z_P^3}{8Y_P^3 Z_P},$$

et permet d'exprimer X_R et Z_R en fonction de X_P, Y_P, Z_P :

$$\begin{cases} X_R &= 2X_P Y_P^3 - 18bX_P Y_P Z_P^3 \\ Z_R &= 8Y_P^3 Z_P \end{cases}$$

Pour exprimer Y_R en fonction de X_P, Y_P, Z_P, il suffit de partir de l'égalité $y_R = \lambda(x_p - x_R) - y_P$. Après remplacement et developpement,

$$Y_R = Y_P^4 + 18bY_P^2 Z_P^2 - 27b^2 Z_P^4.$$

Les formules du doublement de point en coordonnées projectives sont :

$$A = Y_P^2, \quad B = 3bZ_P^2, \quad C = (X_P + Y_P)^2 - X_P^2 - Y_P^2, \quad D = (Y_P + Z_P)^2 - Y_P^2 - Z_P^2.$$

$$\begin{cases} X_R &= C.(A - 3B) \\ Y_R &= (A + 3B)^2 - 3(2B)^2 \\ Z_R &= 4Y_P^2.D \end{cases}$$

Ainsi, cette opération de doublement de point s'effectue en $2M_m + 5S_m$ en supposant que X_P^2 et Z_P^2 sont précalculés.

Addition de deux points : $R = P + Q = (x_R, y_R)$. En suivant une démarche analogue à celle du doublement de point, les formules d'addition de deux points $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ en coordonnées projectives sont les suivantes :

$$\text{Soient } A = X_P.Z_Q, \quad B = X_Q.Z_P, \quad C = B - A, \quad D = Y_P.Z_Q, \quad E = Y_Q.Z_P,$$

$$F = E - D, \quad G = Z_P.Z_Q, \quad H = C^2, \quad I = C.H, \quad J = G.F^2.$$

$$\begin{cases} X_R &= C.[(A+B).H - J] \\ Y_R &= F.[H.(2A+B) - J] - D.H \\ Z_R &= -G.I \end{cases}$$

Ainsi, cette opération d'addition de deux points s'effectue en $13M_m + 2S_m$. Pour obtenir les formules d'addition et de doublement de point en coordonnées jacobiennes, il suffit d'appliquer une démarche analogue à celle des coordonnées projectives.

Dans la pratique, il est essentiel de connaître le nombre précis de points d'une courbe elliptique utilisée. Le théorème de Hasse est le tout premier résultat donnant un intervalle dans lequel se trouvent toutes les valeurs possibles du cardinal de la courbe elliptique $E(\mathbb{F}_q)$ avec $q = p^m$, $p > 3$ et $m \geq 1$ un entier. On le notera $\#E(\mathbb{F}_q)$.

Théorème 4. ***Théorème de Hasse***
Soit E une courbe elliptique sur un corps $\mathbb{F}_q$. Alors

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

La valeur $\#E(\mathbb{F}_q) - (q+1)$ est appelée la trace de Frobenius et est notée t. Par conséquent, le cardinal de $E(\mathbb{F}_q)$ est donné par :

$$\#E(\mathbb{F}_q) = q + 1 - t.$$

En fait, comme le rappelle [Was08, §4.2], le Frobenius est un morphisme de la courbe elliptique :

Définition 8. *Le* ***Frobenius****, noté π_p, est une application de la courbe elliptique $E(\mathbb{F}_q)$ d'équation $y^2 = x^3 + ax + b$ sur la courbe elliptique notée $E^{(p)}(\mathbb{F}_q)$ et d'équation $y^2 = x^3 + a^p x + b^p$. Il est défini comme suit :*

$$\begin{array}{rcl} \pi_p : E(\mathbb{F}_q) & \to & E^{(p)}(\mathbb{F}_q) \\ P = (x_P, y_P) & \mapsto & \pi_p(P) = (x_P^p, y_P^p) \end{array}$$

Dans le but de connaître le nombre exact de points d'une courbe elliptique définie sur un corps fini $\mathbb{F}_q$, plusieurs algorithmes de comptage ont été construits. En 1985, Schoof [Sch85] proposa le premier algorithme de comptage de points d'une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ plus efficace que les algorithmes existants, de complexité polynômiale en $log^8 q$ opérations. Les travaux d'Atkin, Elkies, puis ceux de Couveignes aboutissent à un algorithme de complexité en $O(log^5 q)$ opérations. Enfin Schoof, Elkies et Atkin proposent un algorithme d'une complexité en $O(log^4 q)$ opérations et implémenté dans plusieurs logiciels de calculs formels à l'instar de Magma et pariGP. Pour plus de détails sur l'algorithme de Schoof, ses améliorations ainsi que d'autres méthodes de comptage, consulter [BSS00, Sch95, Ked03, Sat02].

A partir de la connaissance du cardinal des courbes elliptiques définies sur un corps fini $\mathbb{F}_q$, il est donc possible de distinguer deux types de courbes elliptiques, à savoir **les courbes supersingulières** et **les courbes ordinaires** [Sil09, Chap V §3 et 4] .

Définition 9. *Une courbe elliptique $E(\mathbb{F}_q)$ de cardinal $q+1-t$ avec $q = p^m$, $p > 3$ et $m \geq 1$ un entier est dite* ***supersingulière*** *si l'une des conditions équivalentes suivantes est satisfaite :*

— $\#E(\mathbb{F}_q) \equiv 1$ *mod(p), ou* $\#E(\mathbb{F}_q) = q+1$ *mod(p), ou* $t = 0$,

— *E n'admet pas de point d'ordre p sur* $\overline{\mathbb{F}_q}$.

Une courbe non supersingulière est dite ***ordinaire****.*

Dans les chapitres suivants, toutes les courbes elliptiques considérées sont des courbes ordinaires.

1.1.3 Notion de diviseur

Dans cette partie, il est rappelé la définition de diviseur associé aux points d'une courbe elliptique ainsi que ses propriétés caractéristiques.

Définition 10. *Soit E une courbe elliptique. Un diviseur de E est une somme formelle de points*

$$D = \sum_{\substack{P \in E \\ n_P \in \mathbb{Z}}} n_P(P)$$

pour laquelle un nombre fini de n_P sont non nuls.

A un diviseur, on associe :

— **son support :** $supp(D) = \{P \in E, n_P \neq 0\}$,

— **son ordre en un point** P **:** $ord_P(D) = n_P$,

— **son degré :** $deg(D) = \sum_{P \in E} n_P$.

L'ensemble des diviseurs de degré zéro est noté $Div^0(E)$.

Notations 2. *Pour tout point P d'une courbe elliptique, (P) désignera son diviseur associé.*

Remarque 3. *Il est important de différencier la notion de «somme formelle» qui intervient dans la définition de diviseur d'une courbe elliptique de celle de «somme de points» d'une courbe. En effet, $n_P(P)$ est le diviseur représentant n_P fois le point P ; tandis que $[n_P]P$ est le point obtenu en additionnant n_P fois le point P sur la courbe elliptique.*

Proposition 2. *L'ensemble des diviseurs d'une courbe elliptique E est un groupe additif pour la loi d'addition suivante : pour tous diviseurs $D = \sum_{\substack{P\in E\\ n_P\in\mathbb{Z}}} n_P(P)$ et $D' = \sum_{\substack{P\in E\\ n'_P\in\mathbb{Z}}} n'_P(P)$ d'une même courbe E, on a :*

$$D + D' = \sum_{\substack{P\in E\\ (n_P,n'_P)\in\mathbb{Z}^2}} (n_P + n'_P)(P)$$

Nous rappelons la définition de quelques notions liées aux fonctions rationnelles qui nous serons utiles dans la suite.

Définition 11. *[Sil09, Chap.2] Soit P un point d'une courbe elliptique $E(\mathbb{F}_q)$. Il existe une fonction u_P appelée **l'uniformisante en** P, s'annulant en P et telle que toute fonction rationnelle f puisse s'écrire $f = u_P^\alpha \times h$, pour α un entier naturel et où h est une fonction rationnelle telle que $h(P) \neq 0, \infty$.*

Remarque 4. *Pour les courbes elliptiques d'équation $y^2 = x^3 + ax + b$, une uniformisante au point à l'infini $\mathcal{O}$ est $u_\mathcal{O} = \frac{x}{y}$ (cf. [Was08, Chap 11]).*

Définition 12. ***l'ordre d'un polynôme** g **en un point P** est le degré du polynôme g_0 de plus petit degré permettant d'écrire la décomposition $g = g_0 \times g_1$, avec $g_0(P) = 0$ et $g_1(P) \neq 0$. On le note $ord_P(g)$.*

Définition 13. *Pour toute fonction rationnelle f sur E, le diviseur $Div(f)$ de f est définie par $Div(f) = \sum_{P\in E} ord_P(f)(P)$. Un diviseur D est dit **principal** s'il existe une fonction rationnelle f vérifiant $D = Div(f)$.*

Proposition 3. *Soient f_1 et f_2 deux fonctions rationnelles ; alors*

$$Div(f_1 \times f_2) = Div(f_1) + Div(f_2)$$

et

$$Div(\frac{f_1}{f_2}) = Div(f_1) - Div(f_2).$$

Proposition 4. *Un diviseur $D = \sum_{\substack{P\in E\\ n_P\in\mathbb{Z}}} n_P(P)$ est principal si et seulement si*

$$deg(D) = 0 \quad et \quad \sum_{\substack{P\in E\\ n_P\in\mathbb{Z}}} [n_P]P = \mathcal{O}.$$

1.1.4 Définition du logarithme discret

La sécurité d'un protocole cryptographique peut reposer sur le problème du logarithme discret qui se traduit en anglais «discret logarithm problem» abrégé (DLP). La difficulté à résoudre le DLP repose d'une part sur le choix du groupe, notamment sur son cardinal. Ceci découle d'une conséquence de la réduction de Pohlig-Hellman qui stipule que : résoudre le DLP dans un groupe d'ordre m est aussi difficile que de le résoudre dans un groupe d'ordre r, où r est le plus grand diviseur premier de m. D'autre part, cette difficulté à résoudre le DLP, trouve aussi son origine dans la structure du groupe utilisé, puisque cette dernière peut rendre ou non certaines attaques possibles. Une courbe elliptique serait un bon choix de groupe convenable pour des applications cryptographiques dont la sécurité repose sur le DLP. En effet, l'arithmétique qu'offre la structure de groupe d'une courbe elliptique est plus efficace. Il n'y a pas de meilleure attaque connue sur elle pour résoudre le DLP autre que les attaques génériques ; les clés de chiffrement sont faciles à générer et plus petites qu'avec RSA. En pratique, on a besoin d'un groupe avec un nombre fini d'éléments. C'est pourquoi, on utilise les courbes elliptiques définies sur un corps fini, habituellement sur un corps premier ou sur le corps fini $\mathbb{F}_{2^n}$.

Le problème du logarithme discret dans le groupe des points rationnels d'une courbe elliptique est le suivant :

Définition 14. *Soit $E(\mathbb{F}_q)$ une courbe elliptique définie sur le corps fini $\mathbb{F}_q$. Le **problème du logarithme discret** (DLP) sur $E(\mathbb{F}_q)$ consiste à trouver x à partir de $[x]P$ où P est un point de la courbe elliptique.*

Dans un contexte plus général, la difficulté théorique du problème du logarithme discret est justifiée par le théorème de Shoup [Sho97] :

Théorème 5. *Dans un groupe générique dont l'ordre est un nombre premier noté n ; la résolution du problème du logarithme discret nécessite au moins $\sqrt{n}$ opérations.*

1.2 Définition et Propriétés des couplages

Les couplages sont des applications bilinéaires non-dégénérées qui jouent un rôle important pour les protocoles cryptographiques actuels à l'instar de l'échange tripartite de clés de Joux. Les couplages sont utilisés en cryptographie sous deux facettes : dans un premier temps, ils ont été utilisés d'un point de vue destructif car ils ont permis de réduire des problèmes difficiles en des problèmes faciles. En effet, en 1993, Menezes, Okamoto et Vanstone ont montré que le couplage de Weil, défini à la section 1.2.1 permet de réduire le DLP de la courbe elliptique vers le DLP dans le groupe multiplicatif

d'un corps fini ; dans un second temps, ils ont été utilisés d'un point de vue constructif car ils ont permis de construire des protocoles originaux tels que les protocoles cryptographiques basés sur l'identité [BF01, Coc01, LQ04] et de simplifier certains protocoles cryptographiques : l'échange tripartite d'Antoine Joux [Jou00] et les schémas de signature courte [BLS04b] en sont des exemples.

Définition 15. *Soient trois groupes G_1, G_2, G_3 de même ordre. On suppose que G_1 et G_2 sont additifs et G_3 multiplicatif.*

Un couplage est une application $e : (G_1, +) \times (G_2, +) \longrightarrow (G_3, \times)$ vérifiant les propriétés suivantes :

1. *Bilinéarité : $\forall P, P' \in G_1,\ \forall Q, Q' \in G_2$*
   - $e(P + P', Q) = e(P, Q)e(P', Q)$
   - $e(P, Q + Q') = e(P, Q)e(P, Q')$
2. *Non Dégénérescence :*
   - $\forall P \in G_1,\ \exists Q \in G_2$ *tel que* $e(P, Q) \neq 1$
   - $\forall Q \in G_2,\ \exists P \in G_1$ *tel que* $e(P, Q) \neq 1$
3. *facilement calculable.*

Propriété : Soit $e : (G_1, +) \times (G_2, +) \longrightarrow (G_3, \times)$ un couplage et soient $P \in G_1$, $Q \in G_2$. Alors $\forall n \in \mathbb{N}$, $e(nP, Q) = e(P, Q)^n = e(P, nQ)$.

Cette importante propriété est facile à vérifier et repose sur la bilinéarité de e. Elle a permis l'utilisation des couplages en cryptologie.

Nous donnons à présent la définition des notions de **point de torsion** et de **degré de plongement** qui sont liées à une courbe elliptique et interviennent dans la définition des couplages dans la pratique.

Définition 16. *Soient E une courbe elliptique définie sur un corps $\mathbb{F}_q$ et l un entier.*

1. *Un point P de la courbe elliptique $E(\overline{\mathbb{F}_q})$ est un point de l-torsion si $[l]P = \mathcal{O}$.*
2. *l'ensemble des points de l-torsion de E est noté $E[l]$.*

Définition 17. *Soient $E(\mathbb{F}_q)$ une courbe elliptique et r un diviseur premier de $\#E(\mathbb{F}_q)$. Le dégré de plongement de la courbe elliptique E relativement à r est le plus petit entier k tel que r divise $q^k - 1$. Autrement dit, k est l'ordre de q dans $\mathbb{F}_r$.*

Le degré de plongement d'une courbe elliptique est un paramètre important permettant de déterminer le groupe dans lequel se trouvent tous les points de l-torsion de E [BK98].

Théorème 6. *[BK98]*
Soient $E(\mathbb{F}_q)$ une courbe elliptique et r un diviseur de $\#E(\mathbb{F}_q)$. Si r est premier avec q et r ne divise pas $q-1$, alors $E[r] \subset E(\mathbb{F}_{q^n})$ pour n un entier positif si et seulement si r divise $q^n - 1$.

Dans la pratique, les groupes G_1, G_2 et G_3 sont d'ordre r et G_3 est un sous groupe de $\mathbb{F}^*_{q^k}$. Pour que les couplages soient faciles à calculer, il faut que k ne soit pas trop grand. Or pour une courbe aléatoire, k est assez grand de l'ordre de r [BK98] et il est difficile de générer une courbe pour laquelle la valeur de k est petite. Pour utiliser le couplage en cryptologie, il faut pouvoir générer des courbes adaptées au couplage (la génération de telles courbes demande d'utiliser la méthode de la multiplication complexe (CM) dans la majorité des cas), ou bien utiliser une courbe supersingulière. Ainsi, les types de courbes adaptées au couplage sont : supersingulières, Cocks-Pinch, Dupond-Enge-Morain, creuses (par exemple les courbes Miyagi-Nakabayashi-Takano) et complètes (par exemple les courbes Barreto-Naehrig). Freeman et al. [FST10] ont proposé la taxonomie des courbes elliptiques adaptées au couplage qu'ils ont baptisé **les courbes elliptiques bien couplées**, en anglais « pairing-friendly elliptic curves». Et ils ont proposé la définition suivante :

Définition 18. *Une courbe elliptique ordinaire E définie sur $\mathbb{F}_q$ est dite bien couplée si :*

- *r, le plus grand diviseur premier de l'ordre de $E(\mathbb{F}_q)$, est tel que $r \geq \sqrt{q}$.*
- *k, le degré de plongement, est tel que $k \leq \frac{log_2(r)}{8}$.*

La démarche de construction, en général, s'effectue en deux étapes [FST10] :

1. la résolution du système

$$\begin{cases} r \mid q+1-t \\ r \mid q^k - 1 \\ Dy^2 = 4q - t^2, \quad \text{pour un entier donné } D \end{cases} \tag{1.2.1}$$

   dont les inconnus sont : le degré de plongement k, un diviseur premier r de $\#E(\mathbb{F}_q)$, la trace t de Frobenius sur $E(\mathbb{F}_q)$ et la taille q du corps fini et un entier D appelé «discriminant de la multiplication complexe i.e. l'unique entier tel que $\frac{4q-t^2}{D}$ soit un carré parfait».
2. la construction d'une courbe elliptique dont les paramètres sont les valeurs de q, r et t trouvés à la première étape en utilisant la méthode de la multiplication complexe.

Parmi ces méthodes de construction des courbes elliptiques, on distingue celles pour lesquelles k est donné à l'avance (par exemple MNT) et celles pour lesquelles k n'est pas connu à l'avance (par exemple Cocks-Pinch).

Il est intéressant de relever que chacun des types de courbes sus-cités présente des faiblesses qui proviennent, soit de sa conception, soit des récentes avancées dans la résolution du DLP sur les corps finis [Bar16] :
En ce qui concerne **les courbes supersingulières :** leur utilisation est proscrite du fait qu'il existe à l'heure actuelle des algorithmes de complexité quasi polynomiale pour la résolution du DLP sur les corps de caractéristique 2 et 3. De plus en caractéristique $p \geq 5$, il existe des méthodes de calcul qui excèdent en rapidité comparativement au cas de corps premier.
Les courbes du type Cocks-Pinch [CP01] : Avec une grande probabilité, l'entier y du système (1.2.1) a la même taille en bits que r et par conséquent $\log_2 q \approx 2\log_2 r$; ce qui affecte l'efficacité des couplages.
Les courbes du type Dupont-Enge-Morain [DEM05] : Le nombre total de courbes qui peuvent être construites pour des tailles cryptographiques est très petit si on se restreint aux courbes $E(\mathbb{F}_q)$ pour lesquelles $\#E(\mathbb{F}_q)$ et $2(q+1) - \#E(\mathbb{F}_q)$ ont de grands facteurs premiers.
Les courbes creuses (par exemple MNT [MNT01]) : l'ensemble des valeurs possibles que peut prendre le degré de plongement k est très petit, et toutes ces valeurs sont divisibles soit par 2 soit par 3.
Les courbes complètes (par exemple BN [BN05]) : Les nombres premiers construits par cette méthode sont $2\varphi(k)$-SNFS et donc les attaques NFS ont une petite complexité asymptotique.

1.2.1 Les couplages les plus utilisés en cryptographie

Soit $E : y^2 = x^3 + ax + b$ une courbe elliptique définie sur un corps $\mathbb{F}_q$ de caractéristique $p > 3$, r un grand diviseur premier de l'ordre du groupe des points rationnels de la courbe elliptique tel que r^2 ne divise pas le cardinal de $\#E(\mathbb{F}_q)$ afin d'éviter que tous les points r-tosion soient dans $E(\mathbb{F}_q)$. Soit $E[r]$ l'ensemble des points de r-torsion et k le degré de plongement de E relativement à q et r, on a $E[r] \subset E(\mathbb{F}_{q^k})$ lorsque $k > 1$. On désigne par μ_r l'ensemble des racines r-ièmes de l'unité dans $\mathbb{F}_{q^k}^*$. Avec ces notations, on a les définitions suivantes :

Définition 19. *Soient $S \in E(\mathbb{F}_{q^k})$ et $n \in \mathbb{Z}$. Une fonction de Miller $f_{n,S}$ [Mil04] de longueur n est une fonction dans $\mathbb{F}_{q^k}(E)$ de diviseur $Div(f_{n,S}) = n(S) - ([n]S) - (n-1)(\mathcal{O})$.*

Définition 20. *Le **couplage de Weil** [Wei48], noté e_W, est l'application bilinéaire et non dégénérée définie par :*

$$e_W : E[r] \times E[r] \longrightarrow \mu_r \quad (P, Q) \mapsto (-1)^r \frac{f_{r,P}(Q)}{f_{r,Q}(P)}.$$

Définition 21. *Le **couplage de Tate réduit**, noté e_r, est l'application*

bilinéaire et non dégénérée définie par

$$e_r : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})[r] \to \mu_r, (P,Q) \mapsto f_{r,P}(Q)^{\frac{q^k-1}{r}}.$$

Remarque 5. *1. La non-dégénérescence des couplages de Weil et Tate est satisfaite si le point Q n'appartient pas au sous groupe de $E[r]$ engndré par P.*

2. Une définition plus générale du couplage de Weil se trouve dans le livre [Wei48].

3. La définition complète du couplage de Tate est donné dans [BSS05, §IX.5].

Pour définir des versions améliorées du couplage de Tate appelées couplage de Ate et couplage Twisted Ate [HSV06], on considère $\pi_q : E\left(\overline{\mathbb{F}_q}\right) \to E\left(\overline{\mathbb{F}_q}\right), (x,y) \mapsto (x^q, y^q)$ l'endomorphisme de Frobenius où $\overline{\mathbb{F}_q}$ est la clôture algébrique du corps fini $\mathbb{F}_q$. La relation entre la trace t de l'endomorphisme de Frobenius et l'ordre du groupe est donnée par [Was08, Theorem 4.3] : $\#E(\mathbb{F}_q) = q+1-t$ et π_q a exactement deux valeurs propres 1 et q. Ceci nous permet de considérer $P \in \mathbb{G}_1 = E\left(\overline{\mathbb{F}_q}\right)[r] \cap \mathrm{Ker}(\pi_q - [1]) = E(\mathbb{F}_q)[r]$ et $Q \in \mathbb{G}_2 = E\left(\overline{\mathbb{F}_q}\right)[r] \cap \mathrm{Ker}(\pi_q - [q])$.

Définition 22. *1. Le **couplage Ate** réduit, noté e_A, est l'application bilinéaire et non dégénérée définie comme suit :*

$$e_A : \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r, (Q,P) \mapsto f_{t-1,Q}(P)^{\frac{q^k-1}{r}}.$$

2. Soit E une courbe elliptique admettant une tordue de degré d, $m = pgcd(k,d)$ et $e = k/m$.
*Le **couplage Twisted Ate** réduit, noté e_{TA}, est l'application :*

$$e_{TA} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mu_r, (Q,P) \mapsto f_{T^e,P}(Q)^{\frac{q^k-1}{r}}.$$

où $T = t-1$, t désignant la trace de Frobenius sur E.

Pour montrer que l'application e_A (respectivement e_{TA}) est bien bilinéaire et non dégénérée, il suffit d'utiliser la relation liant le couplage Ate (respectivement le couplage Twisted Ate) au couplage de Tate.

1.2.2 Couplage optimal

La notion de couplage optimal introduit par Vercauteren [Ver10] permet de calculer un couplage en utilisant des fonctions de Miller dont chacune d'elles a une longueur approximativement égale à $\log_2 r/\varphi(k)$, où φ est la fonction indicatrice d'Euler.

Définition 23. *Soient $S \in E[r]$, $n \in \mathbb{Z}$ et un polynôme $h(z) = \Sigma h_i z_i \in \mathbb{Z}[z]$ tel que $h(n) \equiv 0 \pmod r$.*
La fonction de Miller étendue $f_{n,h,S}$ est la fonction rationnelle normalisée de diviseur

$$\sum_{i=0}^{deg\, h} h_i[([n^i]S) - (\mathcal{O})].$$

La longueur de $f_{n,h,S}$ est le maximum des valeurs absolues des coefficients h_i.

Lemme 1. *On a l'égalité suivante :*

$$f_{n,S} = f_{n,n-z,S}.$$

Preuve En effet, pour établir cette égalité, il suffit de montrer que $Div(f_{n,S}) = Div(f_{n,n-z,S})$. On a $h(z) = n - z$ i.e. $h_0 = n$ et $h_1 = -1$;

$$\begin{aligned} Div(f_{n,n-z,S}) &= \sum_{i=0}^{1} h_i[([n^i]S) - (\mathcal{O})] \\ &= h_0\,[(S) - (\mathcal{O})] + h_1\,[([n]S) - (\mathcal{O})] \\ &= n\,[(S) - (\mathcal{O})] - [([n]S) - (\mathcal{O})] \\ &= n(S) - n(\mathcal{O}) - ([n]S) + (\mathcal{O}) \\ &= n(S) - ([n]S) - (n-1)(\mathcal{O}) \\ &= Div(f_{n,S}). \end{aligned}$$

□

Théorème 7. ***Couplage optimal Ate*** *[Ver10]*
Il exsite un polynôme $h(z) = \Sigma h_i z_i \in \mathbb{Z}[z]$ avec $h(q) \equiv 0 \pmod r$ tel que $|h_i| \leq r^{1/\varphi(k)}$ et l'application

$$\begin{array}{rccc} e_o : & \mathbb{G}_2 \times \mathbb{G}_1 & \to & \mu_r \\ & (Q, P) & \longmapsto & f_{q,h,Q}(P)^{(q^k-1)/r} \end{array} \tag{1.2.2}$$

est un couplage.

Remarque 6. *1. Ces petits coefficients h_i peuvent être obtenus en général à partir des vecteurs courts du réseau suivant :*

$$L = \begin{pmatrix} r & 0 & 0 & \cdots & 0 \\ -q & 1 & 0 & \cdots & 0 \\ -q^2 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -q^{\varphi(k)-1} & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{1.2.3}$$

L'algorithme LLL appliqué aux lignes de L permet de trouver le vecteur le plus court de ce réseau.

2. *Le polynôme* $h(z) = \Sigma h_i z_i \in \mathbb{Z}[z]$ *avec* $h(q) \equiv 0 \pmod r$ *tel que* $|h_i| \leq r^{1/\varphi(k)}$ *du théorème précédent sera appelé polynôme de Vercauteren.*

En général, l'expression des couplages utilisés en cryptographie est donnée en terme de fonctions de Miller. Ces dernières se calculent efficacement grâce à l'algorithme de Miller [Mil04]. C'est pour cette raison que cet algorithme est utilisé pour calculer de manière efficace les couplages.

1.2.3 Calcul du couplage - algorithme de Miller

L'algorithme de Miller repose sur l'égalité de Miller : $f_{a+b,Q} = \frac{f_{a,Q} f_{b,Q} l_{[a]Q,[b]Q}}{v_{[a+b]Q}}$ où a et b sont des entiers, Q un point de la courbe elliptique, $l_{[a]Q,[b]Q}$ désigne l'équation de la droite $([a]Q[b]Q)$ et $v_{[a+b]Q}$ celle de la verticale au point $[a+b]Q$. Lorsque $a = 1$, la fonction $f_{1,Q}$ est la fonction constante égale à 1.

Exemple 3. *Nous montrons comment l'égalité de Miller est utilisée pour calculer* $f_{5,Q}$.

1. $f_{5,Q} = f_{4+1,Q} = f_{4,Q} \times f_{1,Q} \times \frac{l_{[4]Q,[1]Q}}{v_{[5]Q}} = f_{4,Q} \times \frac{l_{[4]Q,Q}}{v_{[5]Q}}$, *car* $f_{1,Q} = 1$;
2. *En décomposant* 4 *en* $4 = 2 \times 2$, *on a* $f_{4,Q} = f_{2,Q} \times f_{2,Q} \times \frac{l_{[2]Q,[2]Q}}{v_{[4]Q}}$
   $f_{4,Q} = f_{2,Q}^2 \times \frac{l_{[2]Q,[2]Q}}{v_{[4]Q}}$; *on en déduit que* $f_{5,Q} = f_{2,Q}^2 \times \frac{l_{[2]Q,[2]Q}}{v_{[4]Q}} \times \frac{l_{[4]Q,Q}}{v_{[5]Q}}$;
3. $f_{2,Q} = f_{1,Q} \times f_{1,Q} \times \frac{l_{[1]Q,[1]Q}}{v_{[2]Q}} = \frac{l_{Q,Q}}{v_{[2]Q}}$;
4. $f_{5,Q} = \left(\frac{l_{Q,Q}}{v_{[2]Q}}\right)^2 \times \frac{l_{[2]Q,[2]Q}}{v_{[4]Q}} \times \frac{l_{[4]Q,Q}}{v_{[5]Q}}$.

L'égalité de Miller permet de calculer la fonction $f_{n,Q}$ par une méthode d'exécution itérative construite sur le schéma de l'exponentiation rapide à base d'addition et de doublement.

L'algorithme de Miller utilise la méthode d'addition et doublement comme chaine d'addition pour n pour calculer $f := f_{n,Q}(P)$ en prenant en entrée $n = n_s 2^s + \cdots + n_1 2 + n_0 > 0$ avec $n_i \in \{-1, 0, 1\}$, $P \in \mathbb{G}_1$ et $Q \in \mathbb{G}_2$. Plus précisément, il se présente de la façon suivante :

Algorithme 1 : algorithme de Miller

Entrée n$=\sum_{j=0}^{s} n_j 2^j \in \mathbb{N}$, $n_j \in \{-1, 0, 1\}$, $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$
Sortie $f_{n,Q}(P) \in \mathbb{F}_{q^k}^*$, $[n]Q$
$f \leftarrow 1$, $T \leftarrow Q$
pour j ***allant de*** $s-1$ ***à*** 0 **faire**
  $f \leftarrow f^2.l_{T,T}(P)/v_{[2]T}(P)$; $T \leftarrow [2]T$ Etape de doublement
  si $n_j = 1$ **alors**
    $f \leftarrow f.l_{T,Q}(P)/v_{T+Q}(P)$; $T \leftarrow T + Q$ Etape d'addition
  fin
  si $n_j = -1$ **alors**
    $f \leftarrow f.l_{T,-Q}(P)/v_{T-Q}(P)$; $T \leftarrow T - Q$ Etape d'addition
  fin
fin
renvoie f.

Lors du calcul de $f_{n,Q}$, sa longueur n permet de déterminer à la fois le nombre d'étapes de doublement qui vaut $\lfloor \log_2 n \rfloor$, et le nombre d'étapes d'addition qui est le poids de Hamming de n dans l'algorithme de Miller.

L'utilisation des tordues permettent de faire efficacement certains calculs pendant l'exécution. Cela est bien expliqué dans la section suivante.

1.2.4 Utilisation des tordues

Les tordues d'une courbe elliptique permettent de calculer efficacement les couplages. En fait, les étapes de doublement et d'addition des points dans l'algorithme de Miller sont faites dans l'extension de corps $\mathbb{F}_{q^k}$. L'utilisation des tordues permettent d'effectuer ces opérations plutôt dans un sous corps de $\mathbb{F}_{q^k}$. Plus précisement, une tordue d'une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$ est une courbe elliptique E' définie sur $\mathbb{F}_q$ qui est isomorphe à E sur une clôture algébrique de $\mathbb{F}_q$. Le plus petit entier d tel que E et E' soient isomorphes sur $\mathbb{F}_{q^d}$ est appelé **degré de la tordue**. Dans cette thèse, nous nous focalisons sur les courbes elliptiques ordinaires de degré de plongement 9, 15 et 27 respectivement. Elles admettent des tordues d'ordre trois. Les constructions explicites de telles courbes peuvent être trouvées dans [LZZW08], [DCC05] et [BLS02]. L'équation générale de ces courbes est donnée par $E : y^2 = x^3 + b$. L'équation définissant la tordue E' a la forme $y^2 = x^3 + b\omega^6$ où $\{1, \omega, \omega^2\}$ est la base du $\mathbb{F}_{q^{k/3}}$-espace vectoriel $\mathbb{F}_{q^k}$ et l'isomorphisme entre E' et E est l'application

$$\psi : E' \longrightarrow E : (x', y') \longmapsto (x'/\omega^2, y'/\omega^3).$$

En utilisant cet isomorphisme, les points Q de $\mathbb{G}_2$ peuvent être considérés égaux à $(x\omega^2, y\omega^3) \in \mathbb{G}_2' \subset \psi^{-1}(E(\mathbb{F}_{q^{k/3}}))$ où $x, y \in \mathbb{F}_{q^{k/3}}$. D'amples détails sur les tordues peuvent être trouvés dans [CLN10].

1.2.5 Exponentiation finale et méthode des réseaux pour le calcul du couplage optimal

Nous supposons que q est un nombre premier i.e. $q = p$. **L'exponentiation finale** est l'étape consistant à élever à la puissance $\frac{p^k-1}{r}$ le résultat obtenu au terme de l'étape de la boucle de Miller lors de l'exécution de l'algorithme de Miller pour le calcul des couplages tels que celui de Tate réduit. Calculer efficacement l'exponentiation finale est devenu une tâche laborieuse. On remarque que cet exposant peut être subdivisé en deux parties comme suit :

$$\frac{p^k-1}{r} = \left[\frac{p^k-1}{\phi_k(p)}\right] \cdot \left[\frac{\phi_k(p)}{r}\right],$$

$\phi_k(x)$ étant le k-ième polynôme cyclotomique.

L'exponentiation finale est donc calculée comme $f^{\frac{p^k-1}{r}} = \left[f^{\frac{p^k-1}{\phi_k(p)}}\right]^{\frac{\phi_k(p)}{r}}$. Le calcul de la première partie $A = f^{\frac{p^k-1}{\phi_k(p)}}$ est généralement moins coûteuse car elle revient à effectuer quelques multiplications, inversions et élévations à la puissance p dans $\mathbb{F}_{p^k}$. La seconde partie $A^{\frac{\phi_k(p)}{r}}$ est la plus complexe et est appelée partie difficile. Une méthode efficace pour calculer la partie difficile est décrite par Scott *et al.* [SBC+09]. Ils suggèrent d'écrire $d = \frac{\phi_k(p)}{r}$ en base p comme $d = d_0 + d_1 p + ... + d_{\varphi(k)-1} p^{\varphi(k)-1}$ et de trouver une courte chaine d'additions vectorielles qui soit, dans le cadre du calcul de A^d, plus efficace que la méthode naive. En se basant sur le fait qu'une puissance d'un couplage en est encore un, Fuentes *et al.* suggèrent dans [FKR11] d'appliquer la méthode de Scott *et al.* avec une puissance d'un multiple quelconque d' de d qui ne possède pas r pour diviseur. Ce qui remplacerait le calcul de A^d par une exponentiation plus efficace. Trouver le polynôme $d'(x)$ revient à appliquer l'algorithme LLL à la matrice M formée par $\mathbb{Q}$-combinaisons linéaires des éléments $d(x), xd(x), ..., x^{deg(r)-1}d(x)$. Nous utiliserons cette méthode qui a connu du succès dans le cadre des courbes elliptiques de degré de plongement $8, 12$ et 18 respectivement pour améliorer, dans les sections 3.3 et 3.4, le calcul de l'exponentiation finale lorsque le degré de plongement prend la valeur 9 et la valeur 15, le cas $k = 27$ ayant déjà été traité.

1.2.6 Mise en place d'un système cryptographique à base de couplage

Le but de cette section est de préciser toutes les contraintes mathématiques à prendre en compte pour assurer la fiabilité et la sécurité d'un cryptosystème à base de couplage.

Pour qu'un cryptosystème basé sur le couplage soit sûr, il faut et il suffit que le problème du logarithme discret ne puisse être résoluble en un temps raisonnable ni dans le groupe multiplicatif de $\mathbb{F}_{q^k}$, ni dans le groupe $E(\mathbb{F}_q)$. Les couplages sont donc vulnérables à la fois à des attaques liées aux courbes elliptiques et à celles spécifiques aux corps finis. Les paragraphes suivants inventorient ces attaques présentées de manière détaillée dans [Was08, Chap V] et dans [Bar16], tout en adjoignant à chacune d'elles d'éventuelles recommandations permettant de l'éviter.

Attaques et recommandations pour le choix des courbes convenables

Les attaques génériques du logarithme discret sont des attaques valables sur un groupe quelconque. Soit E une courbe elliptique définie sur $\mathbb{F}_q$ et P un générateur de $\langle P\rangle \subset E$ d'ordre r, où r est le plus grand facteur de l'ordre de $E(\mathbb{F}_q)$. Connaissant $Q = [x]P$, il faut trouver x. Voici un listing de ces attaques génériques avec une description de chacune d'elles ainsi que d'éventuelles recommandations à suivre pour les éviter :

- **Force brute** : On calcule les points P, $[2]P$,..., jusqu'à ce qu'on trouve x. Dans le pire des cas, il faut effectuer $\boldsymbol{r}$ opérations.
- **Baby Step Giant Step, (BSGS) :** On se donne une valeur $n < r$ et on calcule les pas de bébé : P, $[2]P$, ..., $[n]P$; puis les pas de géant : $Q + [n]P$, $Q + [2n]P$,... jusqu'à avoir une collision :

$$\alpha P = Q + \beta P \Longrightarrow Q = (\beta - \alpha)P.$$

  Cette attaque a une complexité en $O(\sqrt{r})$ opérations pour $n \approx \sqrt{r}$.
- **Pollard ρ [Pol78] :** On utilise en plus une marche aléatoire (i.e. une suite $x_{i+1} = f(x_i)$ où f est une fonction aléatoire). On précalcule $l \leq 1000$ valeurs $T_i = [a_i]Q + [b_i]P$, où les $1 \leq a_i, b_i \leq r-1$ sont tirés au hasard. On part de $x_0 = [a_0]Q + [b_0]P$ et on calcule la suite $x_{i+1} = f(x_i) = x_i + T_{H(x_i)}$, où $H : E \to [1, l]$. Une collision donne

$$\alpha Q + \beta P = \alpha' Q + \beta' P \Longrightarrow x = \frac{\beta' - \beta}{\alpha - \alpha'} mod(r).$$

  Sa complexité est de $\sqrt{r}$ opérations (idem pour une parallélisation). l'algorithme parallelisé du **Pollard $\boldsymbol{\rho}$** est l'une des attaques la plus efficace connue sur $E(\mathbb{F}_q)$. Pour un niveau de sécurité n donné, il faut

prendre $\log_2 r = 2n$ et donc $\log_2 \#E(\mathbb{F}_q) \geq 2n$. D'après le théorème de Hasse, q et $\#E(\mathbb{F}_q)$ ont approximativement la même taille. Il est donc recommandé d'avoir $\mathbf{\log_2 q \geq \log_2 r = 2n}$ pour éviter cette attaque.

- **Pohlig-Hellman :** la difficulté à résoudre le DLP repose sur la taille du plus grand facteur premier de l'ordre de E. Il est préférable de choisir une courbe E telle que $\boldsymbol{\#E = r}$ soit premier ou $\boldsymbol{\#E = nr}$ ($\boldsymbol{n}$ petit).
- **Descente de Weil (attaques GHS, GTTD) [GHS02, FG98] :** Si $q = p^m$ avec m composé, alors on peut transférer le problème du logarithme discret sur un groupe où il y a des attaques sous-exponentielles. Il est donc préférable de prendre $\boldsymbol{p}$ **premier** ou $\boldsymbol{p = 2}$ et $\boldsymbol{m}$ **premier**.
- **Courbes anormales et transfert sur $\mathbb{F}_p$** : Une courbe définie sur $\mathbb{F}_p$ (avec p premier) est dite **anormale** si $\#E = p$. Dans ce cas, le DLP peut être transféré de E vers $\mathbb{F}_p$ où les attaques sous-exponentielles sont possibles. Il est donc recommandé de vérifier que $\boldsymbol{\#E \neq p}$.
- **Courbes supersingulières et transfert sur $\mathbb{F}^*_{q^k}$ (attaque MOV) [MOV93] :** On peut transférer le DLP de E sur $\mathbb{F}_{q^k}$ en utilisant les couplages, k étant le plus petit entier tel que $q^k \equiv 1 \bmod(r)$. En général, $k \cong r$ donc il est plus facile de résoudre le DLP sur E par la force brute sur E que par des méthodes sous exponentielles sur $\mathbb{F}^*_{q^k}$. Si $\#E \equiv 1 \bmod(p)$, alors E est dite supersingulière et k est petit (≤ 6). L'attaque MOV est praticable pour $k \leq 20$. Il est recommandé de vérifier que $\boldsymbol{\#E \neq 1 \mod(p)}$ et que $\boldsymbol{k > 20}$.
- **Faille sur la tordue d'une courbe elliptique [BMM00]** : Lors de certaines implémentations de la multiplication scalaire, il se peut qu'on utilise seulement l'abscisse x des points de la courbe elliptique E : $y^2 = x^3 + ax + b$. Par injection d'erreur, on peut transférer le DLP de la courbe E vers sa tordue E' : $\epsilon y^2 = x^3 + ax + b$, où ϵ n'est pas un carré dans $\mathbb{F}_q$. Il est donc recommandé d'utiliser des courbes elliptiques telles que $\boldsymbol{\#E(\mathbb{F}_q)}$ et $\boldsymbol{2(q+1) - \#E(\mathbb{F}_q)}$ **aient tous deux de grands facteurs premiers**.

Attaques spécifiques aux corps finis

Les meilleurs algorithmes de résolution du problème du logarithme discret dans le groupe multiplicatif de $\mathbb{F}_{q^k}$ sont du type «calcul d'indice» [Kra22, Kra24] et ont une complexité sous-exponentielle.

Lorsque la caractéristique du corps $\mathbb{F}_q$ n'est pas petite, les meilleures complexités sont toutes obtenues avec le même algorithme, celui du crible algébrique du corps des nombres (NFS, «Number Field Sieve» en anglais) [JL03]. Ils ont une complexité sous-exponentielle.

Les complexités asymptotiques des différentes versions du NFS dans $\mathbb{F}_{p^n}$

Rappelons que $L_{p^n}[\alpha, c] = e^{(c+O(1))(\log p^n)^{\alpha}(\log\log p^n)^{1-\alpha}}$ est la notation de la complexité asymptotique :

— Pour $\alpha = 1$, la complexité est exponentielle.

— Pour $\alpha = 0$, la complexité est polynomiale.

— Pour $0 < \alpha < 1$, la complexité est sous-exponentielle.

Les différentes variantes du NFS de meilleure complexité asymptotique sont les suivantes :

En **grande Caractéristique** : ce cas de figure n'est pas réellement utilisée en cryptographie basée sur le couplage.

Lorsque n est premier :

— si p n'a pas une forme spéciale : $L_{p^n}[1/3, (64/9)^{1/3} = 1,923]$ (**General Joux-Lercier, (GJL) [JL03]**).

— si p a une forme spéciale : $L_{p^n}[1/3, (32/9)^{1/3} = 1,526]$ (**Joux-Pierrot [JP13], SNFS**)

En **caractéristique moyenne** :

Lorsque n est premier

— p n'est pas spécial : $L_{p^n}[1/3, (96/9)^{1/3}] = 2,201]$ **(Conjugation)**

— p est spécial : $L_{p^n}[1/3, (64/9)^{1/3}] = 1,923]$ (**Joux-Pierrot** [JP13])

Lorsque n est composé : ExTNFS est le meilleur,

— p n'est pas spécial : $L_{p^n}[1/3, (48/9)^{1/3} = 1,74]$

— p est spécial : $L_{p^n}[1/3, (32/9)^{1/3} = 1,526]$.

Impact des récentes améliorations des attaques Number Field Sieve (NFS)

Lorsque la caractéristique p a une forme spéciale, par exemple un faible poids de Hamming, il existe une variante du NFS appelée **SNFS** (special Number Field Sieve) de meilleure complexité asymptotique qui permet de factoriser. C'est en 2013 qu'une version de l'algorithme SNFS de même complexité asymptotite a été proposé par Joux et Pierrot [JP13] afin de résoudre le DLP dans les corps $\mathbb{F}_{q^k}$ pour $k \neq 1$. Cependant, cet algorithme présente deux inconvénients : il n'est pas pratique et les tailles des clés n'ont pas changé. C'est dans le but de palier à ces inconvénients que Barbulescu et *al.* [BGK15] réhabiliterons en 2015 un algorithme d'Olivier Schirokauer [Sch00] et ils l'ont baptisé crible algébrique des tours d'extensions et se traduit en anglais «Tower Number Field Sieve, **TNFS**». Le nouvel algorithme, malheureusement, manque lui aussi de practicabilité ; c'est pour cette raison que Kim et Barbulescu proposerons finalement en 2016 une méthode nommée **exTNFS** combinant le TNFS et la méthode de Joux-Pierrot, afin de résoudre simultanément le problème de practicabilité et la recommandation de changer la taille des clés [KB16].

1.3 Quelques notions mathématiques générales

1.3.1 Quelques définitions sur les réseaux

Définition 24. *Un réseau L est un sous-groupe additif discret de $\mathbb{R}^m$, c'est-à-dire un ensemble de vecteurs vérifiant :*

1. *Le vecteur nul est dans le réseau.*
2. *L'opposé de tout vecteur du réseau est dans le réseau.*
3. *La somme de deux vecteurs du réseau est encore dans le réseau.*
4. *Tous les points du réseau sont isolés i.e. il existe une constante positive $\epsilon > 0$ tel que pour tout $v \in L$, $L \cap \{w \in \mathbb{R}^m / \|v - w\| < \epsilon\} = \{v\}$.*

Définition 25. *Soient $u_1, u_2, ..., u_n \in \mathbb{R}^m$ des vecteurs linéairement indépendants. Le réseau L engendré par $u_1, u_2, ..., u_n$ est l'ensemble des combinaisons linéaires à coefficients entiers, des vecteurs $u_1, u_2, ..., u_n$,*

$$L = \{a_1u_1 + a_2u_2 + ... + a_nu_n \quad tel\ que \quad a_1, a_2, ..., a_n \in \mathbb{Z}\}.$$

Définition 26. *Une base d'un réseau est un ensemble de vecteurs indépendants qui engendre ce réseau. La dimension d'un réseau est le nombre de vecteurs d'une base de ce réseau.*

Dans le cadre de ce travail, nous utiliserons essentiellement les réseaux entiers. De manière formelle, on a la définition suivante :

Définition 27. *Un réseau entier est un réseau dont tous les vecteurs sont à coordonnées entières. En d'autres termes, un réseau entier est un sous groupe additif de $\mathbb{Z}^m$,pour $m \geq 1$.*

Exemple 4. $\mathbb{Z}^m = \{(x_1, x_2, ..., x_m) \quad tel\ que \quad x_1, ..., x_m \in \mathbb{Z}\}$ *est un réseau constitué de tous les vecteurs à coordonnées entières.*

1.3.2 Notion de complexité

Dans cette section, nous donnons une signification formelle correspondante aux appréciations telles que «facile» ou «difficile» qui sont souvent attribuées à certains problèmes mathématiques. Nous évaluons la complexité d'un algorithme par le nombre d'opérations élémentaires nécessaire qu'il execute en fonction de la taille de son entrée. La taille de l'entrée, s, est le nombre de symboles nécessaires pour l'écrire dans une base donnée. Ici nous considérons la base de numération binaire. La taille de s est égale à $log_2(s)$. On note $C(s)$ la complexité d'un algorithme qui prend un nombre s en entrée et on distingue les différents types de complexité suivants :

- si $C(s) \leq D\log_2(s)$ pour une constante $D \in \mathbb{R}$, on dit que l'algorithme est à complexité linéaire (en ce sens qu'elle correspond à la taille de l'écriture binaire de n).
- si $C(s) \leq D\log_2(s)^\epsilon$, $D \in \mathbb{R}$ et $\epsilon \in \mathbb{N}$, on dit que l'algorithme est à complexité polynomiale en $log_2(s)$ (linéaire si $\epsilon = 1$, quadratique si $\epsilon = 2$, cubique si $\epsilon = 3$, etc...).
- $C(s) \leq Ds = De^{\log_2(s)}$ pour $D \in \mathbb{R}$, on dit que l'algorithme est à complexité exponentielle.
- Si $C(s) \leq e^{\epsilon \log_2(s)^a \log_2(\log_2(s))^{1-a}}$ pour $\epsilon \in \mathbb{R}$ et $a \in [0,1]$, on dit que l'algorithme est à complexité sous-exponentielle. En particulier, si $a = 0$, il s'agit de la complexité polynomiale alors que si $a = 1$, il s'agit de la complexité exponentielle.

Un problème est dit facile (resp. très facile ou très difficile) si l'algorithme qui le résout est à complexité (resp. linéaire ou exponentielle).

Chapitre 2

Peut-on construire des courbes elliptiques OEF convenables à la cryptographie à base de couplage ?

Sommaire

Il est connu que toutes les courbes elliptiques utilisées dans les cryptosystèmes actuels construits à base de couplage sont définies sur des corps premiers [BLS02, BN05, DEM05, FST10]. Cependant, tous les corps premiers ne possèdent pas toujours une arithmétique qui se prête à une implémentation efficace. Et sachant que toutes les opérations sur les courbes elliptiques sont exécutées en utilisant les opérations arithmétiques dans le corps de base, il est nécessaire d'opérer un choix judicieux d'une classe de corps qui présente des avantages pour une implémentation efficace. Une solution pourrait être d'utiliser les extensions de corps finis optimales car cette famille tire avantage de l'arithmétique rapide se trouvant dans le processeur. En outre, il importe de mentionner le fait qu'il n'y a pas encore eu, jusqu'à présent, de proposition de construction des courbes elliptiques définies sur des extensions de corps finis optimales, adaptées à la cryptographie à base de couplage. Ce qui nous amène à nous interroger sur la possibilité de construire des courbes elliptiques OEF convenables à la cryptographie à base de couplage. Autrement dit, existe-t-il des courbes elliptiques OEF convenables à la cryptographie à base de couplage ?

2.1 Définition des courbes elliptiques OEF

En 1998, Daniel Bailey et Christof Paar ont introduit une classe de corps finis appelée en anglais «Optimal Extension Field, (OEF)». Ces extensions de corps optimales sont définis sur un corps de base de la taille d'un mot machine et tirent avantage de l'arithmétique rapide se trouvant dans le processeur. De manière formelle, on a la définition suivante :

Définition 28. *[BP98]*
Une Extension de corps optimale (OEF) est un corps fini $\mathbb{F}_{p^m}$ tel que :

1. *p est un nombre pseudo-Mersenne premier i.e. un nombre premier de la forme $2^n \pm c$, avec $log_2 c \leq \frac{1}{2}n$.*
2. *Il existe un binôme irréductible $f(x) = x^m - \omega$ sur $\mathbb{F}_p$.*

Définition 29. *On appelera une courbe elliptique OEF, toute courbe elliptique définie sur les extensions de corps optimales.*

2.2 Essai de génération de courbes OEF adaptées au couplage

Pour proposer un algorithme de génération de paramètres nécessaires pour la construction d'une courbe elliptique OEF bien couplée, il est important de se situer tout d'abord dans un contexte plus général, à savoir celui des extensions de corps fini. Ainsi, il faut rappeler les contraintes à prendre en compte lors de la conception d'un algorithme de génération de courbes elliptiques définies sur des extensions de corps adaptées à la cryptographie à base de couplage ainsi que les difficultés à contourner.
C'est pour cette raison que nous commençons par énumérer toutes les attaques connues sur les courbes elliptiques définies sur des extension de corps fini dans un premier temps. Et sachant les corps OEF ne sont que des cas particuliers, il va de soi que les dites attaques s'apppliquent aussi sur elles.
Une courbe elliptique définie sur une extension de corps $\mathbb{F}_q$ avec $q = p^m$, p premier et $m > 1$ un entier est vulnérable à certaines attaques : la descente de Weil [FG98], le calcul d'indice basé sur la décomposition [Gau09], le problème de Diffie-Hellman statique [Gra10] en sont des exemples.
Dans les paragraphes suivants, il est donné une description succincte de chacune de ces attaques.

L'attaque de la descente de Weil proposée par Frey et *al.* [FG98] consiste à transférer le problème du logarithme discret de $E(\mathbb{F}_{q^m})$ où q est un nombre premier ou une puissance d'un nombre premier, à la jacobienne d'une courbe $\mathcal{C}$ de dimension m sur $\mathbb{F}_q$. Elle calcule alors le logarithme discret sur cette jacobienne en utilisant un algorithme de calcul d'indice.

L'attaque reposant sur l'algorithme de calcul d'indice basé sur la décomposition développé par Gaudry [Gau09] a une complexité de $O(q^{2-2/m})$ pour $m \geq 3$ et s'applique sur toutes les courbes (hyper-)elliptiques définies sur des extensions de corps de petit degré. Cette attaque combine les idées issues de la définition du calcul d'indice proposées par Semeav [Sem04] ainsi que l'attaque de la descente de Weil.

Diem [Die11] a prouvé que lorsque p^m croît de telle sorte que m^2 soit de l'ordre de $\log_2 p$, l'algorithme de Gaudry a une complexité sous exponentielle. Peu après, Joux et Vitse [JV10] ont amélioré ce calcul d'indice lorsque $m \geq 5$ et $\log_2 p \leq O(m^3)$. Il est important de relever que les attaques de la descente de Weil et du calcul d'indice basées sur la décomposition sont un peu plus efficaces que les attaques génériques, et inefficaces pour la résolution du problème du logarithme discret sur les courbes elliptiques en pratique [ZWL13].

Granger [Gra10] a trouvé le meilleur algorithme connu qui résout le problème de Diffie-Hellman statique sur les courbes elliptiques définies sur un corps fini $\mathbb{F}_{q^m}$ où m est un entier composé, en temps heuristique de l'ordre de $O(q^{1-\frac{1}{m+1}})$. Estibals [Est10] a montré qu'une protection contre cette attaque consiste à révoquer une clé après un certain nombre d'utilisation.
Une revue exhaustive de ces attaques est proposée dans [ZWL13] ; Zhang et al. y suggèrent de choisir un grand nombre premier p et un entier $m \geq 5$ afin d'empêcher ces attaques connues en pratique.

Supposons que E soit une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ où $q = p^m$, avec p un nombre premier et $m \geq 1$ un entier. Soit $r \geq 5$ un facteur premier de $\#E(\mathbb{F}_q)$ et k le plus petit entier tel que $r|q^k - 1$, appelé degré de plongement. Après avoir pris en considération toutes les contraintes et la recommandation sus évoquées, supposons maintenant que E soit une courbe elliptique définie sur une extension de corps $\mathbb{F}_q$ avec $q = p^m$, p étant un grand nombre premier et m un entier supérieur ou égal à 5.
Afin de garantir un calcul efficace du couplage sur E, il est nécessaire que les conditions suivantes puissent être satisfaites :

- q est une puissance d'un nombre premier.
- il existe un nombre premier $r \geq \sqrt{q}$ divisant $\#E(\mathbb{F}_q)$ et ne divisant aucun des $\#E(\mathbb{F}_{p^i})$, pour $1 \leq i < m$.
- t est relativement premier à p, où t est la trace de Frobenius.
- r divise $q + 1 - t$.
- r divise $\phi_k(t-1)$, où k est le degré de plongement de E et ϕ_k le $k^{\text{ième}}$ polynôme cyclotomique.
- $4q - t^2 = Dy^2$ pour un entier positif suffisamment petit D et pour un entier y où D est le discriminant de la multiplication complexe. De plus, si $D < 10^{13}$ alors E peut être construite à l'aide de la méthode de la multiplication complexe.

Maintenant si nous nous restreignons au problème de la génération des

paramètres qui entrent dans la construction des courbes elliptiques OEF ordinaires bien couplées . Dans le but de générer ces paramètres, nous avons passé en revue toutes les méthodes de construction des courbes elliptiques bien couplées de la littérature dont [FST10] en fait une taxonomie de ces courbes. Il en ressort que la majorité de ces techniques de construction repose sur la méthode de multiplication complexe pour construire ces courbes, le degré de plongement k est connue à l'avance et la caractéristique du corps sur lequel sera définie la courbe elliptique bien couplée désirée n'est pas connue à l'avance, i.e. elle se construit et donc ne peut être donnée comme une entrée de l'algorithme de génération des paramètres de la dite courbe. Ces algorithmes se terminent par un test de primalité sur cette valeur construite. Il s'en dégage l'idée selon laquelle cette caractéristique ne pourrait pas avoir une forme préfinie et très particulière contrairement au cas des courbes elliptiques OEF. D'où ces méthodes de construction ne sont pas appropriées pour générer des courbes elliptiques OEF bien couplées.

De ce constat, il s'ensuit que pour un éventuel algorithme de génération des paramètres d'une courbe elliptique OEF ordinaire bien couplée, il faut nécessairement :

1. un degré de plongement k connue à l'avance car il est peu probable voire impossible d'obtenir une valeur de k qui soit plus petit que $\log_2 q$ s'il faut le déterminer connaissant r et q d'après [BK98].
2. imposer p à être pseudo Mersenne premier. Ce qui implique que la caractéristique doit, soit être connue à l'avance, soit construite de telle sorte qu'elle soit pseudo Mersenne premier dès les premières étapes de l'algorithme de génération.

Nous proposons l'approche suivante qui consiste à prendre comme paramètres d'entrée k le degré de plongement, D le discriminant de la multiplication complexe, p la caractéristique du corps désiré, n le niveau de sécurité qu'on souhaite atteindre. Les paramètres à rechercher sont : m le degré de l'extension de corps, t la trace de Frobenius de la courbe E désirée, q la taille du corps OEF et r un grand diviseur premier de $\#E(\mathbb{F}_q)$.
Sachant que pour obtenir une courbe elliptique d'un niveau de sécurité équivalent à AES n-bits, q doit être de la taille $2n$. Ainsi, en connaissant le niveau de sécurité souhaité et la taille de p, on peut en déduire la valeur m du degré de l'extension de corps. Nous en déduisons de la relation $q = p^m$ que $m = \dfrac{2n}{\log_2 p}$.

Connaissant p, m, D et à partir de la condition $4p^m - t^2 = Dy^2$, on peut déjà trouver la valeur de t la trace de Frobenius en utilisant une extension de l'algorithme proposé par Corniacchia [Coh95, p.34-36] que nous baptisons «Corniacchiapp». L'algorithme de Cornacchia permet de résoudre les équations du type $x^2 + |c|y^2 = 4p$ avec p un nombre premier impair et c un entier

tel que $c \equiv 0$ ou $1 \ (mod\ 4)$ et $|c| < 4p$. L'algorithme Corniacchiapp permet de résoudre des équations du type $x^2 + |c|y^2 = 4p^m$ et $c \equiv 0$ ou $1 \ (mod\ 4)$ et $|c| < 4p^m$. Il prend en entrée les paramètres p, m et D et donne en sortie la valeur t, du couple de solution (t, y), s'il existe, de l'équation $t^2 + Dy^2 = 4p^m$. Ces étapes sont illustrées dans l'algorithme 2 se trouvant à la fin de ce chapitre. Nous l'avons implémenté sur Sagemath.
Ensuite le calcul de $\phi_k(t-1)$ peut être effectué. Si $\phi_k(t-1)$ est un nombre premier, alors on pose $r = \phi_k(t-1)$ sinon on attribue à r la valeur du plus grand facteur premier de $\phi_k(t-1)$. Mais il est fort probable que la valeur de $\phi_k(t-1)$ soit très grande (de la taille cryptographique), mais l'obtention d'une valeur du paramètre r est étroitement liée à la résolution de l'épineux problème de la factorisation qui reste encore à l'heure actuelle un problème difficile. De plus qu'est ce qui nous garantit que r va diviser $q+1-t$? Cette approche ne nous permet pas donc de conclure.

2.3 Conclusion

Dans ce chapitre, nous avons proposé une approche de construction de courbes elliptiques bien couplées définies sur les extensions de corps optimales bien que nous rencontrons jusqu'à date de sérieux problèmes d'incompatibilité des relations impliquant les paramètres recherchés.

Algorithme 2 : Cornacchiapp

Entrée : p, m, D
Sortie : t, y
$q \leftarrow p^m$
si *(*$D \equiv 0 (mod\ \ 4)$ *et* $|D| < 4q$*) ou (*$D \equiv 1 (mod\ \ 4)$ *et* $|D| < 4q$**) alors**
    $h \leftarrow \left(\frac{D}{p}\right)$
    si $h == -1$ **alors**
        renvoie "l'équation n'a pas de solution."
    fin
    trouver un entier v tel que $v^2 \equiv D \ \ (mod\ \ q)$ et $0 \leq v < q$.
    si $v \not\equiv D \ \ (mod\ \ 2)$ **alors**
        $v \leftarrow q - v$
    fin
    $a \leftarrow 2q$
    $b \leftarrow v$
    $l \leftarrow \lfloor 2\sqrt{q} \rfloor$
    si $b > l$ **alors**
        $s \leftarrow a (mod\ \ b)$
        $a \leftarrow b$
        $b \leftarrow s$ et aller à l'étape 16
    fin
    $c \leftarrow (4q - b^2)/|D|$
    si $|D| \nmid (4q - b^2)$ *ou c n'est pas le carré d'un entier* **alors**
        renvoie "l'équation n'a pas de solution."
    fin
    sinon
        $t \leftarrow b$
        renvoie t
    fin
fin
sinon
    renvoie "Une telle solution n'existe pas."
fin

Chapitre 3

Couplage optimal Ate sur les courbes elliptiques de degrés de plongement 9, 15 et 27 respectivement

Sommaire

3.1 Présentation

Ce chapitre est entièrement consacré aux calculs du couplage optimal Ate sur les courbes elliptiques de degré de plongement impair, notamment pour $k = 9, 15$ et 27 au niveau de sécurité de $128, 192$ et 256-bits respectivement, d'après la table 3.1 [FST10].

Niveau de sécurité	Taille en bits de r	Taille en bits de q^k	k $\rho \approx 1$	k $\rho \approx 2$
80	160	960 – 1280	6 – 8	3 – 4
128	256	3000 – 5000	12 – 20	6 – 10
192	384	8000 – 10000	20 – 26	10 – 13
256	512	14000 – 18000	28 – 36	14 – 18

TABLE 3.1 – Taille en bit des paramètres de courbes et leurs degrés de plongement correspondant pour obtenir un niveau de sécurité désiré.

Il est important de mentionner que les résultats présentés dans ce chapitre ont été obtenus au cours de la période dans laquelle quelques avancées dans le calcul du logarithme discret sur les corps finis ont été observées [JP13, BGK15]. Mais les algorithmes proposés à cet effet étaient considérés non-pratiques et les recommandations de tailles n'avaient pas été changées. C'est pour cette raison que nous nous sommes appuyés sur les recommandations de tailles de paramètres de la table 3.1 proposées par Freeman et *al.* [FST10] pour mener notre étude.

Dans ce chapitre, des améliorations considérables sont apportées au calcul du couplage optimal Ate sur les courbes sus-citées. Ces courbes admettent des tordues de degrée 3 ; ce qui permet de faire des calculs dans des sous-corps et d'utiliser la technique d'élimination des dénominateurs [BLS04a, Theorem 2].

Une arithmétique détaillée dans les tours d'extension de corps associés à $\mathbb{F}_{p^9}, \mathbb{F}_{p^{15}}$ et $\mathbb{F}_{p^{27}}$ est proposée dans ce chapitre. La méthode proposée par Fuentes et al.[FKR11] basée sur les réseaux est utilisée pour calculer l'exponentiation finale dans les cas $k = 9, 15$ afin d'obtenir une expression simplifiée. L'évaluation du coût du couplage optimal Ate dans les cas $k = 9$ et $k = 15$ est explicitée comparativement aux travaux de [MGI09]. Les résultats obtenus sont des améliorations par rapport aux précédents travaux [LT12], [MGI09] et [ZL12] respectivement pour les cas $k = 9, 15$ et 27. Précisement, notre contribution (voir table 3.3 pour la comparaison) dans ce travail se résume comme suit :

1. La détermination d'un coût explicite du calcul du couplage optimal Ate pour les courbes elliptiques sus-citées. Ceci inclut une bonne sélection de paramètres pour une boucle de Miller courte et une expo-

nentiation finale efficace. En particulier, nous supprimons une inversion dans le corps $\mathbb{F}_{p^{27}}$ pour le calcul de la boucle de Miller dans le cas $k = 27$.

2. Le détail de l'arithmétique dans la tour d'extension de corps de $\mathbb{F}_{p^9}, \mathbb{F}_{p^{15}}$ et $\mathbb{F}_{p^{27}}$. Principalement, nous donnons les coûts, comme expliqué dans la section 3.2, du calcul de l'endomorphisme de Frobenius et de l'inversion dans les sous-groupes cyclotomiques de $\mathbb{F}^\star_{p^9}, \mathbb{F}^\star_{p^{15}}$ et $\mathbb{F}^\star_{p^{27}}$.
3. L'amélioration des coûts de l'exponentiation finale en supprimant $24M_9 + 5S_9, 26M_{15} + 173S_{15}$ et $20M_{27}$ opérations pour les courbes elliptiques de degré de plongement $9, 15$ et 27 respectivement, comparativement aux résultats des travaux existants.

Pour ce qui est du reste de ce chapitre, la section 3.2 donne le détail de l'arithmétique dans la tour d'extension de corps de $\mathbb{F}_{p^9}, \mathbb{F}_{p^{15}}$ et $\mathbb{F}_{p^{27}}$ ainsi que les coûts de ces opérations arithmétiques. Les sections 3.3, 3.4 et 3.5 présentent l'arithmétique dans les sous corps, une estimation du coût de la boucle de Miller ainsi que de l'exponentiation finale pour $k = 9, 15$ et 27 respectivement. Une analyse comparative avec les travaux précédents est faite dans chacune de ces sections. La section 3.6 donne une conclusion à ce chapitre dans laquelle une comparaison générale des résultats obtenus dans ce travail et les précédents résultats de la littérature est faite.

3.2 Arithmétique dans $\mathbb{F}_{p^9}$, $\mathbb{F}_{p^{15}}$ et $\mathbb{F}_{p^{27}}$

Habituellement, le résultat d'un couplage s'exprime comme un élément de l'extension de corps $\mathbb{F}_{p^k}$. Ainsi l'efficacité du calcul d'un couplage dépend fortement de l'arithmétique dans les sous corps de $\mathbb{F}_{p^k}$. Ce dernier peut être construit sous forme de tour d'extension de corps. Dans cette section, la tour d'extension de corps finis $\mathbb{F}_{p^9}, \mathbb{F}_{p^{27}}$ et $\mathbb{F}_{p^{15}}$ respectivement est rappelée et les coûts explicites des opérations arithmétiques dans ces corps sont également donnés .

Concernant l'arithmétique dans les corps finis $\mathbb{F}_{p^9}$ et $\mathbb{F}_{p^{27}}$, nous supposons $p \equiv 1 \bmod 3$ en nous appuyant sur les travaux de Barreto et *al.* [BLS02] sur la construction de courbes elliptiques bien couplées de degré de plongement 9 et 27. Ce qui implique que $\mathbb{F}_{p^{27}}$ peut être représenté comme $\mathbb{F}_p[X]/(X^k - \alpha)$, pour $k = 3^i, i = 2, 3$ où α n'est pas un résidu cubique modulo p ([LN97], Theorem 3.75). Dans les conditions où $p \equiv 1 \bmod 3$, on peut prendre $\alpha = 2$. Il en découle que $X^3 - 2$ est irréductible sur $\mathbb{F}_p$.

Ainsi une extension cubique sera construite en utilisant les polynômes $X^3 - \alpha_i$ où $\alpha_i = 2^{1/3^i}, i = 0, 1, 2$ [BS04]. Des tours d'extension pour les corps $\mathbb{F}_{p^{27}}$ et $\mathbb{F}_{p^9}$ sont alors données par :

$$\mathbb{F}_{p^3} = \mathbb{F}_p[u] \text{ avec } u^3 = 2$$

$$\mathbb{F}_{p^9} = \mathbb{F}_{p^3}[v] \text{ avec } v^3 = 2^{1/3}$$
$$\mathbb{F}_{p^{27}} = \mathbb{F}_{p^9}[w] \text{ avec } w^3 = 2^{1/9}$$

Les coûts du calcul des puissances de l'endomorphisme de Frobenius et d'inversions cyclotomiques sont donnés dans le Lemme 2 pour le corps fini $\mathbb{F}_{p^9}$.

Lemme 2. *Dans le corps fini* $\mathbb{F}_{p^9}$,
- *L'inverse d'un élément* α *du sous groupe cyclotomique* $G_{\phi_3(p^3)}$ *est calculé comme* $\alpha^{-1} = \alpha^{p^3}.\alpha^{p^6}$ *et son coût est de* $\mathbf{36S_1 + 84A_1}$.
- *Le calcul de chacune des puissances* p^3 ; p^6 *de l'endomorphisme de Frobenius coûte* $\mathbf{6M_1 + 6A_1}$.
- *Le calcul de chacune des puissances* p ; p^2 ; p^4 ; p^5, p^7, p^8 *de l'endomorphisme de Frobenius coûte* $\mathbf{8M_1 + 6A_1}$.

Preuve

Inversion cyclotomique : Soit $a = a_0 + a_1 v + a_2 v^2 \in \mathbb{F}_{p^9}$ avec $a_i \in \mathbb{F}_{p^3}$. Supposons que a appartient au sous groupe cyclotomique $G_{\phi_3(p^3)}$, i.e.

$$a^{p^6+p^3+1} = 1;$$

alors

$$a^{-1} = a^{p^6} a^{p^3}.$$

Pour calculer $a^{p^6}a^{p^3}$, nous avons besoin des valeurs de v^{p^3} et v^{p^6}. Puisque $v^3 = 2^{1/3}$, nous avons :

$$v^{p^3} = v^{3(p^3-1)/3+1} = v^{3(p^3-1)/3} v = (v^3)^{(p^3-1)/3} v = (2^{1/3})^{(p^3-1)/3} v.$$

Soit $\mu = (2^{1/3})^{(p^3-1)/3}$; on a $\mu \neq 1$ et $\mu^3 = 1$ ainsi μ est une racine cubique primitive de l'unité dans $\mathbb{F}_{p^3}$. Nous obtenons

$$v^{p^3} = \mu v \text{ et } v^{p^6} = (v^{p^3})^{p^3} = (\mu v)^{p^3} = \mu (v)^{p^3} = \mu\mu v = \mu^2 v.$$

Nous avons alors

$$a^{p^3} = a_0^{p^3} + a_1^{p^3} v^{p^3} + a_2^{p^3} (v^2)^{p^3} = a_0 + a_1 v^{p^3} + a_2 (v^2)^{p^3} = a_0 + a_1 \mu v + a_2 \mu^2 v^2$$

$$\text{et } a^{p^6} = (a^{p^3})^{p^3} = a_0 + a_1 (\mu v)^{p^3} + a_2 (\mu^2 v^2)^{p^3} = a_0 + a_1 \mu^2 v + a_2 \mu^4 v^2.$$

Ainsi lorsqu'on utilise $v^3 = 2^{1/3}$ et $\phi_3(\mu) = \mu^2 + \mu + 1 = 0$, on a finalement :

$$a^{p^6} a^{p^3} = (a_0^2 - a_1 a_2 2^{1/3}) + (a_2^2 2^{1/3} - a_0 a_1) v + (a_1^2 - a_0 a_2) v^2.$$

Donc le calcul de l'inverse d'un élément $a \in \mathbb{F}_{p^9}$ satisfaisant $a^{p^6+p^3+1} = 1$ nécessite 3 carrés dans $\mathbb{F}_{p^3}$, 3 multiplications dans $\mathbb{F}_{p^3}$ et 3 additions dans $\mathbb{F}_{p^3}$.

D'où ce calcul coûte
$3(3S_1 + 3M_1 + 8A_1) + 3(6M_1 + 17A_1) + 3(3A_1) = 36M_1 + 84A_1$ en supposant que le coût d'une multiplication est le même pour un carré.

Opérateurs Frobenius : La puissance p^i de l'endomorphisme de Frobenius est l'application $\pi^i : \mathbb{F}_{p^9} \to \mathbb{F}_{p^9}, a \mapsto a^{p^i}$.

Soit $a \in \mathbb{F}_{p^9}$, $a = a_0 + a_1 v + a_2 v^2$ avec $a_i \in \mathbb{F}_{p^3}$ alors

$$\pi(a) = a^p = a_0^p + a_1^p v^p + a_2^p (v^2)^p = a_0^p + a_1^p v^p + a_2^p (v^p)^2.$$

$a_0 \in \mathbb{F}_{p^3}$ peut être écrite comme suit $a_0 = g_0 + g_1 u + g_2 u^2$, avec $g_i \in \mathbb{F}_p$, $i \in \{0, 1, 2\}$. Ainsi,

$$a_0^p = g_0 + g_1 u^p + g_2 (u^2)^p.$$

Par ailleurs, nous avons :

$$u^p = u^{3(p-1)/3+1} = (u^3)^{(p-1)/3} u = 2^{(p-1)/3} u$$

et puisque 2 n'est pas un cube dans $\mathbb{F}_p$,

$$2^{(p-1)/3} \neq 1.$$

Soit $\alpha = 2^{(p-1)/3}$; alors $\alpha \neq 1$ et $\alpha^3 = 1$; i.e. α est une racine cubique primitive de l'unité dans $\mathbb{F}_p$ et $u^p = \alpha u$. Donc

$$a_0^p = g_0 + g_1 u^p + g_2 (u^2)^p = g_0 + g_1 \alpha u + g_2 \alpha^2 u^2.$$

Et de manière analogue, nous avons :

$$a_1^p = g_3 + g_4 u^p + g_5 (u^2)^p = g_3 + g_4 \alpha u + g_5 \alpha^2 u^2$$

$$a_2^p = g_6 + g_7 u^p + g_8 (u^2)^p = g_6 + g_7 \alpha u + g_8 \alpha^2 u^2.$$

Maintenant pour le calcul de v^p, on observe que :

$$v^p = v^{3(p-1)/3+1} = (v^3)^{(p-1)/3} v = (2^{1/3})^{(p-1)/3} v = 2^{(p-1)/9} v$$

ainsi si $\beta = 2^{(p-1)/9}$, nous aurons

$$\beta \neq 1, \quad \beta^3 = 2^{(p-1)/3} = \alpha \neq 1 \quad \text{et} \quad \beta^9 = 1.$$

D'où β est une racine neuvième primitive de l'unité dans $\mathbb{F}_p$ et $v^p = \beta v$. Finalement,

$$a^p = g_0 + g_1 \alpha u + g_2 \alpha^2 u^2 + (g_3 \beta + g_4 \alpha \beta u + g_5 \alpha^2 \beta u^2) v + (g_6 \beta^2 + g_7 \alpha \beta^2 u + g_8 \alpha^2 \beta^2 u^2) v^2.$$

Les relations algébriques :

$$\alpha = \beta^3, \quad \alpha\beta = \beta^4, \quad \alpha\beta^2 = \beta^5, \quad \alpha^2\beta = \beta^7, \quad \alpha^2\beta^2 = \beta^8.$$

conduisent à

$$a^p = (g_0 + g_1\beta^3 u + g_2\beta^6 u^2) + (g_3\beta + g_4\beta^4 u + g_5\beta^7 u^2)v + (g_6\beta^2 + g_7\beta^5 u + g_8\beta^8 u^2)v^2$$

Le coût de la puissance p de l'endomorphisme de Frobenius est : $8M_1 + 6A_1$.

Pour le calcul de chacune des puissances p^2, p^4, p^5, p^7 et p^8 de l'endomorphisme de Frobenius, la démarche est similaire à celle du calcul de la puissance p de l'endomorphisme. Ainsi, nous obtenons que les coût des puissances p^2, p^4, p^5, p^7 et p^8 de l'endomorphisme de Frobenius ont la même valeur que le coût de la puissance p de l'endomorphisme de Frobenius.

Pour la puissance p^3 de l'endomorphisme de Frobenius, nous constatons du calcul de l'inversion cyclotomique que $v^{p^3} = \mu v$. Alors

$$a^{p^3} = a_0 + a_1\mu v + a_2\mu^2 v^2 = (g_0 + g_1 u + g_2 u^2) + (g_3 + g_4 u + g_5 u^2)\mu v + (g_6 + g_7 u + g_8 u^2)\mu^2 v^2.$$

$t = \mu^2$ étant précalculé lors de l'inversion cyclotomique ; nous avons finalement

$$a^{p^3} = (g_0 + g_1 u + g_2 u^2) + (g_3\mu + g_4\mu u + g_5\mu u^2)v + (g_6 t + g_7 t u + g_8 t u^2)v^2.$$

Le coût de la puissance p^3 de l'endomorphisme de Frobenius est : $6M_1 + 6A_1$. C'est également la même valeur que celui du coût de la puissance p^6 de l'endomorphisme de Frobenius.

□

Le Lemme 3 donne les coûts du calcul de l'endomorphisme de Frobenius et des inversions cyclotomiques dans le corps fini $\mathbb{F}_{p^{27}}$.

Lemme 3. *Dans le corps fini $\mathbb{F}_{p^{27}}$,*

- *L'inverse d'un élément α du sous groupe cyclotomique $G_{\phi_3(p^9)}$ est calculé comme $\alpha^{-1} = \alpha^{p^9}.\alpha^{p^{18}}$ et son coût est de $\mathbf{216S_1 + 759A_1}$.*
- *Le calcul de chacune des puissances p^3, p^6, p^9 de l'endomorphisme de Frobenius coûte $\mathbf{18M_1 + 18A_1}$.*
- *Le calcul de chacune des puissances p, p^2, p^4, p^5, p^7, p^8 de l'endomorphisme de Frobenius coûte $\mathbf{26M_1 + 18A_1}$.*

Preuve

Inversion cyclotomique : nous suivons une démarche similaire à celle de la preuve du lemme 3.2. L'élément $a = a_0 + a_1 w + a_2 w^2 \in \mathbb{F}_{p^{27}}$ avec $a_i \in \mathbb{F}_{p^9}$ étant dans le sous groupe cyclotomique $G_{\phi_3(\mathbb{F}_{p^9})}$, alors il satisfait la relation $a^{p^{18}+p^9+1} = 1$. D'où

$$a^{-1} = a^{p^{18}} a^{p^9}.$$

Pour calculer $a^{p^{18}} a^{p^9}$, il nous faut les valeurs de w^{p^9} et $w^{p^{18}}$. Puisque $w^3 = 2^{1/9}$, nous avons alors

$$w^{p^9} = w^{3(p^9-1)/3+1} = w^{3(p^9-1)/3} w = (w^3)^{(p^9-1)/3} w = (2^{1/9})^{(p^9-1)/3} w.$$

Soit $\sigma = (2^{1/9})^{(p^9-1)/3}$; alors $\sigma \neq 1$ et $\sigma^3 = 1$. Donc σ est une racine cubique primitive de l'unité dans $\mathbb{F}_{p^9}$ i.e. $\phi_3(\sigma) = 0$. Nous obtenons

$$w^{p^9} = \sigma w$$

et nous pouvons à présent calculer $w^{p^{18}}$ comme

$$w^{p^{18}} = (w^{p^9})^{p^9} = (\sigma w)^{p^9} = \sigma(w)^{p^9} = \sigma\sigma w = \sigma^2 w.$$

Nous avons alors

$$a^{p^9} = a_0 + a_1 w^{p^9} + a_2 (w^2)^{p^9} = a_0 + a_1\sigma w + a_2\sigma^2 w^2$$

et

$$a^{p^{18}} = (a^{p^9})^{p^9} = a_0 + a_1(\sigma w)^{p^9} + a_2(\sigma^2 w^2)^{p^9} = a_0 + a_1\sigma^2 w + a_2\sigma^4 w^2.$$

Après développement et réduction en utilisant $w^3 = 2^{1/9}$ et $\phi_3(\sigma) = \sigma^2 + \sigma + 1 = 0$, on obtient

$$a^{p^{18}} a^{p^9} = (a_0^2 - a_1 a_2 2^{1/9}) + (a_2^2 2^{1/9} - a_0 a_1)w + (a_1^2 - a_0 a_2)w^2$$

Donc le calcul de l'inverse d'un élément $a \in \mathbb{F}_{p^{27}}$ satisfaisant $a^{p^{18}+p^9+1} = 1$ nécessite 3 carrés dans $\mathbb{F}_{p^9}$, 3 multiplications dans $\mathbb{F}_{p^9}$, 3 additions dans $\mathbb{F}_{p^9}$. Ce calcul coûte $3(36M_1+95A_1)+3(36M_1+149A_1)+3(9A_1) = 216M_1+759A_1$.

Opérateurs Frobenius : La puissance p^i de l'endomorphisme de Frobenius est l'appplication $\pi^i : \mathbb{F}_{p^{27}} \to \mathbb{F}_{p^{27}}, a \mapsto a^{p^i}$.
Soit $a = a_0 + a_1 w + a_2 w^2$ avec $a_i \in \mathbb{F}_{p^9}$ un élément de $\mathbb{F}_{p^{27}}$.

$$\pi(a) = a^p = (a_0 + a_1 w + a_2 w^2)^p = a_0^p + a_1^p w^p + a_2^p (w^2)^p.$$

L' élément $a_0 \in \mathbb{F}_{p^9}$ peut être écrit comme

$$a_0 = (h_0+h_1u+h_2u^2)+(h_3+h_4u+h_5u^2)v+(h_6+h_7u+h_8u^2)v^2, \;\; h_i \in \mathbb{F}_p, \;\; 0 \leq i \leq 8.$$

On a

$$a_0^p = (h_0 + h_1 u + h_2 u^2 + (h_3 + h_4 u + h_5 u^2)v + (h_6 + h_7 u + h_8 u^2)v^2)^p$$

$$u^p = u^{3(p-1)/3+1} = (u^3)^{(p-1)/3} u = 2^{(p-1)/3} u.$$

Puisque 2 n'est pas un cube dans $\mathbb{F}_p$, nous avons

$$\alpha = 2^{(p-1)/3}, \;\; \alpha \neq 1, \;\; \text{et} \;\; \alpha^3 = 1.$$

C'est-à-dire que α est une racine cubique primitive de l'unité dans $\mathbb{F}_p$ et $u^p = \alpha u$. Ainsi,

$$v^p = v^{3(p-1)/3+1} = (v^3)^{(p-1)/3} v = (2^{1/3})^{(p-1)/3} v = 2^{(p-1)/9} v.$$

Nous avons $\beta = 2^{(p-1)/9} \neq 1$ et $\beta^9 = 1$. D'où β est une racine neuvième primitive de l'unité dans $\mathbb{F}_p$ et $v^p = \beta v$. Ainsi

$$w^p = w^{3(p-1)/3+1} = \left(w^3\right)^{(p-1)/3} v = \left(2^{1/9}\right)^{(p-1)/3} v = 2^{(p-1)/27} v.$$

Nous remarquons également que

$$\gamma = 2^{(p-1)/27} \neq 1, \quad \gamma^3 = 2^{(p-1)/9} = \beta \neq 1, \quad \gamma^9 = 2^{(p-1)/3} = \alpha \neq 1, \quad \text{et} \quad \gamma^{27} = 1.$$

D'où γ est une racine vingt-septième primitive de l'unité dans $\mathbb{F}_p$ et $w^p = \gamma w$.

$$\begin{aligned} a_0^p = & \quad \left((h_0 + h_1 u + h_2 u^2) + (h_3 + h_4 u + h_5 u^2)v + (h_6 + h_7 u + h_8 u^2)v^2\right)^p \\ = & \quad (h_0 + h_1 u^p + h_2 (u^2)^p) + (h_3 + h_4 u^p + h_5 (u^2)^p) v^p \\ & \quad + (h_6 + h_7 u^p + h_8 (u^2)^p)(v^2)^p \\ = & \quad (h_0 + h_1 \alpha u + h_2 \alpha^2 u^2) + (h_3 + h_4 \alpha u + h_5 \alpha^2 u^2)\beta v \\ & \quad + (h_6 + h_7 \alpha u + h_8 \alpha^2 u^2)\beta^2 v^2 \end{aligned}$$

c'est-à-dire que

$$a_0^p = (h_0 + h_1 \alpha u + h_2 \alpha^2 u^2) + (h_3 \beta + h_4 \alpha \beta u + h_5 \alpha^2 \beta u^2) v + (h_6 \beta^2 + h_7 \alpha \beta^2 u + h_8 \alpha^2 \beta^2 u^2) v^2.$$

$$\begin{aligned} a_1^p = & \quad (h_9 + h_{10} u + h_{11} u^2) + (h_{12} + h_{13} u + h_{14} u^2)v + (h_{15} + h_{16} u + h_{17} u^2)v^2)^p \\ = & \quad (h_9 + h_{10} u^p + h_{11} (u^2)^p) + (h_{12} + h_{13} u^p + h_{14} (u^2)^p) v^p \\ & \quad + (h_{15} + h_{16} u^p + h_{17} (u^2)^p)(v^2)^p \\ = & \quad (h_9 + h_{10} \alpha u + h_{11} \alpha^2 u^2) + (h_{12} + h_{13} \alpha u + h_{14} \alpha^2 u^2)\beta v \\ & \quad + (h_{15} + h_{16} \alpha u + h_{17} \alpha^2 u^2)\beta^2 v^2. \end{aligned}$$

Alors

$$a_1^p = (h_9 + h_{10} \alpha u + h_{11} \alpha^2 u^2) + (h_{12} \beta + h_{13} \alpha \beta u + h_{14} \alpha^2 \beta u^2) v + (h_{15} \beta^2 + h_{16} \alpha \beta^2 u + h_{17} \alpha^2 \beta^2 u^2) v^2.$$

$$\begin{aligned} a_2^p = \quad & (h_{18}+h_{19}u+h_{20}u^2)+(h_{21}+h_{22}u+h_{23}u^2)v+(h_{24}+h_{25}u+h_{26}u^2)v^2)^p \\ = \quad & (h_{18}+h_{19}u^p+h_{20}(u^2)^p)+(h_{21}+h_{22}u^p+h_{23}(u^2)^p)v^p \\ & +(h_{24}+h_{25}u^p+h_{26}(u^2)^p)(v^2)^p \\ = \quad & (h_{18}+h_{19}\alpha u+h_{20}\alpha^2u^2)+(h_{21}+h_{22}\alpha u+h_{23}\alpha^2u^2)\beta v \\ & +(h_{24}+h_{25}\alpha u+h_{26}\alpha^2u^2)\beta^2v^2 \end{aligned}$$

Ainsi

$$\begin{aligned} a_2^p = \quad & (h_{18}+h_{19}\alpha u+h_{20}\alpha^2u^2)+(h_{21}\beta+h_{22}\alpha\beta u+h_{23}\alpha^2\beta u^2)v \\ & +(h_{24}\beta^2+h_{25}\alpha\beta^2u+h_{26}\alpha^2\beta^2u^2)v^2. \end{aligned}$$

Nous avons

$$\pi(a)=(a_0+a_1w+a_2w^2)^p=a_0^p+a_1^pw^p+a_2^p(w^2)^p=a_0^p+a_1^p\gamma w+a_2^p\gamma^2w^2.$$

Après remplacement et développement, nous aurons :

$$\begin{aligned} \pi(a) = \quad & (h_0+h_1\alpha u+h_2\alpha^2u^2)+(h_3\beta+h_4\alpha\beta u+h_5\alpha^2\beta u^2)v \\ + \quad & (h_6\beta^2+h_7\alpha\beta^2u+h_8\alpha^2\beta^2u^2)v^2+[(h_9+h_{10}\alpha u+h_{11}\alpha^2u^2) \\ + \quad & (h_{12}\beta+h_{13}\alpha\beta u+h_{14}\alpha^2\beta u^2)v+(h_{15}\beta^2+h_{16}\alpha\beta^2u+h_{17}\alpha^2\beta^2u^2)v^2]\gamma w \\ + \quad & [(h_{18}+h_{19}\alpha u+h_{20}\alpha^2u^2)+(h_{21}\beta+h_{22}\alpha\beta u+h_{23}\alpha^2\beta u^2)v \\ + \quad & (h_{24}\beta^2+h_{25}\alpha\beta^2u+h_{26}\alpha^2\beta^2u^2)v^2]\gamma^2w^2. \end{aligned}$$

Nous avons les relations algébriques suivantes :

$$\alpha=\beta^3,\ \ \alpha\beta=\beta^4,\ \ \alpha\beta^2=\beta^5,\ \ \alpha^2\beta=\beta^7\ \text{ et }\ \alpha^2\beta^2=\beta^8.$$

Donc

$$\begin{aligned} \pi(a) = \quad & (h_0+h_1\beta^3u+h_2\beta^6u^2)+(h_3\beta+h_4\beta^4u+h_5\beta^7u^2)v \\ + \quad & (h_6\beta^2+h_7\beta^5u+h_8\beta^8u^2)v^2+[(h_9\gamma+h_{10}\beta^3\gamma u+h_{11}\beta^6\gamma u^2) \\ + \quad & (h_{12}\beta\gamma+h_{13}\beta^4\gamma u+h_{14}\beta^7\gamma u^2)v+(h_{15}\beta^2\gamma+h_{16}\beta^5\gamma u+h_{17}\beta^8\gamma u^2)v^2]w \\ + \quad & [(h_{18}\gamma^2+h_{19}\beta^3\gamma^2u+h_{20}\beta^6\gamma^2u^2)+(h_{21}\beta\gamma^2+h_{22}\beta^4\gamma^2u+h_{23}\beta^7\gamma^2u^2)v \\ + \quad & (h_{24}\beta^2\gamma^2+h_{25}\beta^5\gamma^2u+h_{26}\beta^8\gamma^2u^2)v^2]w^2. \end{aligned}$$

Les valeurs suivantes sont précalculées :

$$\begin{array}{cccccc}
\lambda_0 = \beta^2 & \lambda_1 = \beta^3 & \lambda_2 = \beta^4 & \lambda_3 = \beta^5 & \lambda_4 = \beta^6 & \lambda_5 = \beta^7 \\
\lambda_6 = \beta_8 & \lambda_7 = \gamma^2 & \lambda_8 = \beta\gamma & \lambda_9 = \lambda_0\gamma & \lambda_{10} = \lambda_1\gamma & \lambda_{11} = \lambda_2\gamma \\
\lambda_{12} = \lambda_3\gamma & \lambda_{13} = \lambda_4\gamma & \lambda_{14} = \lambda_5\gamma & \lambda_{15} = \lambda_6\gamma & \lambda_{16} = \lambda_0\lambda_7 & \lambda_{17} = \lambda_1\lambda_7 \\
\lambda_{18} = \lambda_2\lambda_7 & \lambda_{19} = \lambda_3\lambda_7 & \lambda_{20} = \lambda_4\lambda_7 & \lambda_{21} = \lambda_5\lambda_7 & \lambda_{22} = \lambda_6\lambda_7 & \lambda_{23} = \beta\lambda_7.
\end{array}$$

D'où

$$\begin{aligned}
\pi(a) = {} & [(h_0 + h_1\lambda_1 u + h_2\lambda_4 u^2) + (h_3\beta + h_4\lambda_2 u + h_5\lambda_5 u^2)v + (h_6\lambda_0 + h_7\lambda_3 u \\
& + h_8\lambda_6 u^2)v^2] + [(h_9\gamma + h_{10}\lambda_{10} u + h_{11}\lambda_{13} u^2) + (h_{12}\lambda_8 + h_{13}\lambda_{11} u + h_{14}\lambda_{14} u^2)v \\
& + (h_{15}\lambda_9 + h_{16}\lambda_{12} u + h_{17}\lambda_{15} u^2)v^2]w + [(h_{18}\lambda_7 + h_{19}\lambda_{17} u + h_{20}\lambda_{20} u^2) \\
& + (h_{21}\lambda_{23} + h_{22}\lambda_{18} u + h_{23}\lambda_{21} u^2)v + (h_{24}\lambda_{16} + h_{25}\lambda_{19} u + h_{26}\lambda_{22} u^2)v^2]w^2.
\end{aligned}$$

Le coût de la puissance p de l'endomorphisme de Frobenius est : $26M_1 + 18A_1$. Pour le calcul de chacune des puissances p^2, p^4, p^5, p^7, p^8 de l'endomorphisme de Frobenius, la démarche est similaire à celle du calcul de la puissance p de l'endomorphisme de Frobenius. Ainsi, nous obtenons que les coût des puissances p^2, p^4, p^5, p^7 et p^8 de l'endomorphisme de Frobenius ont la même valeur que le coût de la puissance p de l'endomorphisme de Frobenius.

Pour la puissance p^9 de l'endomorphisme de Frobenius, on constate du calcul de l'inversion cyclotomique que $w^{p^9} = \sigma w$. Alors

$$\begin{aligned}
a^{p^9} = {} & a_0 + a_1\sigma w + a_2\sigma^2 w^2 \\
= {} & [(h_0 + h_1 u + h_2 u^2) + (h_3 + h_4 u + h_5 u^2)v + (h_6 + h_7 u + h_8 u^2)v^2] \\
+ {} & [(h_9 + h_{10} u + h_{11} u^2) + (h_{12} + h_{13} u + h_{14} u^2)v + (h_{15} + h_{16} u + h_{17} u^2)v^2]\sigma w \\
+ {} & [(h_{18} + h_{19} u + h_{20} u^2) + (h_{21} + h_{22} u + h_{23} u^2)v + (h_{24} + h_{25} u + h_{26} u^2)v^2]\sigma^2 w^2.
\end{aligned}$$

Ensuite nous avons :

$$\begin{aligned}
a^{p^9} = {} & [(h_0 + h_1 u + h_2 u^2) + (h_3 + h_4 u + h_5 u^2)v + (h_6 + h_7 u + h_8 u^2)v^2] \\
+ {} & [(h_9\sigma + h_{10}\sigma u + h_{11}\sigma u^2) + (h_{12}\sigma + h_{13}\sigma u + h_{14}\sigma u^2)v + (h_{15}\sigma + h_{16}\sigma u \\
+ {} & h_{17}\sigma u^2)v^2]w + [(h_{18}\sigma^2 + h_{19}\sigma^2 u + h_{20}\sigma^2 u^2) + (h_{21}\sigma^2 + h_{22}\sigma^2 u \\
+ {} & h_{23}\sigma^2 u^2)v + (h_{24}\sigma^2 + h_{25}\sigma^2 u + h_{26}\sigma^2 u^2)v^2]w^2.
\end{aligned}$$

$s = \sigma^2$ étant précalculé, nous aurons enfin :

$$\begin{aligned} a^{p^9} = \quad & [(h_0 + h_1u + h_2u^2) + (h_3 + h_4u + h_5u^2)v + (h_6 + h_7u + h_8u^2)v^2] \\ + \quad & [(h_9\sigma + h_{10}\sigma u + h_{11}\sigma u^2) + (h_{12}\sigma + h_{13}\sigma u + h_{14}\sigma u^2)v \\ + \quad & (h_{15}\sigma + h_{16}\sigma u + h_{17}\sigma u^2)v^2]w + [(h_{18}s + h_{19}su + h_{20}su^2) \\ + \quad & (h_{21}s + h_{22}su + h_{23}su^2)v + (h_{24}s + h_{25}su + h_{26}su^2)v^2]w^2. \end{aligned}$$

Le coût de la puissance p^9 de l'endomorphisme de Frobenius est : $18M_1 + 18A_1$. C'est la même valeur que celui des puissances p^3, p^6 de l'endomorphisme de Frobenius.

□

Dans le cas du corps $\mathbb{F}_{p^{15}}$, avec $p \equiv 1 \bmod 5$ d'après [DCC05]. Le polynôme $X^5 - \alpha$ est irréductible sur $\mathbb{F}_p[X]$ si et seulement si α n'est ni une racine cubique, ni une racine cinquième dans $\mathbb{F}_p$ d'après [LN97, Theorem 3.75]. Une tour d'extension pour $\mathbb{F}_{p^{15}}$ peut être construite comme suit :

$$\mathbb{F}_{p^5} = \mathbb{F}_p[u] \text{ avec } u^5 = 2$$

$$\mathbb{F}_{p^{15}} = \mathbb{F}_{p^5}[v] \text{ avec } v^3 = u \text{ où } u \in \mathbb{F}_{p^5}$$

Les coûts du calcul des puissances de l'endomorphismes de Frobenius et des inversions cyclotomiques sont donnés dans le Lemme 4.

Lemme 4. *Dans le coprs fini $\mathbb{F}_{p^{15}}$,*

- *L'inverse d'un élémet α de $G_{\phi_3(p^5)}$ est calculé comme $\alpha^{-1} = \alpha^{p^5}.\alpha^{p^{10}}$ et son coût est de $\mathbf{54S_1 + 822A_1}$.*
- *Le calcul de chacune des puissances p^5 ; p^{10} de l'endomorphisme de Frobenius coûte $\mathbf{10M_1 + 12A_1}$.*
- *Le calcul de chacune des puissances p ; p^2 ; p^3 ; p^4 ; p^6 ; p^7 ; p^8 ; p^9 de l'endomorphisme de Frobenius coûte $\mathbf{14M_1 + 12A_1}$.*

Preuve

Inversion cyclotomique : Un élément $a = a_0 + a_1v + a_2v^2 \in \mathbb{F}_{p^{15}}$ avec $a_i \in \mathbb{F}_{p^5}$ dans le sous groupe cyclotomique $G_{\phi_3(p^5)}$ satisfait l'égalité $a^{p^{10}+p^5+1} = 1$ i.e.

$$a^{-1} = a^{p^{10}}a^{p^5}.$$

Puisque $v^5 = 2^{1/3}$, nous avons :

$$v^{p^5} = v^{5(p^5-1)/5+1} = v^{5(p^5-1)/5}v = \left(v^5\right)^{(p^5-1)/5}v = \left(2^{1/3}\right)^{(p^5-1)/5}v.$$

Soit $\omega = \left(2^{1/3}\right)^{(p^5-1)/5}$, nous avons $\omega \neq 1$ et $\omega^5 = 1$. Donc ω est une racine cinquième de l'unité dans $\mathbb{F}_{p^5}$. Nous obtenons :

$$v^{p^5} = \omega v, \quad \text{et} \quad v^{p^{10}} = \left(v^{p^5}\right)^{p^5} = \left(\omega v\right)^{p^5} = \omega\left(v\right)^{p^5} = \omega\omega v = \omega^2 v.$$

$$\begin{aligned} a^{p^5} &= (a_0 + a_1 v + a_2 v^2)^{p^5} = a_0^{p^5} + a_1^{p^5} v^{p^5} + a_2^{p^5} (v^2)^{p^5} \\ &= a_0 + a_1 v^{p^5} + a_2 (v^2)^{p^5} \\ &= a_0 + a_1 \omega v + a_2 \omega^2 v^2. \end{aligned}$$

$$a^{p^{10}} = (a^{p^5})^{p^5} = a_0 + a_1 (\omega v)^{p^5} + a_2 (\omega^2 v^2)^{p^5} = a_0 + a_1 \omega^2 v + a_2 \omega^4 v^2.$$

$$a^{p^{10}} a^{p^5} = (a_0 + a_1 \omega^2 v + a_2 \omega^4 v^2)(a_0 + a_1 \omega v + a_2 \omega^2 v^2).$$

Après développement et réduction, sachant que $v^3 = u$ et $\phi_5(\omega) = 0$, nous obtenons

$$a^{p^{10}} a^{p^5} = (a_0^2 + (1+\omega^4) a_1 a_2 u) + \omega (a_2^2 u + (1+\omega) a_0 a_1) v + \omega^2 (a_1^2 \omega + (1+\omega^2) a_0 a_2) v^2$$

Donc le calcul de l'inverse d'un élément $a \in \mathbb{F}_{p^{15}}$ satisfaisant $a^{p^{10}+p^5+1} = 1$ nécessite 3 carrés dans $\mathbb{F}_{p^5}$, 3 multiplications dans $\mathbb{F}_{p^5}$, 3 additions dans $\mathbb{F}_{p^5}$. D'où son coût est $3(9M_1 + 137A_1) + 3(9M_1 + 137A_1) + 3(5A_1) = 54M_1 + 822A_1$.

Opérateurs Frobenius : La puissance p^i de l'endomorphisme de Frobenius est l'application $\pi^i : \mathbb{F}_{p^{15}} \to \mathbb{F}_{p^{15}}, a \mapsto a^{p^i}$.
Soit $a \in \mathbb{F}_{p^{15}}$; $a = a_0 + a_1 v + a_2 v^2$ avec $a_i \in \mathbb{F}_{p^5}$.

$$\pi(a) = a^p = (a_0 + a_1 v + a_2 v^2)^p = a_0^p + a_1^p v^p + a_2^p (v^2)^p.$$

$a_0 \in \mathbb{F}_{p^5}$ i.e. $a_0 = g_0 + g_1 u + g_2 u^2 + g_3 u^3 + g_4 u^4, \quad g_i \in \mathbb{F}_p$.

$$a_0^p = (g_0 + g_1 u + g_2 u^2 + g_3 u^3 + g_4 u^4)^p = g_0 + g_1 u^p + g_2 (u^2)^p + g_3 (u^3)^p + g_4 (u^4)^p$$

car $g_i^p = g_i$. Nous avons aussi :

$$u^p = u^{5(p-1)/5+1} = (u^5)^{(p-1)/5} u = 2^{(p-1)/5} u.$$

Puisque 2 n'est pas une puissance cinquième dans $\mathbb{F}_p$; nous avons ainsi $2^{(p-1)/5} \neq 1$. Soit $\theta = 2^{(p-1)/5}$, $\theta \neq 1$ et $\theta^5 = 1$. C'est-à-dire que θ est une racine cinquième primitive de l'unité dans $\mathbb{F}_p$ et $u^p = \theta u$.

$$a_0^p = g_0 + g_1 u^p + g_2 (u^2)^p + g_3 (u^3)^p + g_4 (u^4)^p = g_0 + g_1 \theta u + g_2 \theta^2 u^2 + g_3 \theta^3 u^3 + g_4 \theta^4 u^4.$$

$$a_1^p = g_5 + g_6 u^p + g_7 (u^2)^p + g_8 (u^3)^p + g_9 (u^4)^p = g_5 + g_6 \theta u + g_7 \theta^2 u^2 + g_8 \theta^3 u^3 + g_9 \theta^4 u^4.$$

$$a_2^p = g_{10} + g_{11} u^p + g_{12} (u^2)^p + g_{13} (u^3)^p + g_{14} (u^4)^p = g_{10} + g_{11} \theta u + g_{12} \theta^2 u^2 + g_{13} \theta^3 u^3 + g_{14} \theta^4 u^4.$$

Par ailleurs, nous avons

$$v^p = v^{5(p-1)/5+1} = (v^5)^{(p-1)/5} v = (2^{1/3})^{(p-1)/5} v = (2^{1/3})^{(p-1)/5} v.$$

$2^{1/3}$ n'est pas une puissance cinquième dans $\mathbb{F}_p$; ainsi $(2^{1/3})^{(p-1)/5} \neq 1$.
Posons $\beta = (2^{1/3})^{(p-1)/5}$, nous avons $\beta \neq 1$; $\beta^5 = 1$. D'où β est une racine cinquième primitive de l'unité dans $\mathbb{F}_p$ et $v^p = \beta v$.

$$\begin{aligned}
a^p = & (a_0 + a_1 v + a_2 v^2)^p = a_0^p + a_1^p v^p + a_2^p (v^2)^p \\
= & (g_0 + g_1\theta u + g_2\theta^2 u^2 + g_3\theta^3 u^3 + g_4\theta^4 u^4) \\
+ & (g_5 + g_6\theta u + g_7\theta^2 u^2 + g_8\theta^3 u^3 + g_9\theta^4 u^4) v^p \\
+ & (g_{10} + g_{11}\theta u + g_{12}\theta^2 u^2 + g_{13}\theta^3 u^3 + g_{14}\theta^4 u^4)(v^p)^2 \\
a^p = & (g_0 + g_1\theta u + g_2\theta^2 u^2 + g_3\theta^3 u^3 + g_4\theta^4 u^4) \\
+ & (g_5\beta + g_6\theta\beta u + g_7\theta^2\beta u^2 + g_8\theta^3\beta u^3 + g_9\theta^4\beta u^4) v \\
+ & (g_{10}\beta^2 + g_{11}\theta\beta^2 u + g_{12}\theta^2\beta^2 u^2 + g_{13}\theta^3\beta^2 u^3 + g_{14}\theta^4\beta^2 u^4) v^2.
\end{aligned}$$

Nous précalculons les valeurs suivantes :

$$\begin{array}{cccccc}
c_0 = \theta^2 & c_1 = \theta^3 & c_2 = \theta^4 & c_3 = \beta^2 & c_4 = \theta\beta & c_5 = c_0\beta \\
c_6 = c_1\beta & c_7 = c_2\beta & c_8 = \theta c_3 & c_9 = c_0 c_3 & c_{10} = c_1 c_3 & c_{11} = c_2 c_3
\end{array}$$

Ainsi

$$\begin{aligned}
\pi(a) = & (g_0 + g_1\theta u + g_2 c_0 u^2 + g_3 c_1 u^3 + g_4 c_2 u^4) + (g_5\beta + g_6 c_4 u + g_7 c_5 u^2 \\
& + g_8 c_6 u^3 + g_9 c_7 u^4) v + (g_{10} c_3 + g_{11} c_8 u + g_{12} c_9 u^2 + g_{13} c_{10} u^3 + g_{14} c_{11} u^4) v^2.
\end{aligned}$$

Le coût de la puissance p de l'endomorphisme de Frobenius est : $14M_1 + 12A_1$. C'est la même valeur que celui des puissances $p^2, p^3, p^4, p^6, p^7, p^8, p^9$ de l'endomorphisme de Frobenius.

Pour la puissance p^5 de l'endomorphisme de Frobenius, nous constatons du calcul de l'inversion cyclotomique que $v^{p^5} = \omega v$ d'après le calcul de l'inverse cyclotomique. Alors

$$\begin{aligned}
a^{p^5} = & (g_0 + g_1 u + g_2 u^2 + g_3 u^3 + g_4 u^4) + (g_5 + g_6 u + g_7 u^2 + g_8 u^3 + g_9 u^4) v^{p^5} \\
& + (g_{10} + g_{11} u + g_{12} u^2 + g_{13} u^3 + g_{14} u^4)(v^{p^5})^2 \\
= & (g_0 + g_1 u + g_2 u^2 + g_3 u^3 + g_4 u^4) + (g_5\omega + g_6\omega u + g_7\omega u^2 + g_8\omega u^3 + g_9\omega u^4) v \\
& + (g_{10}\omega^2 + g_{11}\omega^2 u + g_{12}\omega^2 u^2 + g_{13}\omega^2 u^3 + g_{14}\omega^2 u^4) v^2.
\end{aligned}$$

Nous précalculons $d = \omega^2$.

$$\begin{aligned}
\pi^5(a) = & a^{p^5} = (g_0 + g_1 u + g_2 u^2 + g_3 u^3 + g_4 u^4) + (g_5\omega + g_6\omega u + g_7\omega u^2 + g_8\omega u^3 \\
& + g_9\omega u^4) v + (g_{10} d + g_{11} d u + g_{12} d u^2 + g_{13} d u^3 + g_{14} d u^4) v^2.
\end{aligned}$$

Le coût de la puissance p^5 de l'endomorphisme de Frobenius : $10M_1 + 12A_1$. C'est la même valeur que celui de la puissance p^{10} de l'endomorphisme de Frobenius. □

Notre contribution, ici, est le calcul des puissances de l'endomorphisme de Frobenius et des inversions dans les sous groupes cyclotomiques d'ordre $\phi_n(.)$ de $\mathbb{F}^{\star}_{p^k}$. Un récapitulatif des coûts des opérations dans les tours d'extension de corps décrites ci-dessus est dressé dans la table 3.2. Les coûts du calcul d'un carré, d'une multiplication et d'une inversion proviennent de [LT12], [MGI09] et [ZL12] respectivement pour $k = 9, 15$ et 27. Les détails explicites du coût des puissances de l'endomorphisme de Frobenius et des inversions dans les sous groupes cyclotomiques sont donnés respectivement dans le Lemme 2, le Lemme 3 et le Lemme 4.

Corps	Opérations	Coût littérature	Coût de notre contribution
$\mathbb{F}_{p^3}$	Multiplication M_3	$6M_1 + 17A_1$ [BS04]	
	Elévation au carrée S_3	$6S_1 + 8A_1$ [BS04]	
	Inversion I_3	$I_1 + 9M_1 + 2S_1$ [LMN10]	
$\mathbb{F}_{p^9}$	Multiplication M_9	$36M_1 + 149A_1$ [BS04]	
	Elévation au carrée S_9	$36S_1 + 95A_1$[BS04]	
	Inversion I_9	$I_1 + 63M_1 + 14S_1$ [ZL12]	
	Frobenius $p^3; p^6$		$6M_1 + 6A_1$ [Lemme 2]
	Frobenius $p; p^2; p^4; p^5; p^7; p^8$		$8M_1 + 6A_1$ [Lemme 2]
	Inversion dans $G_{\phi_3(p^3)}$		$36S_1 + 84A_1$ [Lemme 2]
$\mathbb{F}_{p^{27}}$	Multiplication M_{27}	$216M_1 + 1031A_1$ [BS04]	
	Elévation au carrée S_{27}	$216S_1 + 788A_1$ [BS04]	
	Inversion I_{27}	$I_1 + 387M_1 + 86S_1$ [ZL12]	
	Frobenius $p^3; p^6; p^9$		$18M_1 + 18A_1$ [Lemme 3]
	Frobenius $p; p^2; p^4; p^5; p^7; p^8$		$26M_1 + 18A_1$ [Lemme 3]
	Inversion dans $G_{\phi_3(p^9)}$		$216S_1 + 759A_1$ [Lemme 3]
$\mathbb{F}_{p^5}$	Multiplication M_5	$9M_1 + 137A_1$ [MGI11]	
	Elévation au carrée S_5	$9S_1 + 137A_1$ [MGI11]	
	Inversion I_5	$1I_1 + 45M_1 + 5S_1$ [MGI11]	
$\mathbb{F}_{p^{15}}$	Multiplication M_{15}	$45M_1 + 635A_1$ [MGI09]	
	Elévation au carrée S_{15}	$45S_1 + 635A_1$ [MGI09]	
	Inversion I_{15}	$I_1 + 126M_1 + 23S_1 + 1507A_1$ [MGI09]	
	Frobenius $p^5; p^{10}$		$10M_1 + 12A_1$ [Lemme 4]
	Frobenius $p; p^2; p^3; p^4; p^6; p^7; p^8; p^9$		$14M_1 + 12A_1$ [Lemme 4]
	Inversion dans $G_{\phi_3(p^5)}$		$54S_1 + 837A_1$ [Lemme 4]

TABLE 3.2 – Coût des opérations dans les extensions de corps $\mathbb{F}_{p^9}$, $\mathbb{F}_{p^{15}}$ et $\mathbb{F}_{p^{27}}$.

3.3 Courbes elliptiques de degré de plongement 9

Cette section décrit le calcul du couplage optimal Ate (les étapes de l'algorithme de Miller et l'exponentiation finale) sur les courbes elliptiques parametrées définies dans [LZZW08]. Pour ces courbes elliptiques, de courbes elliptiques $\rho = 1.33$, le degré de plongement vaut 9 et les parametres sont :

$$\begin{aligned} p(x) &= ((x+1)^2 + ((x-1)^2(2x^3+1)^2)/3)/4, \\ r(x) &= (x^6 + x^3 + 1)/3, \\ t(x) &= x + 1. \end{aligned} \tag{3.3.1}$$

Cette famille de courbe est la plus efficace pour ce niveau de sécurité.

3.3.1 Couplage optimal Ate

Le couplage optimal Ate étant construit en utilisant la méthode basée sur les réseaux proposée par Vercauteren dans [Ver10], le court vecteur obtenu du réseau L définie par (1.2.3) donne le polynôme optimal $h(z) = \sum_{i=0}^{5} c_i z^i = x - z \in \mathbb{Z}[z]$. Une application directe de la formule (1.2.2) conduit au couplage optimal :

$$\begin{aligned} e_9 : \ \mathbb{G}_2 \times \mathbb{G}_1 &\longrightarrow \mu_r \\ (Q, P) &\longmapsto f_{x,Q}(P)^{\frac{p^9-1}{r}} \end{aligned}$$

3.3.2 Coût de l'exécution de la boucle de Miller

La boucle de Miller consiste en des étapes de doublement et des étapes d'additions. Ces étapes utilisent la fonction $h_{R,S} = \frac{\ell_{R,S}}{v_{R+S}}$ soit en coordonnées affines, soit en coordonnées projectives avec R et S des points de la courbe elliptique. Les travaux de Zhang *et al.* [ZL12, Section 3] présentent de récentes formules favorisant des calculs rapides en coordonnés projectives. L'étape de doublement coûte $9M_1 + 3M_3 + 9S_3$ et le coût de l'étape d'addition est de $9M_1 + 12M_3 + 5S_3$. Pour un coût explicite du calcul de $f_{x,Q}(P)$, le code Pari/GP se trouvant à l'appendice A.1 permet de trouver un x convenable avec un faible poids de hamming et un nombre minimal de bits pour le niveau de sécurité de 128 bits d'après la table 3.1. La meilleure valeur obtenue grâce à ce code est $x = 2^{43} + 2^{37} + 2^7 + 1$ tel que $r(x)$ soit un nombre premier de 257 bits et $p(x)$ un nombre premier de 343 bits. Les valeurs de p et x sont toutes deux congruent à 1 modulo 6 et la courbe elliptique correspondante a pour équation $y^2 = x^3 + 1$ [LT12]. Calculer $f_{x,Q}(P)$ revient à exécuter l'algorithme 1 présenté à la section 1.2.3. Par conséquent, le calcul de $f_{x,Q}(P)$

nécessite 43 étapes de doublements, 3 additions, 42 carrés et 45 multiplications dans $\mathbb{F}_{p^9}$. Donc le coût total pour le calcul de la boucle de Miller du couplage optimal Ate pour les courbes elliptiques de degré de plongement 9 est $43(9M_1+3M_3+9S_3)+3(9M_1+12M_3+5S_3)+42S_9+45M_9$. Ce qui est égal à $45M_9+165M_3+414M_1+42S_9+402S_3$. En utilisant les données arithmétiques consignées dans la table 3.2, le coût global est $3024m_{343}+3924s_{343}$. A notre connaissance, aucun coût explicite avec une valeur spécifique de x n'a été reporté dans la littérature.

3.3.3 Coût du calcul de l'exponentiation finale

Suivant les explications données dans la section 1.2.5, l'exponentiation finale dans ce cas peut être subdivisée comme suit

$$f^{(p^9-1)/r} = \left(f^{p^3-1}\right)^{(p^6+p^3+1)/r} = \left(f^{p^3-1}\right)^d.$$

La valeur de $d = \phi_9(p)/r = (p^6+p^3+1)/r$ peut être exprimée de façon polynômiale par

$d = d(x) = 1/243x^{42} - 4/81x^{41} + 22/81x^{40} - 215/243x^{39} + 145/81x^{38} - 154/81x^{37} - 161/243x^{36} + 169/27x^{35} - 295/27x^{34} + 1940/243x^{33} + 361/81x^{32} - 1459/81x^{31} + 4910/243x^{30} - 640/81x^{29} - 710/81x^{28} + 4262/243x^{27} - 137/9x^{26} + 19/3x^{25} + 290/81x^{24} - 257/27x^{23} + 215/27x^{22} - 124/81x^{21} - 76/27x^{20} + 70/27x^{19} - 79/81x^{18} - 1/3x^{17} + 28/9x^{16} - 1246/243x^15 + 151/81x^{14} + 245/81x^{13} - 976/243x^{12} + 176/81x^{11} - 104/81x^{10} + 482/243x^9 + 8/27x^8 - 41/27x^7 - 107/243x^6 + 20/81x^5 + 61/81x^4 - 431/243x^3 + 61/81x^2 + 29/81x + 757/243.$

L'écriture de d en base p est la suivante :

$d(x) = x^7 - 2x^6 + x^5 + x^4 - 2x^3 + x^2 + 3 + (x^6 - 2x^5 + x^4 + x^3 - 2x^2 + x)p(x) + (x^5 - 2x^4 + x^3 + x^2 - 2x + 1)p(x)^2 + (x^4 - 2x^3 + x^2)p(x)^3 + (x^3 - 2x^2 + x)p(x)^4 + (x^2 - 2x + 1)p(x)^5.$

En appliquant la méthode basée sur les réseaux décrite par Fuentes *et al.* [FKR11] sur la matrice

$$M = \begin{pmatrix} 243d(x) \\ 243xd(x) \\ 243x^2d(x) \\ 243x^3d(x) \\ 243x^4d(x) \\ 243x^5d(x) \end{pmatrix}, \tag{3.3.2}$$

nous trouvons le multiple d' de d :

$$d' = x^3d = k_0 + k_1p + k_2p^2 + k_3p^3 + k_4p^4 + k_5p^5,$$

les polynômes $k_i, i = 0, ..., 5$ étant définis par :

$$k_0 = -x^4 + 2x^3 - x^2, \quad k_1 = -x^3 + 2x^2 - x, \quad k_2 = -x^2 + 2x - 1,$$
$$k_3 = x^7 - 2x^6 + x^5 + 3, \quad k_4 = x^6 - 2x^5 + x^4, \quad k_5 = x^5 - 2x^4 + x^3.$$

Ils vérifient les relations

$$k_2 = -(x-1)^2, \quad k_1 = xk_2, \quad k_0 = xk_1, \quad k_5 = -xk_0, \quad k_4 = xk_5, \quad k_3 = xk_4 + 3.$$

Si nous posons $A = f^{p^3-1}$ alors

- Le coût du calcul de A est de 1 p^3-Frobenius, 1 Inversion dans $\mathbb{F}_{p^9}$ et 1 multiplication dans $\mathbb{F}_{p^9}$.
- Le coût du calcul de A^{k_0}, A^{k_1} et A^{k_4} est de 3 exponentiations par x,
- Le coût du calcul de A^{k_5} est d'une inversion dans le sous groupe cyclotomique et d'une exponentiation par x.
- Le coût du calcul de A^{k_2} est d'une inversion dans le sous groupe cyclotomique et de deux exponentiations par $(x-1)$.
- Le coût du calcul de A^{k_3} est de 2 multiplications, d'un carrée et d'une exponentiation par x.

Remarquons que l'inversion dans le sous groupe cyclotomique $\mathbb{G}_{\phi_3(p^3)}$ d'ordre p^6+p^3+1 se calcule de la façon suivante $A^{-1} = A^{p^3} \cdot A^{p^6}$ (voir Lemme 2 pour plus de détails). Le coût de la partie difficile $A^{d'}$ est alors 2 exponentiations par $x-1$, 5 exponentiations par x, 7 multiplications dans $\mathbb{F}_{p^9}$, une élévation au carré dans $\mathbb{F}_{p^9}$, 2 inversions cyclotomiques $I_{\mathbb{G}_{\phi_3(p^3)}}$ et p, p^2, p^3, p^4, p^5-endomorphismes de Frobenius. En utilisant la valeur de x donnée plus haut, une exponentiation par x coûte $43S_9+3M_9$ tandis qu'une exponentiation par $x-1$ coûte $43S_9+2M_9$. Finalement, la partie difficile coûte $2(43S_9+2M_9)+5(43S_9+3M_9)+7M_9+1S_9+2I_{\mathbb{G}_{\phi_3(p^3)}} = 26M_9+302S_9+2I_{\mathbb{G}_{\phi_3(p^3)}}$ et p, p^2, p^3, p^4, p^5-Endomorphismes de Frobenius. Le coût total de l'exponentiation finale est $1I_9+27M_9+302S_9+2I_{\mathbb{G}_{\phi_3(p^3)}}$ et $p, p^2, 2*p^3, p^4, p^5$-endomorphismes de Frobenius.

3.3.4 Amélioration et comparaison avec les résultats des travaux existants

A partir des résultats de [LT12], l'exponentiation finale vaut $1I_9+51M_9+309S_9$ et $p, p^2, 2*p^3, p^4, p^5$-Frobenius pour les travaux de Le et *al.* En utilisant l'arithmétique de la table 3.2, on obtient donc le coût total de l'exponentiation finale est de $i_{343}+1079m_{343}+10958s_{343}$ pour notre travail et de $i_{343}+1943m_{343}+11138s_{343}$ pour Le et *al.*[LT12].
Nous avons donc supprimé $24M_9+7S_9-2I_{\mathbb{G}_{\phi_3(p^3)}} = 864m_{343}+180s_{343}$ comparativement aux travaux de Le et *al.*

3.4 Courbes elliptiques avec degré de plongement 15

Dans cette section, des formules explicites pour les étapes de la boucle de Miller ainsi que leurs coûts sont données et le coût de l'exponentiation finale du couplage optimal Ate sur les courbes elliptiques paramétrées définies dans [DCC05] est calculé. Pour ces courbes elliptiques, $\rho = 1.5$, le degré de plongement vaut 15 et les paramètres sont :

$$\begin{aligned} p(x) &= (x^{12} - 2x^{11} + x^{10} + x^7 - 2x^6 + x^5 + x^2 + x + 1)/3, \\ r(x) &= x^8 - x^7 + x^5 - x^4 + x^3 - x + 1, \\ t(x) &= x + 1. \end{aligned} \tag{3.4.1}$$

3.4.1 Couplage optimal Ate

L'approche décrite par Vercauteren dans [Ver10] permet d'obtenir un court vecteur à partir du réseau L défini par (1.2.3) qui mène au polynôme optimal $h(z) = \sum_{i=0}^{5} c_i z^i = x - z \in \mathbb{Z}[z]$. Une application directe de la formule (1.2.2) définie un couplage optimal :

$$\begin{aligned} e_o : \ \mathbb{G}_2 \times \mathbb{G}_1 &\longrightarrow \mu_r \\ (Q, P) &\longmapsto f_{x,Q}(P)^{\frac{p^{15}-1}{r}} \end{aligned}$$

3.4.2 Coût de l'exécution de la boucle de Miller

La fonction $h_{R,S} = \frac{\ell_{R,S}}{v_{R+S}}$ utilisée pour le calcul de $f_{x,Q}(P)$ dans ce cas est décrite dans [ZL12] avec R et S des points de la courbe elliptique. Pour un niveau de sécurité de 192 bits sur les courbes elliptiques de degré de plongement $k = 15$, la meilleure valeur de x obtenu à partir du code Pari/GP se trouvant à l'appendice A.2 est $x = 2^{48} + 2^{41} + 2^9 + 2^8 + 1$. De cette valeur, on obtient que $r(x)$ est un nombre premier de 385 bits et $p(x)$ un nombre premier de 575 bits, paramètres permettant d'obtenir, d'après d'après la table 3.1, un niveau de sécurité de 192 bits. p est congru à 1 modulo 5. La boucle de Miller calcule ici $f_{x,Q}$; ce qui nécessite 48 étapes de doublement, 4 étapes d'addition, 47 élévations au carré et 51 multiplications dans $\mathbb{F}_{p^{15}}$. En considérant les récents coûts les plus rapides obtenus pour les étapes de doublement et d'addition dans [ZL12] donnés en coordonnées projectives, la boucle de Miller coûte
$48(15M_1 + 3M_5 + 9S_5) + 4(15M_1 + 12M_5 + 5S_5) + 47S_{15} + 51M_{15}$,
soit $51M_{15} + 192M_5 + 780M_1 + 47S_{15} + 452S_5$. En utilisant l'arithmétique de la table 3.2, le coût total est de $4803m_{575} + 6183s_{575}$. A notre connaissance, aucun coût explicite n'est reporté dans la littérature pour le cas $k = 15$ avec une valeur spécifique de x.

3.4.3 Coût du calcul de l'exponentiation finale

L'exponentiation finale dans ce cas est écrite d'une façon différente :

$$f^{(p^{15}-1)/r} = \left(f^{p^5-1}\right)^{(p^{10}+p^5+1)/r} = \left(f^{p^5-1}\right)^d.$$

Cette décomposition est utilisée au lieu de la décomposition habituelle

$$\frac{p^{15}-1}{r} = \left[\frac{p^{15}-1}{\phi_{15}(p)}\right] \cdot \left[\frac{\phi_{15}(p)}{r}\right]$$

pour des raisons d'efficacité lors du calcul. Observons que $\frac{p^{15}-1}{\phi_{15}(p)} = p^7+p^6+p^5-p^2-p-1$ et $\phi_{15}(p) = p^8-p^7+p^5-p^4+p^3-p+1$ mèneront à plusieurs opérations de multiplications et d'endomorphismes de Frobenius. La valeur de $d = (p^{10}+p^5+1)/r$ exprimé en base p est $1/3x^{11}-2/3x^{10}+1/3x^9+1/3x^6-2/3x^5+1/3x^4-1/3x^3+1/3x^2+1/3x+2/3+(1/3x^{11}-1/3x^{10}-1/3x^9+1/3x^8+1/3x^6-1/3x^5-1/3x^4+1/3x^3-1/3x^2+2/3x+2/3)p(x)+(1/3x^11-1/3x^{10}-1/3x^8+1/3x^7+1/3x^6-1/3x^5-1/3x^3+1/3x^2+1)p(x)^2+(1/3x^{10}-1/3x^9-1/3x^7+1/3x^6+1/3x^5-1/3x^4-1/3x^2+1/3x)p(x)^3+(1/3x^9-1/3x^8-1/3x^6+1/3x^5+1/3x^4-1/3x^3-1/3x+1/3)p(x)^4+(1/3x^8-1/3x^7-1/3x^5+1/3x^4)p(x)^5+(1/3x^7-1/3x^6-1/3x^4+1/3x^3)p(x)^6+(1/3x^6-1/3x^5-1/3x^3+1/3x^2)p(x)^7+(1/3x^5-1/3x^4-1/3x^2+1/3x)p(x)^8+(1/3x^4-1/3x^3-1/3x+1/3)p(x)^9$.
La méthode basée sur les réseaux décrite par Fuentes *et al.* [FKR11], brièvement rappelée à la section 1.2.5, appliquée à la matrice

$$M = \begin{pmatrix} \frac{59049}{19683}d(x) \\ \frac{59049}{19683}xd(x) \\ \frac{59049}{19683}x^2d(x) \\ . \\ . \\ . \\ \frac{59049}{19683}x^7d(x) \end{pmatrix}, \tag{3.4.2}$$

nous trouvons le multiple d' de d :

$$d' = 3x^3d = k_0 + k_1p + ... + k_9p^9,$$

les polynômes $k_i, i = 0, ..., 9$ étant définis par :

$$\begin{array}{lll} k_0 = -x^6+x^5+x^3-x^2, & k_1 = -x^5+x^4+x^2-x, & k_2 = -x^4+x^3+x-1, \\ k_3 = x^{11}-2x^{10}+x^9+x^6-2x^5+x^4-x^3+x^2+x+2, & & k_5 = x^{11}-x^{10}-x^8+x^7+3 \\ k_4 = x^{11}-x^{10}-x^9+x^8+x^6-x^5-x^4+x^3-x^2+2x+2, & & k_6 = x^{10}-x^9-x^7+x^6 \\ k_7 = x^9-x^8-x^6+x^5, & k_8 = x^8-x^7-x^5+x^4, & k_9 = x^7-x^6-x^4+x^3. \end{array}$$

Les polynômes $k_i : i = 0, ..., 9$ vérifient les relations

$k_2 = -(x-1)^2(x^2+x+1)$, $k_1 = xk_2$, $k_0 = xk_1$, $k_9 = -xk_0$, $k_8 = xk_9$
$k_7 = xk_8, k_6 = xk_7, k_5 = xk_6 + 3$, $k_4 = M - (k_1 + k_7)$, $k_3 = M - (k_0 + k_6 + k_9)$
où $M = (k_2 + k_5 + k_8)$.

Posons $A = f^{p^5-1}$; alors

— Le coût du calcul de A vaut 1 p^5-Frobenius, 1 Inversion dans $\mathbb{F}_{p^{15}}$ et 1 multiplication dans $\mathbb{F}_{p^{15}}$.
— Le coût du calcul de A^{k_2} vaut 2 exponentiations par x, 2 exponentiations par $x-1$, 2 multiplications et 1 inversion cyclotomique.
— Le coût du calcul de A^{k_0}, A^{k_1}, A^{k_6}, A^{k_7} vaut 5 exponentiations par x et le calcul de A^{k_9} coûte 1 exponentiation par x et 1 inversion cyclotomique.
— Le coût du calcul de A^{k_5} vaut 1 exponentiation par x, 2 multiplications et un carré dans $\mathbb{F}_{p^{15}}$.
— Le coût du calcul de A^{k_4} vaut 4 multiplications dans $\mathbb{F}_{p^{15}}$ et une inversion cyclotomique.
— Le coût du calcul de A^{k_3} vaut 3 multiplications dans $\mathbb{F}_{p^{15}}$ et une inversion cyclotomique.

Donc le coût du calcul de $A^{d'}$ est de 2 exponentiations par $x-1$, 9 exponentiations par x, 20 multiplications, un carré dans $\mathbb{F}_{p^{15}}$, 4 inversions dans le sous groupe cyclotomique $\mathbb{G}_{\phi_3(p^5)}$ d'ordre $p^{10}+p^5+1$ et les puissances $p; p^2; p^3; p^4; p^5; p^6; p^7; p^8; p^9$ de l'endomorphisme de Frobenius. En utilisant la valeur de x donnée plus haut, le coût de la partie difficile est de :

$$2(48S_{15}+3M_{15})+9(48S_{15}+4M_{15})+20M_{15}+1S_{15}+4I_{\mathbb{G}_{\phi_3(p^5)}} = 529S_{15}+62M_{15}+4I_{\mathbb{G}_{\phi_3(p^5)}}$$

et les puissances $p; p^2; p^3; p^4; p^5; p^6; p^7; p^8; p^9$ de l'endomorphisme de Frobenius. Le coût total de l'exponentiation finale dans ce travail est donc $1I_{15} + 529S_{15} + 63M_{15} + 4I_{\mathbb{G}_{\phi_3(p^5)}}$ et les puissances $p, p^2, p^3, p^4, 2 * p^5, p^6, p^7, p^8, p^9$ de l'endomorphisme de Frobenius.

Remarque 7. *Le coût donné par Le* et al. *[LT12] pour la partie difficile est de* 11 *exponentiations par x,* 22 *multiplications,* 2 *inversions dans* $\mathbb{F}_{p^{15}}$ *et* 9 *endomorphismes de Frobenius. Ces auteurs affirment que le coût d'une inversion dans* $\mathbb{F}_{p^{15}}$ *est négligeable en s'appuyant sur un calcul similaire mais sur les courbes elliptiques de degré de plongement pair, malheureusement, nous ne voyons pas comment cela puisse être possible. En outre, ils considèrent un x de la taille de* 64 *bits, de poids de hamming* 7 *et prétendent que le coût est de* $88M_{15} + 528S_{15}$ *au lieu de* $11(6M_{15} + 64S_{15}) = 88M_{15} + 704S_{15}$. *Si nous comptons les* 2 *inversions dans* $\mathbb{F}_{p^{15}}$ *(ces inverses sont en fait dans le sous groupe cyclotomique* $\mathbb{G}_{\phi_3(p^5)}$*), leur coût final donnera* $88M_{15} + 704S_{15} + 2I_{\mathbb{G}_{\phi_3(p^5)}}$ *et* 11 *endomorphismes de Frobenius, tandis que le nôtre vaut* $62M_{15} + 529S_{15} + 4I_{\mathbb{G}_{\phi_3(p^5)}}$.

3.4.4 Amélioration et comparaison avec les résultats des travaux existants

En considérant la remarque précédente, le coût de l'exponentiation finale dans [LT12] est de $1I_{15} + 704S_{15} + 89M_{15} + 2I_{\mathbb{G}_{\phi_3(p^5)}}$ et les puissances $p, p^2, p^3, p^4, 2*p^5, p^6, p^7, p^8, p^9$ de l'endomorphisme de Frobenius. En utilisant l'arithmétique de la table 3.2, le coût total donne $i_{575} + 3093m_{575} + 24044s_{575}$ pour notre travail et $i_{575} + 4263m_{575} + 31811s_{575}$ pour Le et *al.* [LT12]. Nous constatons que nous avons amélioré leur résultat en supprimant $26M_{15} + 175S_{15} - 2I_{\mathbb{G}_{\phi_3(p^5)}} = 1170m_{575} + 7767s_{575}$ comparativement au résultat de Le et *al.*

3.5 Courbes elliptiques de degré de plongement 27

Les courbes elliptiques paramétrées de degré de plongement 27 sont définies dans [BLS02]. Pour cette famille, $\rho = 10/9$ et les paramètres sont définis par :

$$\begin{array}{lcl} p(x) & = & 1/3(x-1)^2(x^{18}+x^9+1)+x, \\ r(x) & = & 1/3(x^{18}+x^9+1), \\ t(x) & = & x+1. \end{array} \tag{3.5.1}$$

3.5.1 Coût de l'exécution de la boucle de Miller et du calcul de l'exponentiation finale

La boucle de Miller et l'exponentiation finale ont été étudiées dans [ZL12]. Le polynôme optimal obtenu a pour expression $h(z) = \sum_{i=0}^{17} c_i z^i = x - z \in \mathbb{Z}[z]$ et le couplage optimal Ate est donné par

$$e_o : \mathbb{G}_2 \times \mathbb{G}_1 \longrightarrow \mu_r; (Q, P) \longmapsto f_{x,Q}(P)^{\frac{p^{27}-1}{r}}.$$

Les auteurs de [ZL12] ont utilisé le paramètre $x = 2^{28} + 2^{27} + 2^{25} + 2^8 - 2^3$ pour leur calcul avec un niveau de sécurité de 256-bits. Le coût des étapes de Miller qu'ils ont obtenu est de $28(3M_9 + 2S_9 + 1I_9 + 9M_1) + 4(3M_9 + 2S_9 + 1I_9 + 9M_1) + 27(6S_9) + 30(6M_9) + 1I_{27} = 276M_9 + 226S_9 + 32I_9 + 288M_1 + 1I_{27}$ opérations. Le calcul de l'exponentiation finale dans [ZL12] nécessite $1I_{27} + 12M_{27}$, 17 exponentiations par x, 2 exponentiations par $x - 1$ et les puissances $p, p^2, p^3, p^4, p^5, p^6, p^7, p^8, 2*p^9$ de l'endomorphisme de Frobenius. Donc le coût explicite de l'exponentiation finale est de $1I_{27} + 17(5(6M_9) + 28(6S_9) + 36M_1) + 2(6(6M_9) + 28(6S_9) + 36M_1) + 11(6M_9) + 228M_1 = 1I_{27} + 648M_9 + 3192S_9 + 912M_1$.

Alors le coût explicite pour le calcul de la boucle de Miller et l'exponentiation finale données dans leur travail sont de $12627m_{573} + 8670s_{573} + 33i_{573}$ et $24627m_{573} + 114998s_{573} + 1i_{573}$ respectivement.

Remarque 8. *Le coefficient négatif figurant dans l'expression de x affecte l'efficacité du calcul du couplage, puisqu'une inversion dans $\mathbb{F}_{p^{27}}$ est requise lors de l'exécution de l'algorithme de Miller ainsi que* 19 *inversions dans le sous groupe cyclotomique sont requises lors de l'élévation à la puissance x pendant l'exponentiation finale.*

Dans la section suivante, nous montrons comment le choix d'un autre paramètre permet d'éviter ces opérations additionnelles.

3.5.2 Amélioration et comparaison avec les résultats des travaux existants

Dans cette section, le calcul d'inverse dans les sous groupes cyclotomiques ainsi qu'une nouvelle valeur spécifique de x obtenue à l'aide du code Pari/Gp se trouvant à l'appendice A.3, sont utilisés pour améliorer les coûts obtenus par [ZL12]. Précisement, une recherche minutieuse avec ce code Pari/GP a permis de trouver que pour $x = 2^{29}+2^{19}+2^{17}+2^{14}$ telle que r possède un facteur premier de taille de 514 bits, et déterminer un nombre premier p d'une taille de 579 bits qui, conjointement avec cette valeur de x, offre, en vertu de la table 3.1, un niveau de sécurité de 256 bits. Bien que nous ayons une étape de doublement en plus, nous évitons l'inversion dans $\mathbb{F}_{p^{27}}$ et 17 inversions dans le sous groupe cyclotomique $\mathbb{G}_{\phi_3(p^9)}$ lors de l'élévation à la puissance x. Nous effectuons seulement 2 inversions dans le sous groupe cyclotomiques lors de l'élévation à la puissance $x-1$. Le coût de la boucle de Miller revient alors à $29(3M_9 + 2S_9 + 1I_9 + 9M_1) + 3(3M_9 + 2S_9 + 1I_9 + 9M_1) + 27(6S_9) + 30(6M_9) = 276M_9 + 226S_9 + 32I_9 + 288M_1$. En utilisant l'arithmétique de la table 3.2, le coût total pour la boucle de Miller est de $32i_{576} + 12240m_{579} + 8584s_{579}$ pour ce travail : nous avons ainsi supprimé une inversion dans $\mathbb{F}_{p^{27}}$.

Notre coût de l'exponentiation finale est de $1I_{27} + 17(3(6M_9) + 29(6S_9)) + 2(4(6M_9) + 29(6S_9) + 2I_{\mathbb{G}_{\phi_3(p^9)}}) + 11(6M_9) = 1I_{27} + 420M_9 + 3306S_9 + 2I_{\mathbb{G}_{\phi_3(p^9)}}$ et les puissances $p, p^2, p^3, p^4, p^5, p^6, p^7, p^8, 2 * p^9$ de l'endomorphisme de Frobenius. En utilisant l'arithmétique de la table 3.2, le coût total est de $i_{579} + 15735m_{579} + 119534s_{579}$ pour ce travail.

3.6 Comparaison générale et conclusion

La table 3.3 de cette section récapitule les différents coûts obtenus dans notre travail et les compare aux résultats fournis par la littérature. Dans cette table, le gain théorique représente le nombre d'opérations que nous

avons supprimé en suivant notre approche. En supposant que les coûts d'un

Courbes	Références	Boucle de Miller	Exponentiation Finale
$k = 9$ 128-bits niv. sécurité	[LT12]	*coût spécifique non reporté*	$I_1 + 13081M_1$
	Ce travail	$6948M_1$	$I_1 + 12037M_1$
	Gain théorique	RAS	$\mathbf{1044M_1}$
$k = 15$ 192-bits niv. sécurité	[LT12]	*coût spécifique non reporté*	$I_1 + 36074M_1$
	Ce travail	$10986M_1$	$I_1 + 27137M_1$
	Gain théorique	RAS	$\mathbf{8937M_1}$
$k = 27$ 256-bits niv. sécurité	[ZL12]	$33I_1 + 21297M_1$	$I_1 + 139625M_1$
	Ce travail	$32I_1 + 20824M_1$	$I_1 + 135269M_1$
	Gain théorique	$\boldsymbol{I_1 + 473M_1}$	$\mathbf{4356M_1}$

TABLE 3.3 – Comparaison de coûts de la boucle de Miller et l'exponentiation finale.

carré et celui d'une multiplication sont égaux, on obtient, lorsque $k = 15$, que la valeur du coût de l'exponentiation finale est de $I_1 + 27137M_1$ pour notre travail et $I_1 + 36074M_1$ dans le cadre du travail présenté par [LT12]. L'amélioration théorique obtenue dans ce travail est donc de plus de 25%. Une analyse analogue pour $k = 9$ et $k = 27$ conduisent à une amélioration de près de 8% et 3% respectivement.

Dans ce chapitre, des détails et d'importantes améliorations dans le calcul de la boucle de Miller et l'exponentiation finale du couplage optimal Ate sur les courbes elliptiques admettant une tordue de degré 3 et de degré de plongement impair sont donc présentés. Une évaluation explicite est également donnée pour la boucle de Miller dans le cas des courbes elliptiques de degré de plongement 9 et 15.

Chapitre 4

Un couplage plus efficace issu du couplage β-Weil

Sommaire

4.1 Présentation

De nombreux travaux proposés dans la littérature sont focalisés sur l'amélioration de l'efficacité du calcul des couplages. C'est dans ce contexte d'étude des performances du calcul des couplages qu'en 2005 Koblitz et Menezes [KM05] ont évalué l'efficacité du couplage de Weil par rapport à celui de Tate. Il s'en est dégagé l'idée selon laquelle à des hauts niveaux de sécurité, le calcul du couplage de Weil pouvait être plus rapide que celui du couplage de Tate. Mais, en 2006, Granger et al [GPS06] ont réexaminé les techniques d'implémentation des couplages sur les courbes elliptiques ordinaires à divers niveaux de sécurité. Ils ont abouti, contrairement au précédent résultat, à la conclusion que le couplage de Tate était plus efficace

que le couplage de Weil à tous niveaux de sécurité. Et il s'ensuit une série de travaux [BGDM+10a, GF16, LMN10, Ver10, ZWL13] prouvant que le couplage de Tate tout comme ses variantes (optimal ate, twisted ate...) est plus efficace que le couplage de Weil. En 2012, un regain d'espoir est observé par l'introduction d'un couplage optimal du type Weil convenable pour des exécutions en parallèle appelé couplage β-Weil [AFCK+12]. Aranha et al. ont prouvé que ce couplage β-Weil peut être plus efficace que les couplages de type Tate.

Par ailleurs, dans de nombreux protocoles cryptographiques, l'évaluation du produit de m couplages ($m \geq 1$, un entier) est requis plutôt que celui d'un seul [ACD+06, BB04, CCS06, Wat05]. Pour une implémentation efficace de ces produits de couplage, Scott [Sco11] et Granger et al. [GS06] ont proposé séparemment une méthode efficace pour leur calcul. Cette méthode, habituellement appelée technique du «multi-pairing» en anglais, nécessite uniquement une seule élévation au carré par doublement au lieu de m élévations au carré requises de manière naive dans l'extension de corps. Elle peut également être utilisée pour calculer un seul couplage défini comme le produit de fonctions rationnelles ayant la même boucle de Miller. Les travaux appliquant cette technique pour calculer un seul couplage sont ceux de Sakemi et *al.* ainsi que Zhang et *al.* Dans [STNM10], Sakemi et al. appliquent la technique du multi-pairing pour améliorer le calcul du couplage Twisted Ate sur la famille des courbes elliptiques de Barreto-Naehrig tandis que Zhang et al. suggèrent dans [ZWL13] un nouveau couplage sur les courbes elliptiques bien couplées définies sur une extension de corps en supposant l'existence de telles courbes, sur lesquelles la technique du multi-pairing est appropriée pour une implémentation efficace.

Dans ce chapitre, nous portons une attention particulière sur le couplage β-Weil car très peu étudié dans la littérature. C'est ainsi que nous proposons à la section 4.2 une extension de ce couplage, initialement introduit sur les courbes elliptiques ordinaires de degré de plongement pair, sur des courbes elliptiques ordinaires de degré de plongement quelconque. Un exemple d'application du couplage β-Weil étendu aux courbes elliptiques bien couplées de degré de plongement $k = 27$ est donné ainsi qu'une preuve que le calcul de ce couplage sur de telles courbes est plus efficace que le calcul du couplage optimal Ate sur ces dernières. Dans la section 4.3, à partir du couplage β-Weil étendu, nous proposons un nouveau couplage optimal sur les courbes elliptiques ordinaires admettant un polynôme de Vercauteren h ayant pour expression $h(z) = x - z \in \mathbb{Z}[z]$. Ce nouveau couplage est défini comme produit de fonctions rationnelles avec la même boucle de Miller. Ce qui nous permet d'utiliser la technique du multi-pairing pour le calculer. Et son calcul nécessite une simple exponentiation finale en ce sens qu'elle s'évalue en terme de coût des endomorphismes de Frobenius. Enfin, un exemple d'application de ce nouveau couplage sur ces courbes BLS12 et sur la famille de courbes el-

liptiques bien couplées de degré de plongement $k = 15$ est donnée et révélant que ce nouveau couplage peut être plus efficace que le couplage optimal Ate sur ces familles de courbes sus-citées. Notons que les calculs sont effectués pour de hauts niveaux de sécurité suivant les recommandations du NIST [DPP+15] en ce qui concerne la cryptographie à base de couplage.

Dans toute la suite de ce manuscrit, les fonctions de Miller sont considérées normalisées.

4.2 Le couplage β-Weil étendu

Dans cette section, nous rappelons la définition du couplage β-Weil et l'étendons aux courbes elliptiques de degré de plongement quelconque.
On considère E une courbe elliptique ordinaire sur $\mathbb{F}_p$ admettant une tordue de degré d, un degré de plongement k, $e = k/pgcd(k,d)$, r un grand diviseur premier de l'ordre de $E(\mathbb{F}_p)$ et l un diviseur propre de k. Pour chaque $a \in \mathbb{Z}$ et $S \in E[r]$, soit $f_{a,S}$ la $\mathbb{F}_{p^k}$-fonction rationnelle normalisée avec pour diviseur $Div(f_{a,S}) = a(S) - ([a]S) - (a-1)(\mathcal{O})$. soit $h(z) = \sum_{i=0}^{c} h_i z^i \in \mathbb{Z}[z]$ un polynôme tel que $h(a) \equiv 0 (mod\ \ r)$. On désigne par $f_{a,h,S}$ la fonction de Miller étendue qui est une fonction rationnelle normalisée avec pour diviseur $\sum_{i=0}^{c} h_i((a^i S) - (\mathcal{O}))$. En désignant par $\pi_p : E \to E : (x,y) \mapsto (x^p, y^p)$ l'endomorphisme de Frobenius, $\mathbb{G}_1 = E[r] \cap Ker(\pi_p - [1])$, $\mathbb{G}_2 = E[r] \cap Ker(\pi_p - [p])$ et $\mathbb{G}_3$ désigne un sous groupe d'ordre r de $\mathbb{F}_{p^k}^*$.

4.2.1 Le couplage β-Weil et le couplage β-Weil étendu

Théorème 8. *([AFCK+12], Theorem 3)*
Il existe un polynôme de Vercauteren $h(z) = \Sigma h_i z_i \in \mathbb{Z}[z]$ pour lequel l'application

$$\beta : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 \ : (P,Q) \mapsto \prod_{i=0}^{e-1} \left(\frac{f_{p,h,[p^i]P}(Q)}{f_{p,h,Q}([p^i]P)} \right)^{(p^{k/2}-1)p^{e-1-i}} \quad \textit{est un couplage.}$$

Définition 30. *Le couplage défini dans le théorème précédent s'appelle le couplage β-Weil.*

Ici, il est montré, dans le but d'étendre le couplage β-Weil, que pour tout diviseur propre de k, ce couplage peut se définir sur des courbes de degré de plongement quelconque. Nous commençons par donner quelques résultats utiles.

Lemme 5. *[Mil04]*
Pour tout $R \in E$ et pour tous entiers u, v, w et s, on a :

1. $f_{uv,R} = f_{v,R}^u . f_{u,[v]R} = f_{u,R}^v . f_{v,[u]R}$.

2. $f_{u,[v][w]R} = f_{u,[v]([w]R)}$.
3. *En particulier, si* $R \in E[r]$, *alors* $f^s_{r,R} = f_{sr,R}$.

Corollaire 1. *([ZZX08], Theorem 1)*
Soit $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ *et* l *un diviseur propre de* k *; alors l'application*

$$\tilde{b} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P,Q) \mapsto \left(\frac{f_{p^e,P}(Q)}{f_{p^e,Q}(P)}\right)^{p^l-1} \quad \textit{est un couplage.}$$

Lemme 6. *([GHO+07], Lemma 6)*
Soit $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, $\mathcal{O}$ *est l'élément neutre de* $E(\mathbb{F}_p)$ *et* $lc_{\mathcal{O}}(f_{p,Q})$ *est le coefficient dominant de* $f_{p,Q}$ *au point* $\mathcal{O}$. *Si* $lc_{\mathcal{O}}(f_{p,Q}) = 1$ *suivant une* $\mathbb{F}_p$*-uniformisante rationnelle* $u_{\mathcal{O}}$ *au point* $\mathcal{O}$ *alors* $f_{p,Q}(P)$ *est un couplage.*

Théorème 9. *([ZZH08], Theorem 1)*
Soient $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ *et* 1_{μ_r} *l'élément neutre du groupe* $\mathbb{G}_3$.
Si $I : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ *est un couplage tel que* $I(P,Q) = 1_{\mathbb{G}_3}$ $\forall P \in \mathbb{G}_1$ *et* $\forall Q \in \mathbb{G}_2$, *alors l'ensemble de tous les couplages définis de* $\mathbb{G}_1 \times \mathbb{G}_2$ *vers* $\mathbb{G}_3$ *est un groupe multiplicatif ayant pour élément neutre* I.

Pour $1 \le \lambda \le k-1$ et $s_\lambda = p^\lambda \pmod r$, on définit les fonctions θ_{s_λ} et $\theta_{s_\lambda,h}$ par :

$$\theta_{s_\lambda} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P,Q) \mapsto \left(\frac{f_{s_\lambda,P}(Q)}{f_{s_\lambda,Q}(P)}\right)^{p^l-1}$$

et

$$\theta_{s_\lambda,h} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P,Q) \mapsto \left(\frac{f_{s_\lambda,h,P}(Q)}{f_{s_\lambda,h,Q}(P)}\right)^{p^l-1}.$$

Lemme 7. *Pour* $1 \le \lambda \le k-1$, *l'application*

$$\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P,Q) \mapsto \left(\prod_{i=0}^{e-1}\left(\frac{f_{s_\lambda,[p^i]P}(Q)}{f_{s_\lambda,[p^i]Q}(P)}\right)^{p^{e-1-i}}\right)^{p^l-1} \quad \textit{est un couplage. De plus, on a l'égalité suivante :} \left(\prod_{i=0}^{e-1}\left(\frac{f_{s_\lambda,[p^i]P}(Q)}{f_{s_\lambda,[p^i]Q}(P)}\right)^{p^{e-1-i}}\right)^{p^l-1} = \prod_{i=0}^{e-1}\theta_{s_\lambda}([p^i]P,Q)^{p^{e-1-i}}.$$

Preuve

D'après le résultat du Corollaire 1, l'application $(P,Q) \mapsto \left(\frac{f_{p^e,P}(Q)}{f_{p^e,Q}(P)}\right)^{p^l-1}$ est un couplage. Soient a, b des entiers et $R \in E$; en vertu du Lemme 5, on a : $f_{ab,R} = f^a_{b,R} f_{a,bR}$; d'où

$$f_{p^e,P} = \prod_{i=0}^{e-1}\left(f_{p,[p^iP]}\right)^{p^{e-1-i}}.$$

Puisque $s_\lambda = p^\lambda \pmod r$ et $f_{p^\lambda,P} = \prod_{j=0}^{\lambda-1} \left(f_{p,[p^j]P}\right)^{p^{\lambda-1-j}}$, on a :

$$\prod_{i=0}^{e-1} \left(f_{p^\lambda,[p^i]P}\right)^{p^{e-1-i}} = \prod_{i=0}^{e-1} \left(\prod_{j=0}^{\lambda-1} (f_{p,[p^i][p^j]P})^{p^{\lambda-1-j}} \right)^{p^{e-1-i}}$$
$$= \prod_{i=0}^{e-1} \left(\prod_{j=0}^{\lambda-1} (f_{p,[p^i][p^j]P})^{p^{\lambda-1-j}p^{e-1-i}} \right).$$

Posons $a_{ij} = (f_{p,[p^i][p^j]P})^{p^{\lambda-1-j}p^{e-1-i}}$;

$$\text{alors } \prod_{i=0}^{e-1} \left(f_{p^\lambda,[p^i]P}\right)^{p^{e-1-i}} = \prod_{i=0}^{e-1} \left(\prod_{j=0}^{\lambda-1} a_{ij} \right)$$
$$= \prod_{j=0}^{\lambda-1} \left(\prod_{i=0}^{e-1} a_{ij} \right)$$
$$= \prod_{j=0}^{\lambda-1} \left(\prod_{i=0}^{e-1} (f_{p,[p^i]([p^j]P)})^{p^{e-1-i}} \right)^{p^{\lambda-1-j}}$$
$$= \prod_{j=0}^{\lambda-1} (f_{p^e,[p^j]P})^{p^{\lambda-1-j}}.$$

D'où

$$\prod_{i=0}^{e-1} \left(f_{p^\lambda,[p^i]P}\right)^{p^{e-1-i}} = \prod_{j=0}^{\lambda-1} \left(f_{p^e,[p^j]P}\right)^{p^{\lambda-1-j}}. \tag{4.2.1}$$

Il découle de (4.2.1) que

$$\left(\prod_{i=0}^{e-1} \left(\frac{f_{p^\lambda,[p^i]P}(Q)}{f_{p^\lambda,[p^i]Q}(P)} \right)^{p^{e-1-i}} \right)^{p^l-1} = \left(\prod_{j=0}^{\lambda-1} \left(\frac{f_{p^e,[p^j]P}(Q)}{f_{p^e,[p^j]Q}(P)} \right)^{p^{\lambda-1-j}} \right)^{p^l-1}. \tag{4.2.2}$$

D'après le Lemme 6, l'application $(P,Q) \mapsto f_{p^e,Q}(P)$ est un couplage ; d'où

$$f_{p^e,[p^j]Q}(P) = f_{p^e,Q}([p^j]P). \tag{4.2.3}$$

Alors

$$\left(\prod_{j=0}^{\lambda-1} \left(\frac{f_{p^e,[p^j]P}(Q)}{f_{p^e,[p^j]Q}(P)} \right)^{p^{\lambda-1-j}} \right)^{p^l-1} = \left(\prod_{j=0}^{\lambda-1} \left(\frac{f_{p^e,[p^j]P}(Q)}{f_{p^e,Q}([p^j]P)} \right)^{p^{\lambda-1-j}} \right)^{p^l-1}. \tag{4.2.4}$$

Puisque l'application $(P,Q) \mapsto \left(\frac{f_{p^e,P}(Q)}{f_{p^e,Q}(P)} \right)^{p^l-1}$ est un couplage, il vient de (4.2.2) et (4.2.4) que $\left(\prod_{i=0}^{e-1} \left(\frac{f_{p^\lambda,[p^i]P}(Q)}{f_{p^\lambda,[p^i]Q}(P)} \right)^{p^{e-1-i}} \right)^{p^l-1}$ est un produit de couplages. On obtient donc du Théorème 9 que $\left(\prod_{i=0}^{e-1} \left(\frac{f_{p^\lambda,[p^i]P}(Q)}{f_{p^\lambda,[p^i]Q}(P)} \right)^{p^{e-1-i}} \right)^{p^l-1}$ est

un couplage. De plus, d'après le Lemme 6, l'application $(P,Q) \mapsto f_{p^\lambda,Q}(P)$ est un couplage ; on a donc :

$$\left(\prod_{i=0}^{e-1}\left(\frac{f_{p^\lambda,[p^i]P}(Q)}{f_{p^\lambda,[p^i]Q}(P)}\right)^{p^{e-1-i}}\right)^{p^l-1} = \left(\prod_{i=0}^{e-1}\left(\frac{f_{p^\lambda,[p^i]P}(Q)}{f_{p^\lambda,Q}([p^i]P)}\right)^{p^{e-1-i}}\right)^{p^l-1};$$

c'est-à-dire que

$$\left(\prod_{i=0}^{e-1}\left(\frac{f_{p^\lambda,[p^i]P}(Q)}{f_{p^\lambda,Q}([p^i]P)}\right)^{p^{e-1-i}}\right)^{p^l-1} = \prod_{i=0}^{e-1}\theta_{p^\lambda}([p^i]P,Q)^{p^{e-1-i}}. \qquad \square$$

Lemme 8. *Soient $a \in \mathbb{Z}$, $R \in E[r]$ et un polynôme $h(z) = \sum_{j=0}^{n} h_j z^j \in \mathbb{Z}[z]$ tel que $h(a) = rs$, alors $f_{r,R}^s = f_{a,h,R} \cdot \prod_{j=0}^{n} f_{a^j,R}^{h_j}$.*

Preuve Il suffit de montrer que $Div(f_{r,R}^s) = Div(f_{a,h,R}) + Div(\prod_{j=0}^{n} f_{a^j,R}^{h_j})$.

D'une part, nous avons

$$\begin{aligned} Div(f_{r,R}^s) &= s \times Div(f_{r,R}) \\ &= s[r(R) - ([r]R) - (r-1)(\mathcal{O})] \\ &= s[r(R) - (\mathcal{O}) - (r-1)(\mathcal{O})] \\ &= sr(R) - sr(\mathcal{O}) \\ &= h(a)((R) - (\mathcal{O})) \end{aligned}$$

D'autre part, $Div(f_{a,h,R}) = \sum_{j=0}^{n} h_j(([a]^j R) - (\mathcal{O})) = \sum_{j=0}^{n} h_j(([a^j]R) - (\mathcal{O}))$ et

$$\begin{aligned} Div(\prod_{j=0}^{n} f_{a^j,R}^{h_j}) &= \sum_{j=0}^{n} Div(f_{a^j,R}^{h_j}) \\ &= \sum_{j=0}^{n} h_j \times Div(f_{a^j,R}) \\ &= \sum_{j=0}^{n} h_j(a^j(R) - ([a^j]R) - (a^j-1)(\mathcal{O})) \end{aligned}$$

Nous avons :

$$\begin{aligned} Div(\prod_{j=0}^{n} f_{a^j,R}^{h_j}) + Div(f_{a,h,R}) &= \sum_{j=0}^{n} h_j(a^j(R) - ([a^j]R) - (a^j-1)(\mathcal{O})) \\ &\quad + \sum_{j=0}^{n} h_j(([a^j]R) - (\mathcal{O})) \\ &= \sum_{j=0}^{n} h_j a^j(R) - h_j a^j(\mathcal{O}) \\ &= \sum_{j=0}^{n} h_j a^j((R) - (\mathcal{O})) \\ &= h(a)((R) - (\mathcal{O})) \end{aligned}$$

$\square$

Le principal résultat de cette section est résumé dans le théorème suivant.

Théorème 10. *(Couplage β-Weil étendu)*

Il existe un polynôme de Vercauteren $h(z) = \Sigma h_i z_i \in \mathbb{Z}[z]$ tel que l'application

$\beta_k : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P,Q) \mapsto \prod_{i=0}^{e-1} \theta_{p,h}([p^i]P,Q)^{p^{e-1-i}}$ *soit un couplage. Plus précisement, si pour tout $0 \le i \le e-1$ et pour tout $0 \le j \le n$, r ne divise ni sp^{e-1-i} ni h_j, alors l'application β_k est non dégénérée.*

Preuve Soit $h(z) = \sum_{j=0}^{n} h_j z^j$ un polynôme de Vercauteren tel que $h(p) = rs$. D'après le lemme 8, nous avons $f_{r,P}^s = f_{p,h,P}.\prod_{j=0}^{n} f_{p^j,P}^{h_j}$; par conséquent, nous aurons

$$\left(\frac{f_{r,P}(Q)}{f_{r,Q}(P)}\right)^s = \frac{f_{p,h,P}(Q)}{f_{p,h,Q}(P)}.\prod_{j=0}^{n}\left(\frac{f_{p^j,P}(Q)}{f_{p^j,Q}(P)}\right)^{h_j};$$

i.e.

$$\left(\frac{f_{r,P}(Q)}{f_{r,Q}(P)}\right)^{s(p^l-1)} = \left(\frac{f_{p,h,P}(Q)}{f_{p,h,Q}(P)}\right)^{p^l-1}.\left(\prod_{j=0}^{n}\left(\frac{f_{p^j,P}(Q)}{f_{p^j,Q}(P)}\right)^{h_j}\right)^{p^l-1}.$$

D'où $\theta_r(P,Q)^s = \theta_{p,h}(P,Q).\prod_{j=0}^{n}\theta_{p^j}(P,Q)^{h_j}$. Alors

$\beta_k(P,Q) = \prod_{i=0}^{e-1}\theta_{p,h}([p^i]P,Q)^{p^{e-1-i}}$ et

$$\prod_{i=0}^{e-1}\theta_{p,h}([p^i]P,Q)^{p^{e-1-i}} = \prod_{i=0}^{e-1}\left(\theta_r([p^i]P,Q)^s.\prod_{j=0}^{n}\theta_{p^j}([p^i]P,Q)^{-h_j}\right)^{p^{e-1-i}};$$

i.e. $\beta_k(P,Q) = \prod_{i=0}^{e-1}\theta_r([p^i]P,Q)^{sp^{e-1-i}}.\prod_{j=0}^{n}\left(\prod_{i=0}^{e-1}\theta_{p^j}([p^i])P,Q)^{p^{e-1-i}}\right)^{-h_j}$.

D'où d'après le Lemme 7, l'application β_k est un produit de couplages. Ainsi, il découle du Théorème 9 que β_k est un couplage. D'autre part, supposons que pour tous $0 \le i \le e-1$ et $0 \le j \le n$, r ne divise ni sp^{e-1-i} ni h_j ; en vertu du Lemme 7, $\prod_{i=0}^{e-1}\theta_{p^j}([p^i])P,Q)^{p^{e-1-i}}$ est un couplage ; puisque l'application $\theta_r([p^i]P,Q)$ est une puissance du couplage de Weil, il vient que β_k est non dégénérée. □

Lemme 9. *Soit $R \in E[r]$ et un polynôme $h(z) = x - z \in \mathbb{Z}[z]$ tel que $h(p) \equiv 0 \pmod r$. On a l'égalité : $f_{p,x-p,R} = f_{x,R}$.*

Preuve

Il suffit de montrer que $Div(f_{p,x-p,R}) = Div(f_{x,R})$.

$$\begin{aligned} Div(f_{p,x-p,R}) &= \sum_{i=0}^{1} h_i[([p^i]R) - (\mathcal{O})] \\ &= h_0[(R) - (\mathcal{O})] + h_1[([p]R) - (\mathcal{O})] \\ &= x[(R) - (\mathcal{O})] - [([p]R - (\mathcal{O}))] \\ &= x(R) - x(\mathcal{O}) - ([p]R) + (\mathcal{O}) \\ &= x(R) - ([p]R) - (x-1)(\mathcal{O}) \end{aligned}$$

En plus, on a $[x]R = [p]R$ car $[x-p]R = \mathcal{O}$. D'où

$$Div(f_{p,x-p,R}) = x(R) - ([x]R) - (x-1)(\mathcal{O}) = Div(f_{x,R}).$$

□

Remarque 9. *Dans cette remarque, il est mis en exergue le fait que l'application du Théorème 10 sur les courbes elliptiques avec un degré de plongement pair coincide avec la définition du couplage β-Weil proposé par Aranha et* al. *dans [AFCK+12]. Considérons par exemple les courbes Barreto-Lynn-Scott avec degré de plongement 24 (BLS 24) [BLS02]. Cette famille de courbes elliptiques est définie dans $\mathbb{F}_p$ par les polynômes :*

$$\begin{aligned} p(x) &= (x-1)^2(x^8 - x^4 + 1)/3 + x \\ r(x) &= x^8 - x^4 + 1 \\ t(x) &= x + 1, \end{aligned} \tag{4.2.5}$$

cette famille de courbes admet $\rho = 1.25$, une tordue de degré 6 et le polynôme de Vercauteren $h(z) = x - z \in \mathbb{Z}[z]$ d'après la table 3 dans [AFCK+12]. Nous avons $k = 24$, $e = k/pgcd(k,d) = 4$ et 12 est un diviseur propre de k, on peut considéré pour cette famille de courbes, que $l = 12$. En vertu du Théorème 10, le couplage β-Weil étendu sur les courbes BLS 24 est l'application bilinéaire et non dégénérée β_{24} défini par :

$$\beta_{24} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P,Q) \mapsto \left[\prod_{i=0}^{3} \left(\frac{f_{x,[p^i]P}(Q)}{f_{x,Q}([p^i]P)} \right)^{p^{3-i}} \right]^{(p^{12}-1)}.$$

De plus, puisque $r \nmid p^4 + 1$, alors

$$\beta_{24}(P,Q)^{p^4+1} = \left[\prod_{i=0}^{3} \left(\frac{f_{x,[p^i]P}(Q)}{f_{x,Q}([p^i]P)} \right)^{p^{3-i}} \right]^{(p^{12}-1)(p^4+1)}$$

est aussi un couplage. Et en plus, $\beta_{24}(P,Q)^{p^4+1}$ est le couplage β-Weil défini sur les courbes BLS24 par Aranha et al. dans [AFCK+12].

4.2.2 Le couplage β-Weil étendu sur les courbes bien couplées avec $k = 27$

Dans cette section, nous appliquons le Théorème 10 à la famille de courbes elliptiques bien couplées de degré de plongement 27. Cette famille de courbes elliptiques admet une tordue de degré 3 sur $\mathbb{F}_p$, i.e. $d = 3$. Nous avons $k = 27$, $e = k/pgcd(k,d) = 9$ et les seuls diviseurs propres de k sont 3 et 9. Par ailleurs les coûts de chacune des puissances p^3 et p^9 de l'endomorphisme de Frobenius ont la même valeur, d'après la table 3.2. Par conséquent, l peut prendre l'une des valeurs 3 et 9. Sans nuire à la généralité, prenons $l = 3$. En appliquant le Théorème 10, on obtient le résultat suivant :

Proposition 5. *Le couplage de β-Weil pairing étendu sur les courbes elliptiques bien couplées avec $k = 27$ est une application bilinéaire et non dégénérée :*

$$\beta_{27} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P, Q) \mapsto \left[\prod_{i=0}^{8} \left(\frac{f_{x,[p^i]P}(Q)}{f_{x,Q}([p^i]P)} \right)^{p^{8-i}} \right]^{(p^3-1)}.$$

Pour les courbes elliptiques bien couplées avec $k = 27$, on rappelle que la meilleure valeur de x est $2^{29} + 2^{19} + 2^{17} + 2^{14}$. De cette valeur, $r(x)$ a un facteur premier de la taille de 514 bits et $p(x)$ un nombre premier de 579 d'après la section 3.5.2. En prenant en compte les récentes attaques du problème du logarithme discret et en basant notre estimation sur la complexité asymptotique de certaines versions améliorées du NFS, nous obtenons un niveau de sécurité d'environ 214 bits atteint pour ces paramètres [JP13, KB16, MSS16, SS16a, SS16b]. Rappelons que $f_{x,P}(Q)$ et $f_{x,Q}(P)$ sont appelées les fonctions Miller lite et full Miller respectivement, où $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$. Le calcul des fonctions Miller lite $f_{x,P}(Q)$ et full Miller $f_{x,Q}(P)$ se fait par l'exécution de l'algorithme 1 de la section 1.2.3. Plus précisément, pendant le calcul de $f_{x,P}(Q)$, l'arithmétique de la courbe elliptique est effectuée dans $\mathbb{F}_p$; et les équations de droites à coefficients dans $\mathbb{F}_p$ sont évaluées en des éléments de $\mathbb{F}_{p^k}$. Tandis que pour le calcul de $f_{x,Q}(P)$, c'est l'inverse. Les opérations sur la courbe elliptique sont faites dans $\mathbb{F}_{p^k}$ et l'évaluation des équations de droites à coefficients dans $\mathbb{F}_{p^k}$ sont effectuées en des éléments de $\mathbb{F}_p$. Ainsi, le calcul d'une fonction full Miller est plus couteux que celui d'une fonction Miller lite car toutes opérations sont effectuées sur le gros corps $\mathbb{F}_{p^k}$.

Dans ce cas, pour $k = 27$, la fonction de Miller est calculée en coordonnées affines puisque les formules affines procurent un calcul rapide du couplage [ZL12].

D'où les calculs de $f_{x,P}(Q)$ et de $f_{x,Q}(P)$ nécessitent 29 étapes de doublement, 3 étapes d'additions, 28 carrés dans $\mathbb{F}_{p^{27}}$ et 31 multiplications dans $\mathbb{F}_{p^{27}}$. Supposons que les points $[p^i]P$, pour $1 \leq i \leq 8$; sont précalculés. Considérons les notations additionnelles suivantes :
MLite := le coût de la boucle de Miller lite ;
FullM := le coût de la boucle du full Miller ;
PF := le coût de la phase finale ;
EF := le coût de l'exponentiation finale.

Calcul du couplage β_{27}

Le couplage β_{27} défini dans la proposition 5 possède 18 fonctions de Miller. Et ces dernières peuvent être calculées en parallèle en utilisant 6 processeurs qui calculent chacun 3 trois fonctions de Miller. Désignons par

f_1, f_2 et f_3 les fonctions calculées par les trois premiers processeurs, puis par g_1, g_2, g_3 les fonctions calculées par les trois derniers processeurs. Le chemin d'exécution du calcul du couplage β_{27} sur 6 processeurs est la suivante :

1. Le premier processeur calcule $f_1 = f_{x,P}^{p^8}(Q).f_{x,[p]P}^{p^7}(Q).f_{x,[p^2]P}^{p^6}(Q)$; ainsi il éxecute trois boucles de Miller lite, 1 p^8-Frobenius, 1 p^7-Frobenius, 1 p^6-Frobenius et 2 multiplications dans $\mathbb{F}_{p^{27}}$. i.e.
   $3 \times MLite + 2 \times 26m_{579} + 18m_{579} + 2M_{27} = 3 \times MLite + 502m_{579}$.
2. Le second processeur calcule $f_2 = f_{x,[p^3]P}^{p^5}(Q).f_{x,[p^4]P}^{p^4}(Q).f_{x,[p^5]P}^{p^3}(Q)$; ainsi il éxecute trois boucles de Miller lite, 1 p^5-Frobenius, 1 p^4-Frobenius, 1 p^3-Frobenius et 2 multiplications in $\mathbb{F}_{p^{27}}$. i.e.
   $3 \times MLite + 2 \times 26m_{579} + 18m_{579} + 2M_{27} = 3 \times MLite + 502m_{579}$.
3. Le troisième processeur calcule $f_3 = f_{x,[p^6]P}^{p^2}(Q).f_{x,p^7]P}^{p}(Q).f_{x,[p^8]P}(Q)$; ainsi il éxecute trois boucles de Miller lite, 1 p^2-Frobenius, 1 p-Frobenius et 2 multiplications dans $\mathbb{F}_{p^{27}}$. i.e.
   $3 \times MLite + 2 \times 26m_{579} + 2M_{27} = 3 \times MLite + 484m_{579}$.
4. Le 4$^{\text{ième}}$ processeur calcule $g_1 = f_{x,Q}^{p^8}(P).f_{x,Q}^{p^7}([p]P).f_{x,Q}^{p^6}([p^2]P)$; ainsi il éxecute trois boucles du full Miller, 1 p^8-Frobenius, 1 p^7-Frobenius, 1 p^6-Frobenius et 2 multiplications dans $\mathbb{F}_{p^{27}}$. i.e.
   $3 \times FullM + 2 \times 26m_{579} + 18m_{579} + 2M_{27} = 3 \times FullM + 502m_{579}$.
5. Le 5$^{\text{ième}}$ processeur calcule $g_2 = f_{x,Q}^{p^5}([p^3]P).f_{x,Q}^{p^4}([p^4]P).f_{x,Q}^{p^3}([p^5]P)$; ainsi il éxecute trois boucles du full Miller, 1 p^5-Frobenius, 1 p^4-Frobenius, 1 p^3-Frobenius et 2 multiplications dans $\mathbb{F}_{p^{27}}$. i.e.
   $3 \times FullM + 2 \times 26m_{579} + 18m_{579} + 2M_{27} = 3 \times FullM + 502m_{579}$.
6. Le 6$^{\text{ième}}$ processeur calcule $g_3 = f_{x,Q}^{p^2}([p^6]P).f_{x,Q}^{p}([p^7]P).f_{x,Q}([p^8]P)$; ainsi il éxecute trois boucles du full Miller,, 1 p^2-Frobenius, 1 p-Frobenius et 2 multiplications dans $\mathbb{F}_{p^{27}}$. i.e.
   $3 \times FullM + 2 \times 26m_{579} + 2M_{27} = 3 \times FullM + 484m_{579}$.

Le calcul du couplage β_k nécessite donc trois étapes qui sont : le calcul des 18 fonctions de Miller en parallèle sur 6 processeurs, la phase finale et l'exponentiation finale. Le calcul des 18 fonctions de Miller est bien explicité par le chemin d'exécution détaillé ci-dessus. La phase finale est la phase correspondante au calcul de $(f_1 \times f_2 \times f_3) \times (g_1 \times g_2 \times g_3)^{-1}$. L'exponentiation finale consiste en élévation à la puissance $p^3 - 1$. En utilisant les coûts de l'arithmétique donnés dans la table 3.2, l'exponentiation finale par $p^3 - 1$ nécessite : 1 p^3-Frobenius, 1 multiplication dans $\mathbb{F}_{p^{27}}$ et 1 inversion dans $\mathbb{F}_{p^{27}}$ i.e. $18m_{579} + 1M_{27} + 1I_{27} = 18m_{579} + 216m_{579} + 1i_{579} + 387m_{579} + 86s_{579} = 1i_{579} + 621m_{579} + 86s_{579}$. La phase finale revient à calculer $(f_1 \times f_2 \times f_3) \times (g_1 \times g_2 \times g_3)^{-1}$, soit 1 inversion et 5 multiplications dans $\mathbb{F}_{p^{27}}$ i.e. $1i_{579} + 1467m_{579} + 86s_{579}$. Puisque le calcul d'une fonction Miller lite est moins coûteux que celui de le fonction full Miller, le coût du calcul de la fonction Miller lite est ignoré dans le reste de cette section. La table 4.1 résume le coût de la fonction full

Miller $f_{x,Q}$ obtenu dans la section 3.5.2 en coordonnées affines, le coût de la phase finale et le coût de l'exponentiation finale.

Etape	Coût
FullM	$32i_{579} + 12240m_{579} + 8584s_{579}$
Phase Finale (PF)	$1i_{579} + 1467m_{579} + 86s_{579}$
Exponentiation Finale (EF)	$1i_{579} + 621m_{579} + 86s_{579}$

TABLE 4.1 – Coût de la boucle du Full Miller en coordonnées affines, la phase finale et l'exponentiation finale

4.2.3 Comparaison

Pour établir une comparaison entre le coût du couplage optimal Ate et celui du couplage β-Weil étendu sur les courbes elliptiques de degré de plongement 27, nous effectuons une analyse qui ne se focalise que sur le coût du couplage optimal Ate et celui de g_1 auquel on ajoute les coûts de la phase finale et de l'exponentiation finale par (p^3-1). Car le calcul de g_1 est le plus couteux et les autres calculs se faisant en parallèle. Supposons que le coût d'un carré est le même que celui d'une multiplication ($M_1 = S_1$) et que $I_1 = 10M_1$. Désignons par G_1 le coût de g_1. Nous avons ainsi les coûts suivants : $G_1 + PF + EF = 98i_{579} + 65234m_{579}$, soit $66214M_1$ pour le couplage β-Weil étendu et $33i_{579}+156093m_{579}$, soit $156423M_1$ pour le couplage optimal ate (cf. table 3.3).

Système de Coordonnées	G_1+PF+EF	Coût du couplage optimal Ate
Affine	$66214M_1$	$156423M_1$ (cf. table 3.3)

TABLE 4.2 – Comparaison des coûts du couplage β-Weil étendu et le couplage optimal Ate en coordonnées affines

La table 4.2 donne les coûts du couplage β-Weil étendu obtenu dans ce travail sur les courbes elliptiques avec $k = 27$ et du couplage optimal Ate sur la même courbe elliptique proposée dans la section 3.5.2. De cette table, il s'en dégage que le calcul du couplage β-Weil étendu supprime 90209 multiplications dans $\mathbb{F}_p$, soit 57,67 % de multiplications dans $\mathbb{F}_p$.

4.3 Un nouveau couplage optimal

4.3.1 Définition du nouveau couplage optimal $\hat{\beta}_k$

En utilisant le fait que le produit de deux couplages est un couplage, on observe que lorsque le polynôme de Vercauteren h a pour expression $h(z) = x - z$, nous pouvons définir un autre couplage plus efficace que le couplage β_k fourni par le Théorème 10. D'où le résultat suivant :

Théorème 11. *Pour toute famille de courbes elliptiques bien couplées parametrées par $p(x), r(x), t(x)$ où le polynôme de Vercauteren h a pour expression $h(z) = x - z$ tel que $h(p) \equiv 0 (mod\ r)$, l'application*
$\tilde{\beta}_k : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P, Q) \mapsto \left(\prod_{i=0}^{e-1} f_{x,[p^i]P}(Q)^{p^{e-1-i}} \right)^{p^l - 1}$ *est bilinéaire. De plus, si $r \nmid (p^l - 1)p^{e-1}$, alors $\tilde{\beta}_k$ est non-dégénérée.*

Preuve D'après le lemme 9, la fonction Miller étendue $f_{p,h,R} = f_{x,R}$ car $h(z) = x - z$ et $h(p) \equiv 0\ \ (mod\ r)$. Dans ce cas, on a
$\beta_k(P, Q) = \prod_{i=0}^{e-1} \left(\frac{f_{x,[p^i]P}(Q)}{f_{x,Q}([p^i]P)} \right)^{(p^l-1)p^{e-1-i}}$. Comme β_k et l'application qui à (P, Q) associe $f_{x,Q}(P)$ sont des couplages, d'après le Théorème 9, le Théorème 10 et le Lemme 6, nous avons :

$$\prod_{i=0}^{e-1} f_{x,[p^i]P}(Q)^{(p^l-1)p^{e-1-i}} = \beta_k(P, Q).\prod_{i=0}^{e-1} f_{x,Q}([p^i]P)^{(p^l-1)p^{e-1-i}}$$

est un couplage. En outre,
$\prod_{i=0}^{e-1} f_{x,[p^i]P}(Q)^{(p^l-1)p^{e-1-i}} = \left(\prod_{i=0}^{e-1} f_{x,[p^i]P}(Q)^{p^{e-1-i}} \right)^{p^l-1} = \tilde{\beta}_k(P, Q)$. Par ailleurs, $\tilde{\beta}_k$ est non-dégénérée si $r \nmid (p^l - 1)p^{e-1}$ car l'application β_k et $f_{x,Q}([p^i]P)$ sont non dégénérées. □

Lemme 10. *([ZWL13], Theorem 3) Soit $P \in \mathbb{G}_1$ et $Q \in \mathbb{G}_2$, alors*
$f_{p,[p^i]P}(Q) = f_{p,\hat{\pi}_{p^i}(P)}(\pi_{p^{k-i}}(Q))^{p^i}$, où $\hat{\pi}_{p^i}$ est le dual de π_{p^i}.

Le couplage défini dans le Théorème 11 peut être transformé, à l'aide du lemme 10, en un produit de fonctions rationnelles avec la même boucle de Miller.

Théorème 12. *Pour toute famille de courbes elliptiques bien couplées parametrées par $p(x), r(x), t(x)$ où le polynôme de Vercauteren h a pour expression $h(z) = x - z$ tel que $h(p) \equiv 0 (mod\ r)$, l'application*
$\hat{\beta}_k : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P, Q) \mapsto \left(\prod_{i=0}^{e-1} f_{x,\hat{\pi}_{p^i}(P)}(\pi_{p^{k-i}}(Q)) \right)^{(p^l-1)p^{e-1}}$ *est un couplage.*

Remarque 10. *1. Puisque $\hat{\pi}_{p^i} \circ \pi_{p^i} = [p^i]$ sur E, il s'ensuit que $\hat{\pi}_{p^i} \circ \pi_{p^i}(P) = [p^i]P$, et comme $\pi_{p^i}(P) = P$ alors on a $\hat{\pi}_{p^i}(P) = [p^i]P$.*

2. Notons qu'on peut utiliser la technique du multi-pairing (voir l'algorithme 3) pour calculer notre nouveau couplage $\hat{\beta}_k$ puisqu'il est défini comme le produit de fonctions rationnelles avec la même boucle de Miller.

Algorithme 3 : algorithme de Miller pour le Multi-pairing

Entrée : n$=\sum_{j=0}^{L} n_j 2^j \in \mathbb{N}$, $n_j \in \{-1, 0, 1\}$, $\{P_1, P_2, ..., P_{e-1}\}$, $\{Q_1, Q_2, ..., Q_{e-1}\}$
Sortie : $\prod_{i=0}^{e-1} f_{n,P_i}(Q_i)$, $\{[n]P_0, [n]P_2, ..., [n]P_{e-1}\}$
$f \leftarrow 1$
pour i ***allant de*** $e-1$ ***à*** 0 **faire**
  $T_i \leftarrow P_i$
fin
pour j ***allant de*** $L-1$ ***à*** 0 **faire**
  $f \leftarrow f^2$
  pour i ***allant de*** $e-1$ ***à*** 0 **faire**
    $f \leftarrow f.l_{T_i,T_i(Q_i)}/v_{[2]T_i}(Q_i)$; $T_i \leftarrow [2]T_i$
  fin
  si $n_j = 1$ **alors**
    pour i ***allant de*** $e-1$ ***à*** 0 **faire**
      $f \leftarrow f.l_{T_i,P_i}(Q_i)/v_{T_i+P_i}(Q_i)$; $T_i \leftarrow T_i + P_i$
    fin
  fin
  si $n_j = -1$ **alors**
    pour i ***allant de*** $e-1$ ***à*** 0 **faire**
      $f \leftarrow f.l_{T_i,-P_i}(Q_i)/v_{T_i-P_i}(Q_i)$; $T_i \leftarrow T_i - P_i$
    fin
  fin
fin
renvoie f.

Dans l'algorithme 3, $l_{R,S}$ est la droite passant par les points R et S et v_{R+S} est la verticale passant par $R+S$ où R et S sont deux points quelconques de la courbe elliptique.

4.3.2 Calcul du nouveau couplage optimal $\hat{\beta}_k$

Posons $P_i = [p^i]P$ et $Q_i = \pi_{p^{k-i}}(Q)$, pour $0 \leq i \leq e-1$. Supposons que les points P_i et Q_i, pour $1 \leq i \leq e-1$ sont précalculés. Le calcul de $\hat{\beta}_k$ nécessite deux étapes principales : le produit de e fonctions de Miller et la simple exponentiation finale. Dans ce cas, la boucle de Miller consiste à calculer $f_{x,P_i}(Q_i)$. Le nouveau couplage $\hat{\beta}_k$ étant le produit de plusieurs fonctions rationnelles avec la même boucle de Miller, il va de soi que la technique du multi-pairing peut être utilisée pour calculer $\hat{\beta}_k$. Ainsi pour évaluer le coût du calcul de $\hat{\beta}_k$, il faut tout d'abord calculer :

C_1 : le coût de l'élévation au carrée dans la boucle de Miller ;

C_2 : le coût des autres opérations dans la boucle de Miller (opérations sur les points et les évaluations des équations de droite) ;

C_3 : le coût de la simple exponentiation finale par $(p^l - 1)p^{e-1}$.

Le coût total de $\hat{\beta}_k$ est alors la somme de C_1, eC_2 et C_3.

4.3.3 Application du couplage $\hat{\beta}_k$ sur des familles de courbes

Dans cette partie, le Théorème 12 est appliqué sur les familles de courbes elliptiques bien couplées de degrés de plongement 15 et 12 respectivement.

Calcul du couplage $\hat{\beta}_k$ sur les courbes elliptiques bien couplées avec $k = 15$

Cette famille de courbes elliptiques est présentée dans la section 3.4. L'approche décrite dans [Ver10] permet d'obtenir la fonction optimale $h(z) = x - z \in \mathbb{Z}[z]$. Cette famille de courbes elliptiques admet une tordue de degré 3, i.e. $d = 3$; d'où $e = k/pgcd(k,d) = 5$. Les seuls diviseurs propres de k étant 3 et 5, nous prenons $l = 5$, puisqu'en vertu la table 3.2, p^5-Frobenius est moins couteux que p^3-Frobenius. L'application du Théorème 12 à cette famille de courbes conduit à la définition suivante :

Définition 31. *Le couplage $\hat{\beta}_k$ sur les courbes elliptiques bien couplées avec $k = 15$ est l'application bilinéaire et non dégénérée :*

$$\hat{\beta}_{15} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P,Q) \mapsto \left(\prod_{i=0}^{4} f_{x,P_i}(Q_i) \right)^{(p^5-1)p^4}.$$

Pour cette famille de courbes elliptiques, la valeur $x = 2^{48} + 2^{41} + 2^9 + 2^8 + 1$ a été trouvée. De cette valeur, on obtient que $r(x)$ est un nombre premier de 385 bits et $p(x)$ un nombre premier de 575 bits d'après la section 3.4.2. En prenant en compte les récentes attaques du problème du logarithme

discret et en basant notre estimation sur la complexité asymptotique de certaines versions améliorées du NFS, nous obtenons un niveau de sécurité d'environ 168 bits atteint pour ces paramètres [JP13, KB16, MSS16, SS16a, SS16b]. Dans ce cas, la boucle de Miller consiste à calculer $f_{x,P_i}(Q_i)$. D'où, en comptant les évaluations des équations de droites associées, la boucle de Miller exécute 48 doublement de points, 4 additions de points, 47 carrés dans $\mathbb{F}_{p^{15}}$ et 51 multiplications dans $\mathbb{F}_{p^{15}}$. Suivant les explications données à la section 4.3.2, pour évaluer le coût du calcul de $\hat{\beta}_{15}$, il faut tout d'abord calculer :

C_1 : le coût de l'élévation au carré dans la boucle de Miller ;

C_2 : le coût des autres opérations dans la boucle de Miller (opérations sur les points et les évaluations de droite) ;

C_3 : le coût de la simple exponentiation finale par $(p^5-1)p^4$.

Le coût total de $\hat{\beta}_{15}$ est alors la somme de C_1, $5C_2$ et C_3. En utilisant les coûts des opérations arithmétiques de la table 4.3 et suivant les coûts des étapes de doublement et d'addition figurant dans la table 4.3 en coordonnées projectives, le coût de la boucle de Miller vaut $48(9s_{575}+37m_{575})+4(5s_{575}+53m_{575})+47S_{15}+51M_{15} = 4283m_{575}+2567s_{575} = 6850m_{575}$ et le coût de l'élévation au carré dans la boucle de Miller C_1 a pour valeur $47S_{15} = 47\times45s_{575} = 2115s_{575}$; d'où $C_2 = 4735m_{575}$. Enfin, l'exponentiation finale nécessite 1 p^5-Frobenius, 1 p^4-Frobenius, 1 inversion dans $\mathbb{F}_{p^{15}}$ et 1 multiplication dans $\mathbb{F}_{p^{15}}$. Donc $C_3 = 10m_{575}+14m_{575}+1i_{575}+149m_{575}+45m_{575} = 1i_{575}+218m_{575}$. Le coût de $\hat{\beta}_{15}$ est $C_1+5C_2+C_3 = 2115m_{575}+5\times4735m_{575}+218m_{575}+i_{575} = i_{575}+26008m_{575}$.

Etape	Coût en coordonnées projectives
Dbl point	$1m_{3b}+5s_{575}+2m_{575}$
Evaluation de la fonction de Miller	$4s_{575}+35m_{575}$
Doublement de Miller	$1m_{3b}+9s_{575}+37m_{575}$
Add point	$2s_{575}+13m_{575}$
Evaluation de la fonction de Miller	$3s_{575}+40m_{575}$
Add Miller	$5s_{575}+53m_{575}$

TABLE 4.3 – Coût des étapes de doublement et d'addition dans la boucle de Miller Lite en coordonnées projectives.

Calcul du couplage $\hat{\beta}_k$ sur les courbes elliptiques bien couplées $k = 12$ avec (BLS12)

En 2002, Barreto, Lynn et Scott ont proposé dans [BLS02] une méthode de génération des courbes elliptiques ordinaires bien couplées sur un corps premier $\mathbb{F}_p$ avec un degré de plongement $k = 12$. BLS12 sont définies sur $\mathbb{F}_p$ par les polynômes suivants :

$$\begin{array}{lcl} p(x) & = & (x-1)^2(x^4-x^2+1)/3+x; \\ r(x) & = & x^4-x^2+1; \\ t(x) & = & x+1. \end{array} \tag{4.3.1}$$

Dans ce cas, le polynôme Vercauteren h a pour expression $h(z) = x - z \in \mathbb{Z}[z]$. Cette famille de courbes elliptiques admet une tordue de degré 6 ; i.e. $d = 6$; d'où $e = k/pgcd(k, d) = 2$. Les seuls diviseurs propres de k étant 2, 3, 4 et 6, nous prenons $l = 6$ puisque l'élévation à la puissance p^6 est équivalent à une conjugaison [BGDM+10b]. L'application du Théorème 12 conduit au résultat suivant :

Proposition 6. *Le couplage $\hat{\beta}_k$ sur les courbes elliptiques bien couplées avec $k = 12$ est l'application bilinéaire et non dégénérée :*

$$\hat{\beta}_{12} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 : (P, Q) \mapsto \left(\prod_{i=0}^{1} f_{x,P_i}(Q_i) \right)^{p(p^6-1)}.$$

D'après [GF16, Proposition 6], pour $x = -2^{107} + 2^{84} + 2^{19}$, $p(x)$ est un nombre premier de 641 bits et $r(x)$ un nombre premier de 428 bits. En prenant en compte les récentes attaques du problème du logarithme discret et en basant notre estimation sur la complexité asymptotique de certaines versions améliorées du NFS, nous obtenons un niveau de sécurité d'environ 160 bits atteint pour ces paramètres [JP13, KB16, MSS16, SS16a, SS16b]. Dans ce cas, la boucle de Miller consiste à calculer $f_{x,P_i}(Q_i)$. D'où ce qui nécessite, en comptant les évaluations des équations de droites associées, 107 doublements de points, 3 additions de points avec les évaluations de droites associées, 106 carrés dans $\mathbb{F}_{p^{12}}$ et 109 multiplications dans $\mathbb{F}_{p^{12}}$. Suivant les explications données à la section 4.3.2, pour évaluer le coût du calcul de $\hat{\beta}_{12}$, il faut tout d'abord calculer :

- C_1 : le coût de l'élévation au carrée dans la boucle de Miller ;
- C_2 : le coût des autres opérations dans la boucle de Miller (opérations sur les points et les évaluations de droite) ;
- C_3 : le coût de la simple exponentiation finale par $p(p^6 - 1)$.

Le coût total de $\hat{\beta}_{12}$ est alors la somme de C_1, $2C_2$ et C_3. Dans ce cas, pour $k = 12$, la fonction de Miller est calculée en coordonnées projectives,

puisqu'ici les formules projectives permettent un calcul rapide du couplage [AFCK+12]. Suivant les coûts des étapes de doublement et d'addition figurant dans la table 4.4 en coordonnées projectives et en utilisant la table 4 de [AFCK+12], nous obtenons comme coût de la boucle de Miller : $107(7s_{641} + 12m_{641})+3(2s_{641}+27m_{641})+106(36m_{641})+109(39m_{641}) = 9432m_{641}+755s_{641} = 10187m_{641}$ et comme coût de l'élévation au carré dans la boucle de Miller C_1 : $106S_{12} = 106 \times 36M_{641} = 3816m_{641}$; d'où $C_2 = 6371m_{641}$. Enfin, l'exponentiation finale nécessite 1 p-Frobenius, 1 multiplication dans $\mathbb{F}_{p^{12}}$ et 1 inversion dans $\mathbb{F}_{p^{12}}$ car l'élévation à la puissance p^6 est équivalent à une conjugaison [BGDM+10b]. D'où $C_3 = 10m_{641} + 1i_{641} + 95m_{641} + 54m_{641} = 1i_{641} + 159m_{641}$. Le coût total de $\hat{\beta}_{12}$ vaut
$C_1 + 2C_2 + C_3 = 3816m_{641} + 2 \times 6371m_{641} + 159m_{641} + i_{641} = 16717m_{641} + i_{641}$.

Etape	Coût en coordonnées projectives
Dbl point	$1m_{3b} + 5s_{641} + 2m_{641}$
Evaluation de la fonction de Miller	$10m_{641}$
Doublement de Miller	$1m_{3b} + 5s_{641} + 12m_{641}$
Add point	$2s_{641} + 13m_{641}$
Evaluation de la fonction de Miller	$14m_{641}$
Add Miller	$2s_{641} + 27m_{641}$

TABLE 4.4 – Coût des étapes de Doublement et d'addition dans la boucle de Miller Lite en coordonnées projectives.

4.3.4 Comparaison et conclusion

Pour faire la comparaison entre le nouveau couplage $\hat{\beta}_k$ et le couplage optimal Ate, tous deux appliqués aux courbes elliptiques de degré de plongement 15 et les courbes elliptiques BLS 12, un récapitulatif du coût total de leurs calculs respectifs est donné dans la table 4.5. Ici le gain théorique représente le nombre d'opérations supprimé par le nouveau couplage $\hat{\beta}_k$.
Le calcul du couplage $\hat{\beta}_k$ supprime $5i_{641}+2234m_{641}$ pour BLS12 et $12115m_{575}$ pour les courbes elliptiques bien couplées avec $k = 15$.

De cette étude, il s'en dégage que notre couplage $\hat{\beta}_k$ peut être plus efficace que le couplage optimal Ate.

Courbes	Références	Couplages	Coût total du couplage	Gain théorique
Courbes BLS 12	[GF16]	optimal Ate	$6i_{641} + 18951m_{641}$	$\mathbf{5i_{641} + 2234m_{641}}$
	cf. section 4.3.3	$\boldsymbol{\hat{\beta}_k}$	$\mathbf{i_{641} + 16717m_{641}}$	
Courbes avec k=15	cf. table 3.3	optimal Ate	$1i_{575} + 38123m_{575}$	$\mathbf{12115m_{575}}$
	cf. section 4.3.3	$\boldsymbol{\hat{\beta}_k}$	$\mathbf{1i_{575} + 26008m_{575}}$	

TABLE 4.5 – Tableau comparatif du coût du calcul des couplages $\hat{\beta}_k$ et optimal Ate sur BLS12 et courbes elliptiques bien couplées avec k=15.

Conclusion et Perspectives

Notre travail a porté sur la recherche des courbes elliptiques bien couplées et sur la construction des nouveaux couplages optimaux définis sur les dites courbes. Nous avons identifié une famille de courbes elliptiques bien couplées ordinaires sur lesquelles nous avons pu construire des couplages optimaux qui peuvent être plus efficaces que le couplage optimal Ate. Plus précisément, ladite famille de courbes est constituée des courbes elliptiques bien couplées ordinaires admettant un polynôme de Vercauteren h ayant pour expression $h(z) = x - z \in \mathbb{Z}[z]$ et tel que $h(p) \equiv 0(mod\ \ r)$. Nous avons aussi amélioré les coûts théoriques du calcul du couplage optimal Ate sur des courbes elliptiques de degré de plongement impair, notamment $k = 9, 15$ et 27. Une bonne sélection des paramètres aux niveaux de sécurité respectifs 128, 192 et 256-bits nous a permis d'améliorer les coûts théoriques pour les étapes de l'algorithme de Miller et de l'exponentiation finale en utilisant la méthode basée sur les réseaux comparativement aux résultats des travaux existants sur de telles familles de courbes.

C'est ainsi qu'après avoir rappelé les notions élémentaires pour la compréhension du calcul et de la construction des couplages, nous avons continué dans le deuxième chapitre à étudier la possibilité de générer des courbes elliptiques ordinaires bien couplées définies sur les extensions de corps optimales. De cette étude, nous avons proposé une approche de construction de ces courbes bien que nous rencontrons jusqu'à date de sérieux problèmes d'incompatibilité des relations impliquant les paramètres recherchés.

Dans le chapitre 3, nous avons apporté quelques améliorations sur les coûts théoriques du calcul du couplage optimal Ate sur des courbes elliptiques de degré de plongement impair, notamment $k = 9, 15$ et 27 dans [FMP16]. Plus précisément, en utilisant la méthode de Fuentes et *al.* [FKR11] basée sur les réseaux, nous améliorons les coûts théoriques pour l'étape de l'exponentiation finale du calcul du couplage optimal Ate sur des courbes elliptiques de degré de plongement $9, 15$ et 27 comparativement aux résultats des travaux existants sur de telles familles de courbes. Le gain théorique étant le nombre d'opérations supprimé suivant notre approche, nous obtenons les gains théoriques respectifs de $1044M_1$, $8937M_1$ et $4356M_1$ pour les courbes elliptiques de degré de plongement $9, 15$ et 27 ; M_1 étant le coût

d'une multiplication dans $\mathbb{F}_p$.

A partir de la divergence autour de l'efficacité du calcul des couplages de Weil et de Tate provenant des articles [KM05] et [GPS06], il y a eu une série d'articles [BGDM+10a, GF16, LMN10, Ver10, ZWL13] prouvant que le couplage de Tate, tout comme ses variantes (optimal Ate, twisted Ate...), est plus efficace que le couplage de Weil. Mais en 2012, un couplage optimal de type Weil, le couplage β-Weil, convenable pour des exécutions en parallèle est introduit par Aranha et *al.* dans [AFCK+12]. Ils ont prouvé que le couplage β-Weil peut être plus efficace que les couplages de type Tate. En revisitant le couplage de β-Weil, nous avons, dans un premier temps, proposé une extension de ce couplage sur des courbes elliptiques ordinaires de degré de plongement quelconque. Et dans un second temps, nous avons identifié une famille de courbes elliptiques ordinaires bien couplées sur lesquelles nous avons construit des couplages optimaux issus du couplage β-Weil étendu qui peuvent être plus efficace que le couplage optimal Ate sur ces courbes elliptiques. Ladite famille de courbes est constituée des courbes elliptiques ordinaires bien couplées admettant un polynôme de Vercauteren h ayant pour expression $h(z) = x - z \in \mathbb{Z}[z]$ et tel que $h(p) \equiv 0 \pmod{r}$, où p est la caractéristique du corps de base sur lequel est défini chacune des courbes et r un grand facteur premier de l'ordre du groupe des points rationnels de chaque courbe. Ainsi nous avons proposé une réponse au problème consistant à trouver des courbes elliptiques ordinaires et des couplages optimaux efficaces sur ces dernières pouvant garantir une implémentation efficace des cryptosystèmes.

Au vu des fulgurantes avancées obtenues dans la résolution du problème du logarithme discret dans les corps finis de petites et moyennes caractéristiques et sachant que les recherches se poursuivent pour pouvoir résoudre ce DLP en un temps quasi polynomial, l'abandon des protocoles cryptographiques à base de couplage pourrait arriver. Néanmoins, un regain d'espoir peut exister à savoir, qu'on peut proposer d'autres contournements à ce déclin en reposant la sécurité du couplage sur l'impossibilité calculatoire de la résolution des problèmes encore difficiles à l'heure actuelle tels que ceux de l'inversion de couplage et de Diffie-Hellman bilinéaire car à l'heure actuelle, on n'a pas encore de solutions à ces problèmes.

Bibliographie

[ACD+06] M. Abdalla, D. Catalano, A. W. Dent, J. Malone-Lee, G. Neven, and N.P. Smart. Identity-based encryption gone wild. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 300–311, 2006.

[AFCK+12] D.F. Aranha, L. Fuentes-Castañeda, E. Knapp, A. Menezes, and F. Rodriguez-Henriquez. Implementing pairings at the 192-bit security level. In *Pairing-Based Cryptography-Pairing 2012- 5th International Conference, Cologne, Germany, May 16-18, 2012, Revised Selected Papers*, pages 177–195, 2012.

[Bar16] R. Barbulescu. A brief history of pairings. In *International Workshop on the Arithmetic of Finite Fields WAIFI 2016, Jul 2016, Gand, Belgium*, pages Springer, 10064, 2016, Arithmetic of Finite Fields WAIFI 2016, 2016.

[BB04] D. Boneh, X. Boyen, and H. Shacham . Short group signatures. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 41–55, 2004.

[BF01] D. Boneh and M. 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*, pages 213–229, 2001.

[BGDM+10a] J.-L. Beuchat, J.E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya. High-speed software implementation of the optimal ate pairing over barreto-naehrig curves. *M. Joye, A. Miyaji, A. Otsuka (eds.) Pairing 2010. LNCS Springer, Heidelberg*, 6487 :21–39, 2010.

[BGDM+10b] J.-L. Beuchat, J.E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya. High-speed software implementation of the optimal ate pairing over barreto-

naehrig curves. *M. Joye, A. Miyaji, A. Otsuka (eds.) Pairing 2010. LNCS*, 6487 :21–39, 2010.

[BGK15] R. Barbulescu, P. Gaudry, and T. Kleinjung. The tower number field sieve. In *Advances in Cryptology - ASIACRYPT 2015, Volume 9453 of Lecture Notes in Comput. Sci*, pages 31–55, 2015.

[BK98] R. Balasubramanian and N. Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the menezes-okamoto-vanstone algorithm. *Journal of Cryptology*, 11 :141–145, 1998.

[BLS02] P. S. L. M. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, pages 257–267, 2002.

[BLS04a] P. S. L. M. Barreto, B. Lynn, and M. Scott. On the selection of pairing-friendly groups. In *Selected Areas in Cryptography-SAC 2003, Lectures Notes in Compt. Sci., vol. 3006*, pages 17–25. Springer–Verlag, Berlin, 2004.

[BLS04b] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *J. Cryptology*, 17(4) :297–319, 2004.

[BMM00] I. Biehl, B. Meyer, and V. Muller. Differential fault attacks on elliptic curve cryptosystems. In *Advances in Cryptology-CRYPTO 2000, volume 1880 of Lecture Notes in Comput.Sci.*, pages 131–146, Springer, 2000.

[BN05] P. S. L. M. Barreto and M. 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*, pages 319–331, 2005.

[BP98] D. Bailey and C. Paar. Optimal extension fields for fast arithmetic in public-key algorithms. In *CRYPTO'98, LNCS 1462*, pages 472–485, Springer, 1998.

[BS04] S. Baktir and B. Sunar. Optimal tower fields. *IEEE Transactions on Computes*, 53(10) :1231–1243, October, 2004.

[BSS00] I. F. Blake, G. Seroussi, and N. P. Smart. Elliptic curves in cryptography. *London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge*, 265, Reprint of the 1999 original,2000.

[BSS05] I.F. Blake, G. Seroussi, and N.P. Smart. *Advances in Elliptic Curves in Cryptography*. London Mathematic Society, Cambridge University Press, 2005.

[CCS06] L. Chen, Z. Cheng, and N.P. Smart. A built-in decisional function and security proof of id-based key agreement protocols from pairings. *IACR Cryptology ePrint Archive*, 2006 :160, 2006.

[CLN10] C. Costello, T. Lange, and M. Naehrig. Faster pairing computations on curves with high-degree twists. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, pages 224–242, 2010.

[Coc01] C. Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding, 8th IMA International Conference, Cirencester, UK, December 17-19, 2001, Proceedings*, pages 360–363, 2001.

[Coh95] H. Cohen. *A course in Computational Algebraic Number Theory*. Springer-Verlag, 1995.

[CP01] C. Cocks and R. G. E. Pinch. Identity-based cryptosystems based on the weil pairing. *Unpublished manuscript*, 170, 2001.

[DBS04] R. Dutta, R. Barua, and P. Sarkar. Pairing-based cryptographic protocols : A survey. *IACR Cryptology ePrint Archive*, 2004 :64, 2004.

[DCC05] P. Duan, S. Cui, and C. W. Chan. Special polynomial families for generating more suitable elliptic curves for pairing-based cryptosystems. *IACR Cryptology ePrint Archive*, 2005 :342, 2005.

[DEM05] R. Dupont, A. Enge, and F. Morain. Building curves with arbitrary small mov degree over finite prime fields. *Journal of Cryptology*, 18 :78–89, 2005.

[Die11] C. Diem. On the discrete logarithm problem in elliptic curves. *Compositio Mathematica*, 147(01) :75–104, 2011.

[DPP+15] D.Moody, R. C. Peralta, R. A. Pelner, A. R. Regenscheid, A. L. Roginsky, and L. Chen. Report on pairing based cryptography. *Journal of Resaerch (NIST-JRES)-120.002*, 120, 2015.

[DSD07] A. J. Devegili, M. Scott, and R. Dahab. Implementing cryptographic pairings over barreto-naehrig curves. In *Pairing-Based Cryptography - Pairing 2007, First International Conference, Tokyo, Japan, July 2-4, 2007, Proceedings*, pages 197–207, 2007.

[Est10] N. Estibals. Compact hardware for computing the tate pairing over 128 bit ecurity supersingular curves. In *Pairing 2010, LNCS 6487*, pages 397–416, Springer, 2010.

[FG98] G. Frey and H. Gangl. How to disguise an elliptic curve (weil descent). In *In talk at ECC'98*, 1998.

[FKR11] L. Fuentes-Castañeda, E. Knapp, and F. Rodríguez-Henríquez. Faster hashing to ${\mathbb G}_2$. In *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, pages 412–430, 2011.

[FMP16] E. Fouotsa, N. EL Mrabet, and A. Pecha. Optimal ate pairing on elliptic curves with embedding degree 9, 15, 27. *IACR Cryptology ePrint Archive*, 2016 :1187, 2016.

[FST10] D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 23(2) :224–280, 2010.

[Gau09] P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *Journal of Symbolic Computation, Elsevier*, 44(12) :1690–1702, 2009.

[GF16] L. Ghammam and E. Fouotsa. On the computation of the optimal ate pairing at the 192 bit security level. *IACR Cryptology ePrint Archive*, 2016 :130, 2016.

[GHO+07] R. Granger, F. Hess, R. Oyono, N. Thériault, and F. Vercauteren. Ate pairing on hyperelliptic curves. *Advances in Cryptology - EUROCRYPT 2007*, LNCS 4515 :430–447, 2007.

[GHS02] S. Galbraith, F. Hess, and N. Smart. Extending the ghs weil descent attack. In *EUROCRYPT 2002, LNCS 2332*, pages 29–44, Springer, 2002.

[GPS06] R. Granger, D. Page, and N.P. Smart. High security pairing-based cryptography revisited. In *Algorithmic Number Theory 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006. Proceedings*, pages 480–494, 2006.

[Gra10] R. Granger. On the static diffie-hellman problem on elliptic curves over extension fields. In *ASIACRYPT 2010, LNCS 6477*, pages 283–302, Springer, 2010.

[GS06] R. Granger and N.P. Smart. On computing products of pairings. *IACR Cryptology ePrint Archive*, 2006 :172, 2006.

[GS10] R. Granger and M. Scott. Faster squaring in the cyclotomic subgroup of sixth degree extensions. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, pages 209–223, 2010.

[Hes08] F. Hess. Pairing lattices. In *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, pages 18–38, 2008.

[HSV06] F. Hess, N. P. Smart, and F. Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10) :4595–4602, 2006.

[JL03] A. Joux and R. Lercier. Improvement to the general number field for discrete logarithms in prime fields. *Mathematics of Computation*, 72(242) :953–967, 2003.

[Jou00] A. Joux. A one round protocol for tripartite diffie hellman. In *Proceedings of ANTS 4, LNCS 1838*, pages 385–394, 2000.

[JP13] A. Joux and C. Pierrot. The special number field sieve in $\mathbb{F}_{p^n}$-application to pairing-friendly constructions. In *Pairing-Based Cryptography - Pairing 2013, Volume 8365 of Lecture Notes in Comput. Sci*, pages 45–61, 2013.

[JV10] A. Joux and V. Vitse. Elliptic curve discrete logarithm problem over small degree extensions fields. application to the static diffie-hellman on $e(\mathbb{F}_{q^5})$. *IACR Cryptology ePrint Archive*, 2010 :157, 2010.

[Kar13] K. Karabina. Squaring in cyclotomic subgroups. *Math. Comput.*, 82(281), 2013.

[KB16] T. Kim and R. Barbulescu. Extended tower number field sieve : A new complexity for medium prime case. In *Advances in Cryptology- CRYPTO 2016, LNCS 9814*, pages 543–571, 2016.

[Ked03] K. Kedlaya. Counting points on hyperelliptic curves using monsky- washnitzer cohomology. *J. Ramanujan Math. Soc.*, 18(4) :417–418, 2003.

[KM05] N. Koblitz and A. Menezes. Pairing-based cryptography at high security levels. In *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, pages 13–36, 2005.

[Kra22] M. Kraitchik. *Théorie des nombres.* Paris, Gauthier-Villars, 1922.

[Kra24] M. Kraitchik. *Recherches sur la théorie des nombres.* Paris, Gauthier-Villars, 1924.

[LMN10] V. Lauter, P.L. Montgomery, and M. Naehrig. An analysis of affine coordinates for pairing computation. In *Proceedings of the 4th international conference on Pairing-based cryptography. Pairing'10, Berlin, Heidelberg, Springer-Verlag*, 2010.

[LN97] R. Lidl and H. Niederreiter. *Finite Fields*. Cambridge university Press, Cambridge, 1997.

[LQ04] B. Libert and J.-J. Quisquater. Identity based undeniable signatures. In *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, pages 112–125, 2004.

[LT12] D. Le and C. Tan. Speeding up ate pairing computation in affine coordinates. In *Information Security and Cryptology - ICISC 2012 - 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers*, pages 262–277, 2012.

[LZZW08] X. Lin, C. Zhao, F. Zhang, and Y. Wang. Computing the ate pairing on elliptic curves with embedding degree $k = 9$. *IEICE Transactions*, 91-A(9) :2387–2393, 2008.

[MGI09] N. El Mrabet, N. Guillermin, and S. Ionica. A study of pairing computation for elliptic curves with embedding degree 15. *IACR Cryptology ePrint Archive*, 2009/370, 2009.

[MGI11] N. El Mrabet, A. Guillevic, and S. Ionica. Efficient mulptiplication finite field extensions of degree 5. In *A. Nitaj and D. Pointcheval (Eds). AFRICACRYPT 2011, LNCS 6737*, pages 188–205, 2011.

[Mil04] V. S. Miller. The weil pairing, and its efficient calculation. *J. Cryptology*, 17(4) :235–261, 2004.

[MNT01] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for fr-reduction. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 84(5 :1234–1243, 2001.

[MOV93] A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on information Theory*, 39(5) :1639–1646, 1993.

[MSS16] A. Menezes, P. Sarkar, and S. Singh. Challenges with assessing the impact of nfs advances on the security of pairing-based cryptography. *IACR Cryptology ePrint Archive*, 2016/1102, 2016.

[Pol78] J. Pollard. Monte carlo methods for index computation (mod p). *Mathematics of Computation*, 32 :918–924, 1978.

[Sat02] T. Satoh. On p-adic point counting algorithms for elliptic curves over finite fields. In *Algorithmic number theory (Sydney, Australia, 2002), volume 2369 of Lecture Notes in Comput. Sci.*, pages 43–66. Springer–Verlag, Berlin, 2002.

[SBC+09] M. Scott, N. Benger, M. Charlemagne, L. J. D. Perez, and E. J. Kachisa. Fast hashing to G_2 on pairing-friendly curves. In *Pairing-Based Cryptography - Pairing 2009, Third International Conference, Palo Alto, CA, USA, August 12-14, 2009, Proceedings*, pages 102–113, 2009.

[Sch85] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, 44(170) :483–494, 1985.

[Sch95] R. Schoof. Counting points on elliptic curves over finite fields. *Journal de Théorie des Nombres de Bordeaux*, 7 :219–254, 1995.

[Sch00] O. Schirokauer. Using number fields to compute logarithms in finite fields. *Mathematics of Computation*, 69 :1267–1283, 2000.

[Sco11] M. Scott. On the efficient implementation of pairing-based protocols. *Cryptography and Coding 2011*, LNCS 7089 :296–308, 2011.

[Sem04] I. Semaev. Summation polynomials and the discrete logarithm problem on elliptic curves. *IACR Cryptology ePrint Archive*, 2004/031, 2004.

[Sho97] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology-Eurocrypt 1997, LNCS VOL. 1233*, pages 256–266, Springer–Verlag, Berlin, 1997.

[Sil09] J.H. Silverman. *The Arithmetic of Elliptic Curves, Second Edition.* Graduate Texts in Mathematics 106, Springer Science+Business Media, LLC, 2009.

[SS16a] P. Sarkar and S. Singh. A generalisation of the conjugation method for polynomial selection for the extended tower number field sieve algorithm. *IACR Cryptology ePrint Archive*, 2016/537, 2016.

[SS16b] P. Sarkar and S. Singh. New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields. In *Advances in Cryptology- EUROCRYPT 2016, LNCS 9665*, pages 429–458, 2016.

[STNM10] Y. Sakemi, S. Takeuchi, Y. Nogami, and Y. Morikawa. Accelering twisted ate pairing with frobenius map, small scalar multiplication, and multi-pairing. *ICISC 2009*, LNCS 5984 :47–64, 2010.

[Ver10] F. Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1) :455–461, 2010.

[Was08] L.C. Washington. *Elliptic Curves, Number Theory and Cryptography*. Discrete Math .Aplli, Chapman and Hall, 2008.

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2005, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pages 114–127, 2005.

[Wei48] A. Weil. *Courbes algébriques et variétés abéliennes (in french)*. Hermann, 1948.

[ZL12] X. Zhang and D. Lin. Analysis of optimum pairing products at high security levels. In *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, pages 412–430, 2012.

[ZWL13] X. Zhang, X. Wang, and D. Lin. On efficient pairings on elliptic curves over extension field. *Pairing-Based Cryptography-Pairing 2012, Lectures Notes in Computer Science*, 7708 :1–18, 2013.

[ZZH08] C.-A. Zhao, F. Zhang, and J. Huang. All pairings are in a group. *IEICE Trans. Fundamentals*, E91-A, no. 10 :3084–3087, 2008.

[ZZX08] C.-A. Zhao, F. Zhang, and D. Xie. Reducing the complexity of the weil pairing computation. *IACR Cryptology ePrint Archive*, 2008 :212, 2008.

Annexe A

Codes Pari GP de génération de bon paramètres des courbes

A.1 Code Pari/GP pour les courbes avec k=9

Ce code me donne un x0 avec un poids de hamming 4, r de taille 257 bits.

```
P=((x+1)^2+((x-1)^2*(2*x^3+1)^2)/3)/4;
R=(x^6+x^3+1)/3;
T=x+1;
for(n=43,44,for(u=-1,1,for(v=-1,1,for(y=-1,1,
for(i=2,n-1,for(j=1,i-1,x0=2^n+u*2^i+v*2^j+y;
RS=subst(R,x,x0); if(type(RS)=="t_INT",
if(ispseudoprime(RS)==1, PS=subst(P,x,x0);
if(type(PS)=="t_INT", if(ispseudoprime(PS)==1,TS=subst(T,x,x0);
if(type(TS)=="t_INT",if(PS%6==1, if(x0%6==1,
if(floor(log(RS)/log(2))>254,
print([n,u,i,v,j,y,x0,x0%6,PS%6,floor(log(RS)/log(2)),
floor(log(PS)/log(2))])
))))))))))))
```

A.2 Code Pari/GP pour les courbes avec k=15

Ce code me donne un x0 avec un poids de hamming 5, r de taille 385 bits.

```
P=(x^12-2*x^11+x^10+x^7-2*x^6+x^5+x^2+x+1)/3;
```

```
R=x^8-x^7+x^5-x^4+x^3-x+1;
T=x+1;
for(n=45,50,for(u=0,1,for(v=0,1,for(w=0,1,for(y=0,1,
for(i=2,n-1,for(j=1,i-1,for(k=1,j-1,x0=2^n+u*2^i+v*2^j+w*2^k+y;
RS=subst(R,x,x0); if(type(RS)=="t_INT", if(ispseudoprime(RS)==1,
PS=subst(P,x,x0); if(type(PS)=="t_INT", if(ispseudoprime(PS)==1,
TS=subst(T,x,x0); if(type(TS)=="t_INT", if(PS%5==1,
print([n,u,i,v,j,w,k,y,x0,x0%5,PS%5,PS%10,
floor(log(RS)/log(2)),floor(log(PS)/log(2))])
))))))))))))))
```

A.3 Code Pari/GP pour les courbes avec k=27

Ce code nous donne un x0 d'un poids de hamming 4, r possède un facteur premier de taille 514 bits.

```
P=1/3(x-1)^2(x^{18}+x^9+1)+x;
R=1/3(x^{18}+x^9+1);
T=x+1;
for(n=28,29,for(u=0,1,for(v=0,1,for(w=0,1,for(y=0,1,
for(i=2,n-1,for(j=1,i-1,for(k=1,j-1,x0=2^n+u*2^i+v*2^j+w*2^k+y;
RS=subst(R,x,x0); if(type(RS)=="t_INT",if(ispseudoprime(RS)==0,
PS=subst(P,x,x0); if(type(PS)=="t_INT", if(ispseudoprime(PS)==1,
TS=subst(T,x,x0); if(type(TS)=="t_INT", if(PS%6==1, if(x0%6==4,
if(floor(log(RS)/log(2))>518,
print([n,u,i,v,j,w,k,y,x0,x0%6,PS%6,floor(log(RS)/log(2)),
floor(log(PS)/log(2))])
))))))))))))))))
```