Conception de courbes elliptiques et applications

Rémi Clarisse

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Mathématiques et leurs interactions*

Par
Rémi CLARISSE

Conception de courbes elliptiques et applications

Thèse présentée et soutenue à Rennes, le jeudi 16 décembre 2021
Unités de recherche : IRMAR (Rennes) et Orange Innovation (Cesson-Sévigné)

Rapporteur·e·s avant soutenance :

Guilhem CASTAGNOS	Maître de conférence, Université de Bordeaux, Talence
Aurore GUILLEVIC	Chargée de recherche Inria, LORIA, Villiers-lès-Nancy

Composition du Jury :

Rapporteur :	Guilhem CASTAGNOS	Maître de conférence, Université de Bordeaux, Talence
Président du Jury :	Jean-Marc COUVEIGNES	Professeur, Université de Bordeaux, Talence
Directeur de thèse :	Sylvain DUQUESNE	Professeur, Université de Rennes 1, Rennes
Examinatrice :	Nadia EL-MRABET	Maître de conférence, MINES Saint-Étienne, Gardanne
Examinateur :	Pierre-Alain FOUQUE	Professeur, Université de Rennes 1, Rennes
Rapporteure :	Aurore GUILLEVIC	Chargée de recherche Inria, LORIA, Villiers-lès-Nancy

Invités :

Reynald LERCIER	Chercheur, DGA-MI, Bruz
Olivier SANDERS	Ingénieur de recherche, Orange Innovation, Cesson-Sévigné

Remerciements

Je commence par remercier les membres du jury, Nadia El-Mrabet, Pierre-Alain Fouque, Reynald Lercier, qui ont accepté d'examiner le travail effectué durant ce doctorat et Jean-Marc Couveignes d'avoir assuré la présidence de ce jury. En plus d'être membres du jury, Guilhem Castagnos et Aurore Guillevic ont accepté la tâche de rapporter cette thèse : je les remercie donc pour avoir consacré une partie de leur été à cette lecture et pour m'avoir fait part de leurs commentaires.

J'ai découvert la crytographie il y a cinq ans de cela, et je me suis empressé de m'inscrire en Master Cryptologie et Sécurité Informatique de l'Université de Bordeaux. Je remercie donc vivement Gilles Zémor et toute l'équipe pédagogique du Master pour les enseignements dispensés et la qualité des cours. C'est en première année de master, lors d'un stage à l'Inria Saclay, que l'idée de faire une thèse est rentrée dans le champs des possibles. Daniel Augot m'a accueilli au sein de son équipe et Luca De Feo m'a encadré et conseillé avec une grande bienveillance. Je les remercie chaleureusement tout deux de m'avoir reçu et supervisé durant trois mois. Je tiens également à remercier Julia, qui m'a accompagné durant mon stage de seconde année, pour la justesse de ses critiques et la qualité de son encadrement. C'est grâce à toi que ce stage me laisse un heureux souvenir.

J'éprouve une grande reconnaissance envers Sylvain et Olivier, mes deux superviseurs, pour m'avoir accepté en thèse. À Sylvain, pour avoir aiguillé mes recherches lorsqu'elles méandraient, avoir su me remonter le moral au bon moment et m'avoir envoyé en conférences rencontrer d'autres chercheu·r·se·s ; à Olivier, pour avoir partagé son savoir avec moi, fait preuve d'une patience sans borne et été un soutien sans faille ; je vous exprime à tous les deux ma gratitude. Je vous remercie aussi de m'avoir laissé la possibilité d'organiser les séances de travaux dirigés des premières années du Master, et j'en profite pour remercier lesdit·e·s étudiant·e·s d'avoir participé à cette expérience enrichissante.

La majeure partie de ma thèse s'est déroulée au sein de l'équipe NCP à Orange Innovation, sur le site d'Orange Atalante. Je tiens à remercier chaque membre de cette équipe (et ses nombreux stagiaires) de m'avoir accueilli et fait bénéficier d'un cadre d'exceptionnel pour mes recherches. J'ai une pensée toute particulière pour les autres doctorants d'Orange Innovation, notamment Adina, Marcel, Zakaria, Elie C., Elie B., Andy ; ce sont

elleux qui ont rendu cette thèse bien plus agréable en l'agrémentant de leurs lumineuses présences. J'en profite aussi pour remercier Gautier et Antoine, avec qui j'ai noué une amitié au gré de rencontres universitaires.

Merci à Cécile Pierrot de m'avoir invité au séminaire de l'équipe CARAMBA du Loria de Nancy. J'en remercie les membres pour leur accueil et les échanges intéressants qui en ont résultés. Merci aussi infiniment à Aurore Guillevic pour ses discutions concernant les courbes elliptiques à couplage et de m'avoir éclairé sur les algorithmes NFS.

Je remercie aussi celles qui m'ont permis de naviguer les arcanes administratives et universitaires : Nelly Loton, Annie Quéméré, Crystelle Innocenti, Marie-Aude Verger et Xhensila Lachambre.

J'ai aussi une pensée spéciale pour Sofiane, que j'ai eu le plaisir d'encadrer pour son stage de fin de master. Je te remercie et te souhaite de faire un doctorat riche en découverte et de t'épanouir au Québec.

À tous ces amis qui gravitent autour de moi : Jared, Stéphanie, Maëva, Anthony, Antton, Geoffrey, Maxime, Clément, Yan, Ambre, Thibault, Paul et d'autres à venir ; mille mercis de m'avoir sorti de mes pérégrinations mathématiques et d'avoir enrobées de douceur ces trois dernières années.

Enfin, je suis rempli de gratitude et d'amour en pensant à ma mère et ma tante, Bernadette et Élisabeth ; je vous remercie de m'avoir soutenu, supporté et d'avoir cru en moi durant toutes ses années d'études. Tout cela aurait été impossible sans vous !

Le dernier mot ira à Allan, avec qui je suis heureux de partager cette mascarade que l'on appelle la vie : merci !

TABLE DES MATIÈRES

Introduction

La cryptographie trouve ses premières applications dans les échanges militaires. En effet, l'interception possible d'un message par l'adversaire, divulguant ainsi les stratégies offensives ou défensives, incitait à chiffrer, *i.e.* rendre inintelligible, le message. On peut par exemple citer le chiffrement de César qui est un schéma de chiffrement par substitution mono-alphabétique, consistant en un simple décalage circulaire des lettres de l'alphabet.

L'un des exemples emblématiques du siècle précédent est l'utilisation de la machine *Enigma* par les forces nazies durant la seconde guerre mondiale. Sa cryptanalyse a été faite par une équipe de *Bletchley Park*, dont faisait partie Alan Turing, et il est estimé que grâce à celle-ci, la guerre a été écourtée de deux à quatre ans [Hin93].

Ces systèmes de chiffrement n'utilisent essentiellement qu'une clé pour chiffrer et déchiffrer des messages. Cela veut dire que l'expéditeur et le récipiendaire ont un secret commun permettant une communication sécurisée entre eux. Là où le bât blesse, c'est qu'il n'est pas évident de partager un secret avec une personne inconnue ou à l'autre bout du monde, surtout lorsque aucun des interlocuteurs n'a de moyen étatique.

L'un des grands écueils de la cryptographie symétrique, c'est à dire lorsque les deux parties ont un secret partagé, est donc la transmission des clés secrètes. Obstacle qu'a su lever un autre pan de la cryptographie, dite cryptographie asymétrique.

Naissance de la cryptographie asymétrique

C'est en 1976 que Whitfield Diffie et Martin Hellman ont donné une *nouvelle direction* à la cryptographie [DH76]. Dans leur article, les auteurs soulignent que les avancées technologiques offrent de nouveaux outils à la puissance de calcul jusqu'alors inatteignable. Leur proposition révolutionnaire est l'utilisation de deux clés, l'une publique servant uniquement au chiffrement et l'autre privée destinée au déchiffrement. Pour ce faire, ils préconisent l'usage de fonctions à sens unique. Simplement dit, ce sont des fonctions dont un sens est facile à calculer et dont la réciproque est plus difficile à effectuer, sous-entendu plus coûteuse en ressources (*e.g.* temps, mémoire, énergie). Une telle fonction est donnée par Diffie et Hellman : l'exponentielle dans un corps fini, dont la réciproque est le loga-

rithme discret. L'exponentiation est faisable en temps polynomial à l'aide d'algorithme du type *square-and-multiply* [Coh+05]. En revanche, s'assurer que la fonction réciproque, le logarithme, est suffisamment dure à calculer se fait en identifiant le problème calculatoire sous-jacent.

Dans un groupe $\mathbb{G}$, noté multiplicativement, soit un élément $g \in \mathbb{G}$ d'ordre n. Résoudre le problème du logarithme discret (DLP, de l'anglais *Discrete Logarithm Problem*) en base g est, étant donné $h \in \mathbb{G}$, trouver $x \in \mathbb{Z}/n\mathbb{Z}$, s'il existe [1], tel que $g^x = h$. Le groupe suggéré par Diffie et Hellman est le groupe multiplicatif d'un corps fini à q éléments : $(\mathbb{F}_q \backslash \{0\}, \times)$.

Ils proposèrent le schéma d'échange de clés suivant : deux interlocuteurs, Alice et Bob n'ont pas de secret en commun, ils ne peuvent donc pas utiliser de schémas de chiffrement symétrique pour sécuriser leurs échanges. Pour partager un secret, ils se mettent publiquement d'accord sur un ordre q et un générateur g du groupe $\mathbb{F}_q^\times = \mathbb{F}_q \backslash \{0\}$. Chacun tire aléatoirement un scalaire secret dans $\mathbb{Z}/(q-1)\mathbb{Z}$, disons a pour Alice et b pour Bob. Ensuite, Alice envoie $A = g^a$ à Bob et il fait de même avec $B = g^b$. Ont alors transité sur le canal public les éléments A et B. Après réception, Alice calcule B^a et Bob calcule A^b. Par commutativité du groupe $\mathbb{F}_q^\times$, les deux protagonistes ont le secret commun $B^a = A^b = g^{ab}$. Un attaquant passif devrait ainsi calculer g^{ab} connaissant les éléments g, g^a et g^b, ce qui est connu comme le problème calculatoire Diffie-Hellman (CDH, pour *Computational Diffie-Hellman* en anglais). Le problème CDH est clairement plus simple que le DLP : si un attaquant peut calculer des logarithmes discrets dans $\mathbb{G}$ alors il n'aura pas de mal à récupérer $a = \log_g A$ et calculer $B^a = g^{ab}$.

L'échange de clé Diffie-Hellman peut être instancié sur n'importe quel groupe $\mathbb{G}$ où le calcul des logarithmes discrets est difficile. Outre le groupe $\mathbb{F}_q^\times$, les points $\mathbb{F}_q$-rationnels d'une courbe elliptique forment un groupe candidat intéressant.

Logarithme : courbe elliptique vs. corps fini

Dans un groupe générique [Sho97], seule la structure de groupe est connue et peut être utilisée pour le calcul du logarithme discret. Dans ces conditions, les meilleurs algorithmes sont du type Pas-de-Bébé, Pas-de-Géant [Sha71] ou Pollard-rho [Pol78], et sont de complexité exponentielle $O(\sum_i e_i(\sqrt{p_i} + \log n))$ pour $n = \prod_i p_i^{e_i}$ l'ordre du groupe [MvV97]. D'ailleurs, il n'est pas possible de trouver d'algorithmes avec une plus petite complexité

1. Il est coutume de directement considérer $\mathbb{G}$ engendré par g, *i.e.* $\mathbb{G}$ est cyclique, assurant ainsi l'existence d'un $x \in \mathbb{Z}/n\mathbb{Z}$ pour tout $h \in \mathbb{G}$.

pour les groupes génériques [Sho97].

Or, les structures de corps finis $\mathbb{F}_q$ peuvent être utilisées pour accélérer le calcul des logarithmes discrets dans le groupe $\mathbb{F}_q^\times$, qui n'est donc pas générique. C'est là tout le fondement du choix de bases adéquates pour les algorithmes par calcul d'indices [JP16] et qui a donné la famille des algorithmes de crible par corps de nombres (NFS pour *Number Field Sieve* en anglais). Ces algorithmes NFS ont une complexité sous-exponentielle, le DLP est donc plus facile à résoudre sur un corps fini que dans un groupe générique. La notation L est utilisée pour décrire la complexité de tels algorithmes sur un corps $\mathbb{F}_q$:

$$L_q(\alpha, c) = \exp\left((c + o(1))(\log q)^\alpha (\log\log q)^{1-\alpha}\right),$$

où $0 \leqslant \alpha \leqslant 1$ et $c > 0$. Intuitivement, lorsque $\alpha = 0$, la complexité L_q est polynomiale en $\log q$ alors que si $\alpha = 1$, la complexité L_q est exponentielle en $\log q$. Pour tout $0 < \alpha < 1$, la complexité $L_q(\alpha)$ est dite sous-exponentielle et la valeur de α va fortement dépendre de la forme de q (*e.g* si $q = 2^n$, alors α tend vers 0 quand n tend vers $+\infty$ [Bar+14] ; si q est premier, alors α sera proche de 1/3 [Gor93]).

Cependant, entre 1985 et 1990, plusieurs articles ont été publiés, démontrant qu'il est possible d'utiliser la jacobienne d'une courbe elliptique (de genre 1) ou hyperelliptique (de genre 2 ou plus). Ce sont Victor Miller [Mil86] et Niel Koblitz [Kob90] qui ont indépendamment introduit le groupe des points d'une courbe elliptique comme groupe approprié pour des cryptosystèmes fondés sur la difficulté du calcul du logarithme discret.

Comme les meilleurs algorithmes de calcul de logarithme discret sur les courbes elliptiques sont ceux génériques, à sécurité équivalente, les courbes offrent de meilleurs performances et ont des paramètres plus petits. En l'occurrence, elles sont encore massivement présentes sur internet, que ce soit pour les signatures numériques ou les échanges de clé Diffie-Hellman (avec ECDSA et ECDHE [Moe+06]). Cependant, elles nous sont d'intérêt pour une autre raison : les couplages cryptographiques.

Couplage et protection de la vie privée

Certaines courbes elliptiques sont compatibles avec une application bilinéaire efficacement calculable, nommée couplage, qui est à valeurs dans un corps fini et prend en entrée deux points d'une même courbe elliptique.

Malheureusement, le premier cas d'utilisation de couplage sur courbe fut une appli-

cation destructive. L'attaque de Menezes, Okamoto et Vanstone [MVO91] met en œuvre le couplage pour transférer le logarithme depuis la courbe, où il est difficile, vers le corps fini, où il est (plus) facile.

Durant la décennie qui suivit l'attaque MOV, il était souhaitable d'éviter les courbes à couplage. Un changement opéra au début des années 2000, où de nombreuses constructions utilisant le couplage ont été publiées, comme par exemple l'échange de clé tripartite de Joux [Jou04], les signatures courtes de Boneh, Lynn et Shacham [BLS01], ou encore le chiffrement fondé sur l'identité (IBE pour *Identity-Based Encryption* en anglais) de Boneh et Franklin [BF03].

Par la suite, des schémas beaucoup plus complexes ont vu le jour ou ont été adaptés aux couplages sur courbes elliptiques. C'est le cas par exemple des signatures de groupe [Cv91]. Les signatures de groupe font partie des schémas permettant une protection de la vie privée. Elles offrent un mécanisme d'anonymisation qui permet de minimiser l'information révélée lors d'une authentification tout en implémentant un garde-fou qui consiste en une procédure de levée d'anonymat.

L'importance des signatures de groupe est visible par la recherche active à leur sujet durant les 30 dernières années [Cv91 ; BMW03 ; BSZ05 ; Bic+10 ; LPY15 ; DS18], et par les divers variants du schéma tels que les DAA [BCC04] (pour *Direct Anonymous Attestation*) et les EPID [BL07b] (pour *Enhanced Privacy ID*), qui sont implémentées dans des milliards de puces [Int16 ; TCG15]. Plus généralement, les signatures de groupe ont des connections avec la plupart des schémas de protection de la vie privée, ce qui donne une plus grande ampleur aux avancées dans la conception des signatures de groupe. Toute amélioration pratique ou théorique dans le domaine des signatures de groupe impactera sans doute de façon similaire les nombreuses primitives qui leur sont apparentées.

Ces constructions ayant besoin des couplages, il devient alors capital de construire des courbes elliptiques à couplage telles que les calculs soient faisables et que la sécurité sur la courbe et sur le corps fini soit suffisante.

Nos contributions

Les travaux de recherches menés durant cette thèse ont eu pour but d'améliorer la mise en œuvre de schémas protégeant la vie privée d'utilisateurs. Généralement, cette amélioration se fait de deux façons différentes : soit en proposant de nouvelles primitives plus performantes, jouant sur l'aspect théorique, soit en suggérant l'utilisation d'outils

permettant des instanciations plus efficaces, jouant alors sur l'aspect pratique. Nous avons exploré ces deux pistes.

Amélioration des signatures de groupe grâce aux propriétés des couplages

Les signatures de groupe [Cv91] suivent une construction, devenue classique, introduite par Bellare, Shi et Zhang [BSZ05]. Celle-ci est souvent nommée SEP, pour *Sign-Encrypt-Prove*, car chaque membre du groupe produit une signature de groupe comme suit :

1. il signe le message avec ses clés personnelles ;
2. il chiffre la signature précédente avec le certificat d'appartenance au groupe ;
3. il prouve que le tout est bien formé.

Les étapes *chiffrer* et *prouver* sont relativement coûteuses, et cette dernière se fait généralement avec un schéma de preuve NIZK [2]. Ces preuves NIZK sont assez complexes, donnent de longues signatures de groupe et impactent la sécurité du schéma [Gro07 ; GS08 ; CGH04]. Une première avancée [Bic+10] a été de retirer le chiffrement, grâce à la randomisation du certificat d'appartenance [CL04], et de combiner la signature avec la preuve en une signature de connaissance, par la transformation Fiat-Shamir [FS87].

Le schéma de signature de groupe que nous proposons [CS20] s'affranchit de la contrainte des preuves, et du modèle de l'oracle aléatoire, grâce aux couplages. En effet, en constatant que les signatures randomisables Pointcheval-Sanders [PS16] et Fuchsbauer-Hanser-Slamanig [FHS15] interagissent bien et que leurs équations de vérification peuvent être combinées, nous avons détaillé un schéma de signature de groupe performant, avec des signatures courtes et des temps de signature plus rapides comparés à l'état de l'art.

Conception de courbes adaptées aux besoins cryptographiques

Le paradigme couramment adopté est de sélectionner et proposer des courbes elliptiques à couplage offrant l'évaluation du couplage (l'application entre la courbe et le corps fini) la plus avantageuse possible. Ce choix est généralement fait indépendamment des protocoles cryptographiques instanciant ces courbes. Notre approche partant de ce constat est d'inverser le cheminement de pensée : à partir de besoins cryptographiques donnés, quelles courbes sont les plus adaptées ? Ce raisonnement est particulièrement approprié lorsque les entités impliquées dans un protocole n'ont pas les mêmes ressources, *e.g.* une authentification entre un thermostat connecté et un serveur d'une multinationale.

2. Preuve non-interactive sans divulgation de connaissance.

Cette idée nous a amené à catégoriser des schémas usant du couplage, et une famille nous a semblé particulièrement pouvoir bénéficier de notre approche. En effet, les signatures de groupe et leurs variantes, comme les DAA [BCC04] et les EPID [BL07b], permettent souvent à leurs utilisateurs de signer en effectuant certaines opérations dans le premier groupe du couplage, la vérification se faisant à l'aide de couplages. D'ailleurs, dans le cas des enclaves cryptographiques devant s'authentifier [Int16 ; TCG15], réduire le coût de la signature est d'autant plus pertinent que ces plateformes sont limitées en ressources (*e.g.* mémoire, puissance).

Ainsi, nous avons appliqué les travaux de Brezing et Weng [BW05] afin de trouver des courbes plus performantes sur le premier groupe du couplage. La nouvelle courbe proposée `BW19-P286` [CDS20] est définie sur un corps d'ordre un premier de 286 bits. Les éléments du premier groupe du couplage sont représentables en usant 9 mots machines sur une architecture 32 bits et les opérations dans ce groupe sont 1.5 à 2 fois plus rapides que sur des courbes classiques (BN, BLS ou KSS, elles définies sur des corps d'au moins 330 bits).

Organisation de ce manuscrit

Dans les deux premiers chapitres, nous aborderons toute la théorie qui permettra d'expliquer le reste du manuscrit. Le Chapitre I sert de préambule à la théorie des nombres. Le Chapitre II introduit les courbes elliptiques, plus particulièrement celles sur corps finis, ainsi que les couplages sur celles-ci. Le Chapitre III donne des courbes à couplage performantes dans le premier groupe du couplage et le Chapitre IV décrit un nouveau schéma de signatures de groupe.

Chapitre I

Notions de théorie des nombres

La théorie des nombres est la branche des mathématiques qui traite historiquement de l'arithmétique des nombres entiers. Elle est connexe à d'autres domaines, tels que l'algèbre, l'analyse ou encore la géométrie. Dans ce premier chapitre, nous nous proposons d'aborder en préambule les principales bases de l'algèbre : les groupes et les anneaux. Nous donnons les définitions utiles au reste du manuscrit, sans nous étendre sur la vaste théorie qui les entoure. Ensuite, nous définissons les corps finis, sur lesquels nous instancierons les courbes elliptiques (section II.1), et détaillons la structure des corps de nombres quadratiques imaginaires, indispensables à la compréhension de la structure de l'anneau d'endomorphismes de certaines courbes elliptiques. Enfin, nous introduisons le problème du logarithme discret et donnons quelques méthodes pour le résoudre.

Les notions abordées dans ce chapitre sont des notions classiques de licence et de master, le lecteur peut donc passer au chapitre suivant sans craindre d'être perdu. Le lecteur souhaitant développer plus en détails les notions de ce chapitre peut consulter le livre [Lan05] et certaines rubriques de [Con].

I.1 Préambule

La principale structure que nous utilisons, outre les courbes elliptiques, est la structure de corps. Avant d'en donner une définition ainsi que quelques généralités à leur propos, rappelons les définitions de groupes (structure des points d'une courbe elliptique et des inversibles d'un corps) et d'anneaux (structure de l'ensemble des endomorphismes sur une courbe elliptique).

I.1.1 Groupes et anneaux

La structure de groupe est l'une des plus simples mais aussi une des plus courantes, particulièrement en cryptographie, où un des problèmes fondamentaux est le problème du

logarithme discret qui se définit dans un groupe.

Définition I.1.1 (Groupe)**.** Soient G un ensemble et $\star : G \times G \to G$ une loi de composition interne. La loi $\star$ fait de G un groupe si elle est

Associative : pour tous $x, y, z \in G$, $(x \star y) \star z = x \star (y \star z)$;

Unifère : il existe $1_G \in G$ tel que pour tout $x \in G$, $x \star 1_G = x$;

Symétrique : pour tout $x \in G$, il existe $y \in G$ tel que $x \star y = 1_G$.

Il est courant d'omettre la loi $\star$ lorsqu'elle est déductible du contexte, on note alors $xy := x \star y$. Lorsque $xy = yx$ pour tout $(x, y) \in G^2$, la loi est dite commutative et le groupe *abélien* ou *commutatif*. En fonction de la notation additive ou multiplicative de la loi interne, la littérature appelle *opposé*, noté $-x$, ou *inverse*, noté x^{-1}, le symétrique de x. Nous notons les groupes de façon multiplicative dans la suite.

L'*ordre* d'un élément x dans un groupe est (s'il existe) le plus petit entier positif a tel que $x^a = 1_G$. Remarquons que si $x^b = 1_G$ alors l'ordre de x divise b. Pour un élément $x \in G$, on note $\langle x \rangle \subset G$ le *groupe engendré* par toutes les puissances de x.

Si un sous-ensemble H d'un groupe G est lui aussi un groupe, on dit que H est un *sous-groupe* de G. La plupart des groupes de ce manuscrit sont commutatifs, faisant de l'ensemble des classes d'équivalence G/H un groupe [1], appelé *groupe quotient*, où deux éléments x et y sont dans la même classe d'équivalence si, et seulement si, $xy^{-1} \in H$.

Définition I.1.2 (Groupe fini et groupe cyclique)**.** Un groupe G est *fini* lorsqu'il a un nombre fini d'éléments, son *cardinal* (aussi dit *ordre*) est noté $\#G \in \mathbb{N}^*$. Un groupe G est *cyclique* lorsqu'il est fini et engendré par un seul élément, *i.e.* il existe $g \in G$ d'ordre fini tel que $\langle g \rangle = G$ et g est appelé *générateur*.

Les groupes cycliques sont le cadre idéal pour présenter le problème du logarithme discret : soit G un groupe dont $g \in G$ est un générateur, alors tout $h \in G$ s'écrit g^a pour un entier a. Il est possible de faire l'analogie avec l'exponentielle réelle : tout réel positif $x > 0$ s'écrit e^y pour un réel y, où e est le nombre d'Euler ; et la fonction logarithme népérien permet d'exprimer y en fonction de x, à savoir $y = \ln(x)$. Dans notre groupe G, les puissances du générateur g décrivent tout le groupe : on note alors $\log_g(h)$ le plus petit entier a tel que $g^a = h$. Cet entier $\log_g(h)$ est le *logarithme discret de h en base g*. Trouver $\log_g(h)$ sachant g et h, c'est résoudre le *problème du logarithme discret*. Nous y reviendrons plus loin en section I.4.

1. Sinon, il faut que H soit un sous-groupe distingué de G, *i.e.* $\forall x \in G, xH = Hx$.

Le théorème de Lagrange donne une relation fondamentale entre l'ordre d'un groupe G et l'ordre d'un sous-groupe $H \subset G$. Dans le cas des groupes commutatifs finis, le théorème de classification des groupes abéliens finis donne même la structure d'un groupe comme somme directe de groupes cycliques.

Théorème I.1.1 (Théorème de Lagrange)**.** *Soient G un groupe fini et H un sous-groupe de G. L'ordre du sous-groupe divise l'ordre du groupe, i.e. $\#H$ divise $\#G$.*

Théorème I.1.2 (Classification des groupes abéliens finis [Fuc15])**.** *Soit G un groupe abélien fini. Il existe un entier $k > 0$ et des entiers $n_i > 0$, pour $1 \leqslant i \leqslant k$, avec n_i divisant n_{i+1}, tels que*

$$G \simeq C_{n_1} \times C_{n_2} \times \cdots \times C_{n_k},$$

en notant C_n le groupe cyclique d'ordre n.

Remarque I.1.1. Il est commun de prendre le groupe additif de l'anneau $\mathbb{Z}/n\mathbb{Z}$ comme groupe cyclique d'ordre n archétypal. Dans ce cas là, l'isomorphisme de groupes du Théorème I.1.2 s'écrit

$$G \simeq \frac{\mathbb{Z}}{n_1\mathbb{Z}} \times \frac{\mathbb{Z}}{n_2\mathbb{Z}} \times \cdots \times \frac{\mathbb{Z}}{n_k\mathbb{Z}}.$$

Maintenant, regardons les structures à deux lois internes, en l'occurrence les anneaux.

Définition I.1.3 (Anneau)**.** Soient A un ensemble et deux lois de composition internes, l'une notée additivement $+$ et l'autre multiplicativement $\times$. Pour que A soit un anneau, les lois doivent vérifier :

- il existe $0_A \in A$ tel que la loi $+$ fasse de A un groupe abélien de neutre 0_A ;
- la loi $\times$ est associative et se distribue sur la loi $+$, c'est-à-dire que pour tous $x, y, z \in A$,

$$x \times (y + z) = x \times y + x \times z \quad \text{et} \quad (x + y) \times z = x \times z + y \times z.$$

Les anneaux que nous rencontrerons sont *unitaires*, à savoir qu'il existe un élément neutre pour $\times$, noté $1_A \in A$. Nous les appellerons anneaux (sans préciser unitaires). Un anneau est dit *commutatif* lorsque la loi $\times$ est commutative. Supposer un anneau commutatif permet de simplifier les définitions, dans le sens où certains éléments ont une propriété *à gauche* mais pas forcément *à droite* et inversement.

Définition I.1.4 (Diviseur de zéro et groupe des inversibles)**.** Soit A un anneau (unitaire, commutatif). Un élément $a \in A$ est un *diviseur de zéro* s'il existe $b \in A$ tel que $a \times b = 0_A$.

Un élément $a \in A$ est une *unité* ou *inversible* s'il existe $b \in A$ tel que $a \times b = 1_A$. L'ensemble des unités forme un groupe pour la loi $\times$, noté $A^\times$ et appelé *groupe des inversibles*.

L'élément neutre de l'addition 0_A n'appartient jamais au groupe des inversibles et un élément a de l'anneau ne peut pas être simultanément diviseur de zéro et inversible. Un anneau A est *intègre* s'il est commutatif et sans diviseur de zéro.

Définition I.1.5 (Algèbre à division et corps)**.** Soit A un anneau (sans diviseur de zéro, pas forcément commutatif). Si $A \setminus \{0_A\} = A^\times$, *i.e.* tout élément non nul est inversible, alors A est dit une *algèbre à division*. Si de plus $(A^\times, \times)$ est commutatif alors A est dit un *corps*.

Homomorphismes de groupes et d'anneaux

Les applications entre les structures algébriques sont très étudiées, notamment car elles permettent d'identifier quelles structures se « ressemblent » et lesquelles sont fondamentalement différentes. Ces applications sont appelées (homo)morphismes et conservent la structure algébrique des ensembles.

Définition I.1.6 (Morphisme de groupes)**.** Soient $(G, \star)$ et $(H, *)$ deux groupes. Un *morphisme de groupes* est une application $\phi : G \to H$ vérifiant

$$\forall x, y \in G, \quad \phi(x \star y) = \phi(x) * \phi(y).$$

Le *noyau* de ϕ est

$$\mathrm{Ker}(\phi) := \{x \in G : \phi(x) = 0_H\} \subset G,$$

et l'*image* de ϕ est

$$\mathrm{Im}(\phi) := \phi(G) = \{y \in H : \exists x \in G, \phi(x) = y\} \subset H.$$

Définition I.1.7 (Morphisme d'anneaux)**.** Soient $(A, +_A, \times_A)$ et $(B, +_B, \times_B)$ deux anneaux. Un *morphisme d'anneaux* est une application $\phi : A \to B$ vérifiant $\phi(1_A) = 1_B$ et

$$\forall x, y \in A, \quad \phi(x +_A y) = \phi(x) +_B \phi(y),$$

$$\forall x, y \in A, \quad \phi(x \times_A y) = \phi(x) \times_B \phi(y).$$

Exactement comme pour les morphismes de groupes, on définit le *noyau* de l'application ϕ par $\mathrm{Ker}(\phi) := \{x \in A : \phi(x) = 0_B\} \subset A$, et l'*image* de ϕ par $\mathrm{Im}(\phi) := \phi(A) \subset B$.

Notons qu'un morphisme de corps a la même définition qu'un morphisme d'anneaux. Les homomorphismes de groupes et d'anneaux se comportent bien par rapport à la bijection, c'est-à-dire que la réciproque d'un homomorphisme bijectif est aussi un homomorphisme, les deux applications s'appelent alors *isomorphismes*.

Idéaux d'anneaux

Le noyau et l'image d'un morphisme de groupes $\phi : G \to H$ sont des sous-groupes, $\mathrm{Ker}(\phi)$ de G et $\mathrm{Im}(\phi)$ de H. Il en résulte que le quotient $G/\mathrm{Ker}(\phi)$ est un groupe. En revanche, pour un morphisme d'anneaux $\psi : A \to B$, l'image $\mathrm{Im}(\psi)$ est bien un sous-anneau de B alors que $\mathrm{Ker}(\psi)$ est un idéal de A, faisant du quotient $A/\mathrm{Ker}(\psi)$ un anneau.

Définition I.1.8 (Idéal)**.** Soit A un anneau (commutatif). Un *idéal* I de A est un sous-groupe additif ayant la propriété d'*absorption* :

$$\forall a \in A, \forall x \in I, \quad ax \in I.$$

Autrement dit, I est stable sous la multiplication par des éléments de A.

Quotienter des anneaux commutatifs par certains idéaux permet de construire des anneaux intègres ou des corps.

Définition I.1.9 (Idéal premier et idéal maximal)**.** Soit A un anneau commutatif. Un idéal $\mathfrak{p} \neq A$ est *premier* si pour tous $a, b \in A$,

$$ab \in \mathfrak{p} \Rightarrow (a \in \mathfrak{p} \text{ ou } b \in \mathfrak{p}).$$

Un idéal $\mathfrak{m} \neq A$ est *maximal* si pour tout idéal I de A,

$$\mathfrak{m} \subset I \subset A \Rightarrow (I = \mathfrak{m} \text{ ou } I = A),$$

i.e. $\mathfrak{m}$ est maximal au sens de l'inclusion.

Nous constatons qu'un idéal maximal est toujours premier, ce qui est notamment illustré par le théorème suivant (car tout corps est un anneau intègre).

Théorème I.1.3. *Soit A un anneau commutatif. Un idéal I est premier si, et seulement si, le quotient A/I est un anneau intègre. Un idéal J est maximal si, et seulement si, le quotient A/J est un corps.*

L'anneau $\mathbb{Z}$ est un anneau particulier dans le sens où c'est un *anneau principal*, à savoir que tous ses idéaux sont engendrés par un unique élément : quel que soit I un idéal de $\mathbb{Z}$, il existe $a \in \mathbb{Z}$ tel que $I = a\mathbb{Z} = \langle a \rangle$. De plus, dans les anneaux principaux, il y a pour les idéaux équivalence entre être premier et être maximal : les idéaux $p\mathbb{Z}$ pour un nombre premier p sont des idéaux maximaux donc premiers et le quotient $\mathbb{Z}/p\mathbb{Z}$ est alors un corps.

I.1.2 Corps et extension de corps

Soit K un corps, *i.e.* un anneau commutatif où chaque élément non nul est inversible. Soit ϕ le morphisme d'anneaux $\mathbb{Z} \to K$ (envoyant donc $1_\mathbb{Z}$ sur 1_K). De deux choses l'une :

— soit le noyau de ϕ est $\{0_\mathbb{Z}\}$ et K contient une copie de $\mathbb{Z} \simeq \mathrm{Im}(\phi)$. Alors, K étant un corps, chaque élément de l'image (hormis $\phi(0_\mathbb{Z})$) est inversible. Le corps K contient donc un sous-corps isomorphe à $\mathbb{Q}$.

— soit le noyau de ϕ est un idéal non nul de $\mathbb{Z}$, donc $\mathrm{Ker}(\phi) = p\mathbb{Z}$ pour un entier $p \in \mathbb{Z}$. Or, $\mathbb{Z}/p\mathbb{Z}$ est isomorphe à un sous-anneau dans le corps K, c'est donc un anneau intègre et alors $p\mathbb{Z}$ est un idéal premier, *i.e.* p est premier. Le corps K contient donc un sous-corps isomorphe à $\mathbb{Z}/p\mathbb{Z}$, pour un certain p premier.

L'ordre du noyau de cette application ϕ est appelé la *caractéristique* du corps K. Nous venons de voir qu'un corps contient toujours un sous-corps isomorphe à $\mathbb{Q}$ ou $\mathbb{Z}/p\mathbb{Z}$, pour p premier. Ils sont donc les corps les plus petits possibles (au sens de l'inclusion) : ils sont appelés *corps premiers*[2].

Soient L un autre corps et $\iota : K \to L$ un morphisme d'anneaux. Il est forcément injectif, car $\iota(1_K) = 1_L$ et donc $\mathrm{Ker}(\iota) = \{0_K\}$ (le seul idéal différent de K tout entier). Ainsi, $K \simeq \iota(K) \subset L$. Le couple (L, ι) est appelé une *extension de corps* de K. En faisant l'identification entre K et son image, nous omettons le morphisme ι et notons directement L/K l'extension. Le corps L est alors un K-espace vectoriel. La dimension de L en tant que K-espace vectoriel est appelé le *degré de l'extension*, noté $[L : K] := \dim_K L$. Si $[L : K]$ est fini, l'extension L/K est dite finie.

Pour une extension L/K, un élément α de L est *algébrique* sur K s'il existe un polynôme $P(X) \in K[X]$ tel que $P(\alpha) = 0$. Le polynôme unitaire de plus petit degré annulant α est appelé son *polynôme minimal*, notons-le $\mu_\alpha(X) \in K[X]$. Le corps $K(\alpha) \subset L$ est alors isomorphe au quotient $K[X]/\langle \mu_\alpha(X) \rangle$. Le corps $K(\alpha)$ est un K-espace vectoriel de

2. En l'occurrence, il faut faire attention à $\mathbb{Q}$, c'est un corps premier mais de caractéristique nulle.

degré $[K(\alpha) : K] = \deg(\mu_\alpha)$ et la famille $(\alpha^i)_{0 \leqslant i < \deg(\mu_\alpha)}$ en est une base. Lorsque tous les éléments de L sont algébriques sur K, l'extension est dite *algébrique*. Un élément de L non-algébrique sur K est dit *transcendent* (sur K).

Théorème I.1.4. *Une extension finie est algébrique.*

Un corps K est dit *algébriquement clos* si tout polynôme à coefficients dans K admet au moins une racine dans K. Une *clôture algébrique* $\overline{K}$ d'un corps K est une extension de corps de K algébriquement close.

Théorème I.1.5. *Tout corps K admet une clôture algébrique $\overline{K}$, unique à isomorphisme près.*

Dans les corps que nous rencontrons, on définit la *trace* et la *norme* d'un élément α dans une extension finie L/K comme la trace et la norme de l'endomorphisme $x \mapsto \alpha x$ de L vu comme un K-espace vectoriel. Sur les corps finis et les corps de nombres, que nous abordons plus loin, la trace est la somme des conjugués et la norme est le produit des conjugués.

I.1.3 Plan projectif

La géométrie dans le *plan projectif* concrétise une intuition provenant du dessin : deux droites parallèles s'intersectent en l'infini (*e.g.* les points de fuite en perspective conique). Cela veut dire que le plan projectif est formé d'un plan affine classique et d'une droite, dite *à l'infini*, qui correspond à toutes les directions de droite possibles (car deux droites parallèles ont la même direction).

Une construction de ce plan projectif se fait grâce aux droites vectorielles en dimension 3. Dans l'espace vectoriel K^3, pour un corps K, deux vecteurs u et v sont colinéaires s'il existe un scalaire $\lambda \in K^\times$ tel que $u = \lambda v$. Il est alors possible de partitionner les vecteurs non nuls de K^3 par la relation d'équivalence, notée $\sim$,

$$(x_1, y_1, z_1) \sim (x_2, y_2, z_2) \Leftrightarrow \exists \lambda \in K^\times, (x_1, y_1, z_1) = (\lambda x_2, \lambda y_2, \lambda z_2). \tag{I.1.1}$$

La classe d'équivalence d'un point $(x, y, z) \in K^3$ est notée $[x : y : z]$.

Si un point (x, y, z) de K^3 est tel que $z \neq 0$ alors $[x : y : z] = [x/z : y/z : 1]$. Ainsi, en notant $\mathbb{A}^2(K)$ le plan affine, il est possible de voir le plan vectoriel d'équation $z = 1$ comme le plan affine $\mathbb{A}^2(K)$ par le plongement $(x, y) \mapsto [x : y : 1]$. Ce que nous faisons

ici, c'est choisir un représentant dans la classe d'équivalence des vecteurs dont la dernière coordonnée est non nulle.

Maintenant, pour la droite à l'infini, on regarde les droites du plan $z = 0$ et on fait la même démarche avec les droites d'équations $y \neq 0$. Pour un point $(x, y, 0)$, si $y \neq 0$ alors $[x : y : 0] = [x/y : 1 : 0]$. Comme précédemment, on identifie les points de la droite $(x, 1, 0)$ avec les droites du plan $z = 0$ de coordonnées y non nulles. Il reste alors une dernière droite, celle d'équation $y = z = 0$, qui s'identifie avec le point à l'infini $[1 : 0 : 0]$ sur la droite à l'infini $[x : 1 : 0]$, pour $x \in K$.

Lorsque l'on regarde les ensembles algébriques affines de dimension 1, *i.e.* les zéros de polynômes à deux variables, et que l'on souhaite considérer leur version projective, il faut *homogénéiser* les polynômes avec une troisième variable, pour garder une cohérence avec la définition de relation d'équivalence (I.1.1). Cela veut dire que chaque monôme doit être de même degré, quitte à multiplier par la bonne puissance de la troisième variable. Pour améliorer la lisibilité, il est courant de noter les variables affines avec des lettres minuscules (non-homogénéisées) et les variables projectives (homogénéisées) avec des majuscules. C'est ce que nous faisons dans la suite. Dans notre cas du plan, une droite affine est d'équation $ax+by+c = 0$ dans $K[x, y]$ et la droite projective correspondante est d'équation $aX + bY + cZ = 0$ dans $K[X, Y, Z]$ (chaque monôme est de degré 1). Autre exemple, une cubique d'équation affine $y^2 - x^3 - ax - b = 0$ correspond à une cubique d'équation projective $Y^2Z - X^3 - aXZ^2 - bZ^3 = 0$ (chaque monôme est de degré 3, et en l'occurrence, c'est l'équation d'une courbe elliptique).

Deux droites d'équation $d : ax + by + c = 0$ et $d' : ax + by + c' = 0$ sont parallèles (distinctes si $c \neq c'$) dans le plan affine. Elles correspondent aux droites projectives d'équation $D : aX + bY + cZ = 0$ et $D' : aX + bY + c'Z = 0$. On remarque que D et D' sont des « prolongements » projectifs des droites affines et que, dans K^3, leurs équations correspondent à des plans vectoriels qui s'intersectent en une droite le plan vectoriel d'équation $Z = 0$ (*i.e.* la droite projective à l'infini). Leur point de concours projectif est alors $[-b : a : 0]$ sur la droite à l'infini. Lorsque $a = 0$, les droites d et d' sont horizontales et elles s'intersectent alors à l'infini en $[1 : 0 : 0]$. Sinon, elles s'intersectent en $[-b/a : 1 : 0]$. Remarquons que lorsque $b = 0$, les droites sont verticales et s'intersectent en $[0 : 1 : 0]$.

Nous avons tous les éléments pour former le plan projectif $\mathbb{P}^2(K)$, sur un corps K : c'est l'ensemble des triplets (x, y, z) quotienté par la relation d'équivalence (I.1.1). La classe d'équivalence de (x, y, z) est alors notée $[X : Y : Z]$. Le plan affine, noté $\mathbb{A}^2(K)$, s'injecte dans $\mathbb{P}^2(K)$ par $(x, y) \mapsto [x : y : 1]$ et de même, l'ensemble des points projectifs $Z \neq 0$

s'identifie au plan affine par $[X : Y : Z] \mapsto (X/Z, Y/Z)$. La droite à l'infini de $\mathbb{P}^2(K)$ d'équation $Z \neq 0$ est une droite projective, notée $\mathbb{P}^1(K)$. La droite $\mathbb{P}^1(K)$ (omettons le $Z = 0$) est formée d'une droite affine par l'identification $x \mapsto [x : 1]$, de réciproque $[X : Y] \mapsto X/Y$ pour $Y \neq 0$, et d'un point à l'infini $[1 : 0]$.

Dans le plan $\mathbb{P}^2(K)$, les droites affines

— horizontales intersectent la droite à l'infini en $[1 : 0 : 0]$;

— verticales intersectent la droite à l'infini en $[0 : 1 : 0]$;

— de coefficient directeur $m \neq 0$ intersectent la droite à l'infini en $[1 : m : 0]$.

Cela permet de visualiser que la droite à l'infini est la droite des coefficients directeurs des droites affines. Une droite d'équation $y = ax + b$ est de coefficient directeur a et intersecte la droite à l'infini en $[1 : a : 0]$. La direction verticale correspondrait alors à « $a = \infty$ » et donc intersecte la droite à l'infini en « $[1/a : 1 : 0] = [0 : 1 : 0]$ ». Cela donne une justification du fait que $[0 : 0 : 0]$ n'est pas un point du plan projectif : il ne correspond à aucune direction de droite affine.

Nous voyons plus bas que les courbes elliptiques sont des variétés projectives : leur environnement ambiant est le plan projectif. Bien que nous faisons toute la théorie en affine, nous rappelons le caractère projectif des courbes lorsque nécessaire : pour définir le point à l'infini et discuter des calculs sur la courbe, car se placer dans le plan projectif permet de simplifier les fractions, *i.e.* de ne pas manipuler de dénominateurs.

I.2 Corps fini

Lorsqu'un anneau fini est sans diviseur de zéro alors il est commutatif et tous ces éléments non nuls sont inversibles : c'est le résultat du théorème de Wedderburn.

Théorème I.2.1 (Wedderburn)**.** *Tout anneau fini sans diviseur de zéro est un corps.*

Comme les corps de caractéristique nulle contiennent $\mathbb{Q}$, tous les corps finis sont de caractéristique strictement positive p première. Notons $\mathbb{F}_p$ le corps fini premier à p éléments. Ce corps $\mathbb{F}_p$ est unique, à unique isomorphisme près : $\mathbb{Z}/p\mathbb{Z}$ est un corps à p éléments et pour tout autre corps à p éléments, l'isomorphisme entre les deux est unique car les éléments neutres s'envoient l'un sur l'autre et engendrent tout le corps. Nous prenons donc directement $\mathbb{F}_p := \mathbb{Z}/p\mathbb{Z}$.

Une extension finie de $\mathbb{F}_p$ est un corps fini $\mathbb{F}_q$ à q éléments dont le degré $[\mathbb{F}_q : \mathbb{F}_p]$ est un entier n. Comme $\mathbb{F}_q$ est un $\mathbb{F}_p$-espace vectoriel de dimension n, il en résulte $\mathbb{F}_q \simeq \mathbb{F}_p^n$ en tant qu'espaces vectoriels et $q = p^n$.

Théorème I.2.2. *Pour p premier et n entier non nul, il existe un corps, unique à isomorphisme près, à p^n éléments.*

La preuve de ce théorème se fait essentiellement grâce à la proposition suivante.

Proposition I.2.1. *Pour p premier et n entier non nul, il existe un polynôme irréductible de degré n dans $\mathbb{F}_p[X]$.*

Soit $P(X) \in \mathbb{F}_p[X]$ un polynôme irréductible de degré n. L'idéal $\langle P(X)\rangle$ est maximal dans $\mathbb{F}_p[X]$, ainsi $\mathbb{F}_p[X]/\langle P(X)\rangle$ est un corps à p^n éléments. Définissons alors

$$\mathbb{F}_{p^n} := \frac{\mathbb{F}_p[X]}{\langle P(X)\rangle}.$$

Le polynôme $P(X)$ n'a pas vraiment d'importance ici, car pour un autre polynôme, le quotient serait isomorphe à $\mathbb{F}_{p^n}$ (en tant que $\mathbb{F}_p$-algèbre). C'est comme cela que nous construisons les corps finis à l'aide d'un ordinateur.

Cependant, il est souvent plus pratique de voir les corps finis comme des ensembles laissés fixes par des puissances du Frobenius dans la clôture algébrique $\overline{\mathbb{F}_p}$ de $\mathbb{F}_p$.

Définition I.2.1 (Frobenius)**.** L'application $\pi : \overline{\mathbb{F}_p} \to \overline{\mathbb{F}_p}$ telle que $\pi(x) = x^p$ pour tout $x \in \overline{\mathbb{F}_p}$ est un morphisme de corps appelé le *morphisme de Frobenius.*

Notons $\pi^n := \pi \circ \cdots \circ \pi : x \mapsto x^{p^n}$ la composée successive n-fois du Frobenius. Les sous-corps de $\overline{\mathbb{F}_p}$ à p^n éléments, pour $n > 0$ entier, sont laissés fixes par π^n. Il est donc possible de visualiser les corps finis comme

$$\mathbb{F}_{p^n} \simeq \mathrm{Ker}(\pi^n - \mathrm{id}) \subset \overline{\mathbb{F}_p}.$$

On remarque que la clôture algébrique $\overline{\mathbb{F}_p}$ est de cardinal infini (dénombrable).

Soit $P(X) \in \mathbb{F}_p[X]$ un polynôme irréductible de degré $n > 1$. Soit $\alpha \in \overline{\mathbb{F}_p}$ une racine de $P(X)$. Alors le Frobenius agit sur les racines par conjugaison, c'est-à-dire que toutes les racines de $P(X)$ sont les itérations successives du Frobenius sur la racine α :

$$\begin{aligned} P(X) &= (X-\alpha)(X-\alpha^p)(X-\alpha^{p^2})\dots(X-\alpha^{p^{n-1}}), \\ &= (X-\alpha)(X-\pi(\alpha))(X-\pi^2(\alpha))\dots(X-\pi^{n-1}(\alpha)), \end{aligned}$$

et $\pi^n(\alpha) = \alpha$, *i.e.* $\alpha \in \mathbb{F}_{p^n}$.

Nous définissons de façon classique la trace et la norme d'un élément de $\mathbb{F}_{p^n}$.

Définition I.2.2 (Trace et norme)**.** Soit $\alpha \in \mathbb{F}_{p^n}$. La trace de α sur $\mathbb{F}_p$ est

$$\mathrm{tr}(\alpha) := \sum_{i=1}^{n} \pi^i(\alpha) = \alpha + \alpha^p + \alpha^{p^2} + \cdots + \alpha^{p^{n-1}} \in \mathbb{F}_p.$$

La norme de α sur $\mathbb{F}_p$ est

$$\mathrm{norm}(\alpha) := \prod_{i=1}^{n} \pi^i(\alpha) = \alpha^{1+p+p^2+\cdots+p^{n-1}} \in \mathbb{F}_p.$$

Un théorème important de théorie des corps énonce que, pour K un corps, tout sous-groupe fini de $K^\times$ est cyclique. Ce qui a des répercutions intéressantes sur les corps finis.

Proposition I.2.2. *Soit $\mathbb{F}_{p^n}$ un corps fini à p^n éléments, où $n \geqslant 1$. Le groupe $\mathbb{F}_{p^n}^\times$ est cyclique d'ordre $p^n - 1$.*

Un élément de $\mathbb{F}_{p^n}$ générateur de $\mathbb{F}_{p^n}^\times$ est dit *primitif* et un polynôme de $\mathbb{F}_p[X]$ dont les racines sont des éléments primitifs est aussi dit primitif.

I.3 Corps de nombres quadratiques imaginaires

Un corps de nombres K est une extension finie du corps $\mathbb{Q}$. Il peut être identifié à un sous-corps de $\mathbb{C}$, nous considérons alors toujours que $\mathbb{Q} \subset K \subset \mathbb{C}$. Les corps de nombres quadratiques imaginaires jouent un rôle important dans la théorie des courbes elliptiques et sont donc (succinctement) étudiés dans cette partie.

Soit $d > 0$ un entier sans facteur carré. Le nombre imaginaire $i \in \mathbb{C}$ vérifie $i^2 = -1$. Ainsi on prend la liberté de noter $\sqrt{-d} := i\sqrt{d}$. L'ensemble $K = \mathbb{Q}(\sqrt{-d})$ est un corps qui est une extension algébrique de degré 2 sur $\mathbb{Q}$, appelé un *corps de nombres quadratiques imaginaires* :

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

Les entiers algébriques de $\mathbb{Q}$ sont les éléments de $\mathbb{Z}$, il est alors légitime de chercher si en prenant une extension finie K de $\mathbb{Q}$, on agrandit aussi l'anneau des entiers. Cet ensemble $\mathcal{O}_K$ des entiers algébriques dans K est formé des éléments de K qui sont racines d'un polynôme unitaire de $\mathbb{Z}[X]$. Avant de décrire $\mathcal{O}_K$, introduisons quelques définitions.

Soit $\alpha \in K$. On note le *conjugué* de $\alpha = a + b\sqrt{-d}$ par $\overline{\alpha} = a - b\sqrt{-d}$ (restriction de la conjugaison de $\mathbb{C}$ à K). On note classiquement $\mathrm{tr}(\alpha) = \alpha + \overline{\alpha} = 2a$ la *trace* de α et $\mathrm{norm}(\alpha) = \alpha\overline{\alpha} = a^2 + b^2 d$ sa *norme*. On constate que $\mathrm{tr}(\alpha)$ et $\mathrm{norm}(\alpha)$ sont des rationnels.

Revenons à la description de $\mathcal{O}_K$. Soit $\alpha \in \mathcal{O}_K$ racine du polynôme $X^2+uX+v \in \mathbb{Z}[X]$. En factorisant ce polynôme sur K, on obtient

$$X^2 + uX + v = (X - \alpha)(X - \overline{\alpha}) = X^2 - (\alpha + \overline{\alpha})X + \alpha\overline{\alpha}.$$

Après identification des coefficients, $\mathrm{tr}(\alpha) = -u$ et $\mathrm{norm}(\alpha) = v$. Ainsi, un élément $\alpha \in K$ est un entier de $\mathcal{O}_K$ si, et seulement si, $\mathrm{tr}(\alpha) \in \mathbb{Z}$ et $\mathrm{norm}(\alpha) \in \mathbb{Z}$.

Notons $\alpha = a + b\sqrt{-d}$. Dire $\alpha \in \mathcal{O}_K$ signifie que $2a \in \mathbb{Z}$ et donc que a est un demi-entier. De deux choses l'une : soit $a \in \mathbb{Z}$, soit $a = c/2$ pour un entier c impair. Pour déterminer à quel ensemble appartient b, on s'aide de $a^2 + b^2 d = v \in \mathbb{Z}$.

— Dans le premier cas, $a \in \mathbb{Z}$ et $b^2 d = v - a^2$ est un entier, impliquant $b \in \mathbb{Z}$ car d est entier sans facteur carré.

— Dans le second cas, $a = c/2$ et $(2b)^2 d = 4v - c^2$ réduit modulo 4 implique $d \equiv 3 \pmod 4$ et $2b$ impair. Autrement dit

$$\alpha = \frac{a}{2} + \frac{b}{2}\sqrt{-d}$$

avec $a, b \in \mathbb{Z}$ impairs, ce qui se réécrit

$$\alpha = \frac{a-b}{2} + \frac{1+\sqrt{-d}}{2}b.$$

Réciproquement, si $\alpha = a + b\sqrt{-d}$ avec a et b entiers, alors $\mathrm{norm}(\alpha) = a^2 + b^2 d \in \mathbb{Z}$ et $\mathrm{tr}(\alpha) = 2a \in \mathbb{Z}$. Si $\alpha = a + b(1 + \sqrt{-d})/2$ avec a et b entiers, alors $\mathrm{tr}(\alpha) = 2a + b \in \mathbb{Z}$ et

$$\begin{aligned}\mathrm{norm}(\alpha) &= \left(a + \frac{b}{2}\right)^2 + \frac{b^2}{4}d \\ &= a^2 + ab + b^2\left(\frac{d+1}{4}\right)\end{aligned}$$

est entier si, et seulement si, $d \equiv 3 \pmod 4$. L'anneau des entiers de K est donc

$$\mathcal{O}_K = \begin{cases} \mathbb{Z}[\sqrt{-d}] & \text{si } d \equiv 1, 2 \pmod 4, \\ \mathbb{Z}[(1+\sqrt{-d})/2] & \text{si } d \equiv 3 \pmod 4. \end{cases}$$

Ordre, conducteur et discriminant

Un *ordre* dans $K = \mathbb{Q}(\sqrt{-d})$ est un anneau $\mathcal{O}$ tel que $\mathbb{Z} \subsetneq \mathcal{O} \subseteq \mathcal{O}_K$. C'est un groupe abélien de type fini de la forme

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

où $f > 0$ est un entier et $\delta = (1 + \sqrt{-d})/2$ si $d \equiv 3 \pmod 4$ ou $\sqrt{-d}$ si $d \equiv 1, 2 \pmod 4$. L'entier f est appelé le *conducteur* de $\mathcal{O}$ et correspond à l'indice $[\mathcal{O}_K : \mathcal{O}]$. Le *discriminant* de $\mathcal{O}$ est

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

qui correspond au discriminant du polynôme quadratique de racine $f\delta$.

Exemple : le corps $\mathbb{Q}(\sqrt{-3}) = \mathbb{Q}(\zeta_3)$

Soit $\zeta_3 \in \mathbb{C}$ une racine primitive troisième de l'unité. Son polynôme minimal est $\Phi_3(X) = X^2 + X + 1$, le troisième polynôme cyclotomique. L'extension $\mathbb{Q}(\zeta_3)$ est donc degré 2 sur $\mathbb{Q}$. Grâce aux formules de calcul de racines de polynôme du second degré, on obtient $\zeta_3 = (1 \pm \sqrt{-3})/2$. Ainsi $\zeta_3 \in \mathbb{Q}(\sqrt{-3})$. De même, $1 + 2\zeta_3 = \pm\sqrt{-3}$ et $\sqrt{-3} \in \mathbb{Q}(\zeta_3)$.

On a donc montré que $\mathbb{Q}(\sqrt{-3}) = \mathbb{Q}(\zeta_3)$ et, plus généralement, que si un corps de nombre K contient une racine primitive troisième de l'unité, alors K contient aussi $\pm\sqrt{-3}$, et inversement.

I.4 Problème du logarithme discret

Nous avons déjà abordé le problème du calcul du logarithme discret (DLP [3]) dans des groupes cycliques. Rappelons-le : soient G un groupe cyclique de générateur g et $h \in G$, le problème du logarithme discret est, sachant g et h, de trouver $x \in \mathbb{Z}$ tel que $g^x = h$, qui est alors noté $x = \log_g(h)$.

Remarque I.4.1. Le logarithme discret n'est pas difficile à calculer dans tous les groupes. Par exemple, le groupe $(\mathbb{Z}/n\mathbb{Z}, +)$ est cyclique et calculer un logarithme discret s'y fait grâce à l'algorithme d'Euclide étendu en $O(\log(n)^2)$.

3. *Discrete Logarithm Problem* en anglais.

Un ensemble ne possédant que la structure de groupe est dit *groupe générique* [Sho97]. Cela signifie entre autre que l'on ne peut avoir recours qu'à l'inversion et la multiplication d'éléments du groupe. Il est aussi possible de comparer les éléments entre eux afin de tester l'égalité. Shoup a prouvé [Sho97] que, dans ces conditions, si $p = \#G$ est premier alors le logarithme discret ne pouvait être calculé qu'en $O(\sqrt{p})$: il n'est pas possible de faire mieux que les algorithmes pas de bébés, pas de géants de Shanks [Sha71] (de complexité $O(\sqrt{p}\log p)$) ou ρ (rho) de Pollard [Pol78] (de complexité $O(\sqrt{p})$).

Nous décrivons dans la suite l'algorithme attribué à Shanks, la réduction Pohlig-Hellman, et discutons aussi du cas des logarithmes discrets dans les corps finis.

Pas de bébés, pas de géants

Dans un groupe générique G engendré par g, l'algorithme pas de bébés, pas de géants [Sha71] procède comme suit pour trouver $x = \log_g(h)$.

Notons $m := \lceil\sqrt{\#G}\rceil$. Par division euclidienne de x par m, il existe q et r des entiers positifs inférieurs à m tels que $x = qm + r$. À savoir,

$$h = g^x \Leftrightarrow hg^{-r} = (g^m)^q.$$

Ainsi, trouver x est équivalent à trouver la paire (q, r). Il est alors possible de stocker dans une liste l'ensemble des (hg^{-r}, r) pour $0 \leqslant r < m$, dits « pas de bébés ». De même, une seconde liste est générée à partir de l'ensemble des $((g^m)^q, q)$ pour $0 \leqslant q < m$, dits « pas de géants ». Il reste alors à trier et comparer ces listes : une collision sur la première coordonnée donne (q, r) tel que $qm + r$ soit le logarithme discret de h en base g.

La complexité du tri des listes est en $O(m \log m)$, celle de la comparaison est négligeable et celle de la génération des listes est en $O(m)$. Ainsi, cet algorithme a une complexité temporelle $O(m \log m)$ et spatiale de $O(m)$.

En pratique, cet algorithme n'est pas utilisé car il est assez gourmand en espace de stockage. C'est pour cela qu'on lui préfère ρ de Pollard, qui est de complexité temporelle $O(\sqrt{\#G})$ mais n'utilise que peu de mémoire (il n'a besoin que d'un nombre constant de registres). Nous ne le détaillons pas ici mais pointons vers [Pol78 ; Pie16].

Pohlig-Hellman

La méthode Pohlig-Hellman [PH78] permet de réduire le problème du logarithme discret d'un groupe G d'ordre n vers plusieurs groupes d'ordre plus petit. Supposons que la factorisation de n soit connue, *i.e.* on connaît les premiers p_i, deux à deux distincts, et les exposants $e_i > 0$, pour $1 \leqslant i \leqslant k$, tels que

$$n = \prod_{i=1}^{k} p_i^{e_i}.$$

Maintenant, si g est un générateur de G et que l'on cherche x tel que $g^x = h$ pour h donné, alors il est possible de calculer $x \pmod{p_i^{e_i}}$ pour tout $1 \leqslant i \leqslant k$, puis de recouvrer $x \pmod{n}$ grâce au théorème des restes chinois. On se retrouve ainsi à calculer des logarithmes discrets dans des groupes d'ordre ${p_i}^{e_i}$ engendrés par $g^{n/p_i^{e_i}} = g^{\prod_{j \neq i} p_j^{e_j}}$.

$$\begin{array}{c} h = g^x \\ n = p_1^{e_1} \times \cdots \times p_k^{e_k} \\ \swarrow \quad \text{Pohlig-Hellman} \quad \searrow \\ x \pmod{p_1^{e_1}} \quad \ldots \quad x \pmod{p_k^{e_k}} \\ \searrow \quad \text{CRT} \quad \swarrow \\ x \pmod{n} \end{array}$$

On peut passer modulo les p_i avec un argument similaire, et « remonter » la solution modulo $p_i^{e_i}$.

En supposant l'utilisation d'algorithmes génériques pour calculer les logarithmes discrets dans les sous-groupes, la complexité totale est $O(\sum_i e_i(\sqrt{p_i} + \log n))$ [MvV97]. Cela implique qu'il est préférable de choisir des groupes cycliques d'ordre premier p car dans, ce cas là, la représentation des éléments est de taille $O(\log p)$ et le calcul du logarithme discret y coûte $O(\sqrt{p})$, ce qui est optimal pour les groupes génériques.

Logarithme discret dans un corps fini

La résolution du DLP dans le groupe multiplicatif $\mathbb{F}_{p^n}^{\times}$ d'un corps fini est rendue plus facile grâce à la structure de corps, en l'occurrence $\mathbb{F}_{p^n}^{\times}$ n'est pas un groupe générique.

Les logarithmes discrets de $\mathbb{F}_{p^n}$ peuvent être calculés en temps sous-exponentiel (et parfois même en temps quasi-polynomial) par des algorithmes de crible par corps de

nombre (NFS [4]). Dans le cas général, il est difficile d'évaluer la complexité de l'algorithme NFS et de ses variants (voir [GM17] pour plus de détails). Pour se donner une idée de la complexité temporelle des algorithmes NFS, on introduit la *notation* L, définie par

$$L_{p^n}[\alpha, c] := \exp\Big((c + o(1))\,(\ln p^n)^{\alpha}(\ln\ln p^n)^{1-\alpha}\Big) \tag{I.4.2}$$

où $\alpha \in [0, 1]$ et $c > 0$. Intuitivement, si $\alpha = 1$ alors $L_{p^n}[\alpha, c]$ est exponentiel en $\log_2(p^n)$ alors qu'il est polynomial en $\log_2(p^n)$ si $\alpha = 0$.

Dans notre étude, les algorithmes NFS sont de complexité asymptotique $L_{p^n}[1/3]$, mais nous y reviendrons plus loin. En attendant, diverses illustrations et explications plus détaillées sont disponibles dans la thèse de Cécile Pierrot [Pie16].

Conclusion du premier chapitre

Nous avons vu que les groupes cycliques sont les candidats adéquats pour faire de la cryptographie, dès que le logarithme discret y est difficile à calculer. En l'occurrence, le DLP est résoluble dans les groupes génériques d'ordre premier p en complexité $O(\sqrt{p})$ et en temps sous-exponentiel pour les corps finis $\mathbb{F}_{p^n}$, grâce aux algorithmes de la famille NFS. Dans le prochain chapitre, nous définissons les courbes elliptiques dont les points rationnels sont justement supposés former un groupe générique. Elles sont donc, a priori, des instanciations difficiles pour le DLP. Nous serons aussi amenés à construire des courbes elliptiques via leur anneau d'endomorphismes, qui est isomorphe à un ordre $\mathcal{O}$ d'un corps de nombres quadratiques imaginaires $\mathbb{Q}(\sqrt{-d})$.

4. *Number Field Sieve* en anglais.

Chapitre II

Courbes elliptiques et couplages

L'utilisation des courbes elliptiques dans la cryptographie a été introduite indépendamment par Victor Miller [Mil86] et Niel Koblitz [Kob87 ; Kob89 ; Kob90] en 1985. Les points sur une courbe elliptique forment un groupe où le logarithme discret est difficile à calculer : à l'heure actuelle, seuls les algorithmes génériques de résolution du DLP fonctionnent. Dans ce chapitre, nous définissons les courbes elliptiques, détaillons leur structure de groupe et nous attardons sur celles définies sur un corps fini. Ensuite, nous voyons un outil cryptographique important : le couplage. Nous décrivons aussi des méthodes de construction de courbes à couplage et étudions leur sécurité.

Nombre de résultats fondamentaux sur les courbes elliptiques sont développés dans le livre de Silverman [Sil09], et pour la partie concernant les couplages, le Chapitre IX de [BSS05] rédigé par Galbraith offre une bonne introduction agrémentée d'exemples. Le Chapitre 2 de la thèse de Costello [Cos12] est une source d'explications claires et illustrées de la construction de couplages sur courbe elliptique. Ce chapitre est fortement inspiré de ces références et le lecteur est invité à les consulter.

II.1 Courbes elliptiques

Dans cette section, nous introduisons les courbes elliptiques et plus précisément leur utilisation en cryptographie.

Une courbe elliptique est une variété algébrique de dimension 1. Cette définition n'est pas très parlante et surtout ne donne aucune indication sur l'application au domaine de la cryptographie. Donnons-en une plus pragmatique.

Dans tout le reste du manuscrit, les corps sont soit finis, soit de caractéristique nulle [1]. Soient deux éléments a et b d'un corps K de caractéristique ni 2, ni 3. Soit $\overline{K}$ la clôture algébrique de K. Considérons l'ensemble E/K (parfois simplement noté E) des couples

1. En théorie, un corps parfait est nécessaire pour travailler sur des courbes algébriques et les corps de caractéristique nulle et les corps finis sont des corps parfaits.

$(x, y) \in \overline{K} \times \overline{K}$ du plan affine $\mathbb{A}(\overline{K})$ tels que

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

adjoint d'un point dit *à l'infini*, noté P_∞, qui n'est pas visible dans le plan affine. La courbe plane E est dite *elliptique* si elle est sans singularité. Une *singularité* de E est un point $(x_0, y_0) \in E$ annulant simultanément les dérivées partielles de la fonction

$$f : (x, y) \mapsto y^2 - x^3 - ax - b.$$

En calculant ces dérivées, nous obtenons le système d'équations suivant :

$$\begin{cases} y_0^2 = x_0^3 + ax_0 + b, \\ \frac{\partial f}{\partial x}(x_0, y_0) = 3x_0^2 + a = 0, \\ \frac{\partial f}{\partial y}(x_0, y_0) = 2y_0 = 0. \end{cases} \tag{II.1.2}$$

Comme $\frac{\partial f}{\partial y}(x_0, y_0) = 0$, l'ordonnée y_0 est nulle. Ainsi, l'abscisse x_0 est racine du polynôme $X^3 + aX + b \in K[X]$ et de sa dérivée $3X^2 + a$. Cela signifie que le polynôme $X^3 + aX + b$ a une racine de multiplicité au moins 2 et que son discriminant $-4a^3 - 27b^2$ est nul. Ainsi une courbe définie par une cubique est elliptique si, et seulement si, $4a^3 + 27b^2$ est non nul dans K.

Définition II.1.1 (Discriminant)**.** Soit K un corps de caractéristique différente de 2 et 3. Le *discriminant* d'une courbe $E/K : y^2 = x^3 + ax + b$ est la quantité $\Delta(E)$ définie par

$$\Delta(E) := -16(4a^3 + 27b^2).$$

Une courbe E est elliptique si, et seulement si, $\Delta(E) \neq 0$.

Points L-rationnels

Une courbe elliptique E/K est un sous-ensemble de $\overline{K} \times \overline{K}$. Il est cependant possible de considérer les points de la courbe qui sont dans une certaine extension L, à savoir $K \subseteq L \subseteq \overline{K}$. Cet ensemble, noté $E(L)$, est l'ensemble des *points L-rationnels* :

$$E(L) := \{(x, y) \in L^2 : y^2 = x^3 + ax + b\} = E \cap L^2.$$

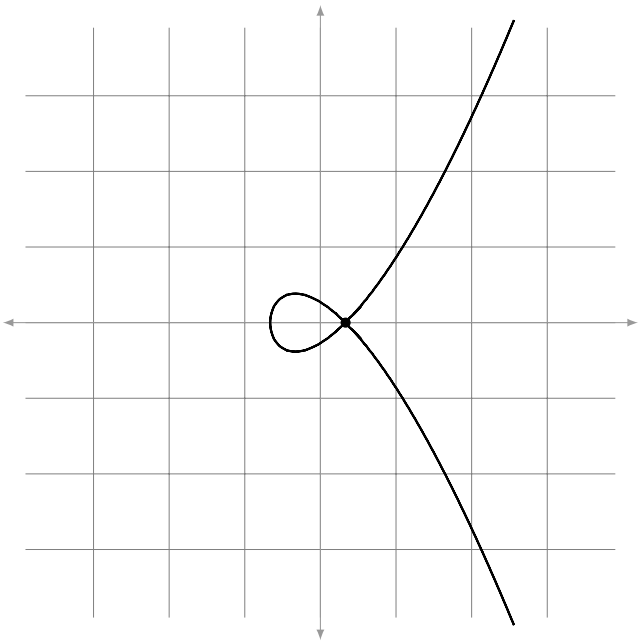

FIGURE II.1 – Courbe d'équation $y^2 = x^3 - x/3 + 2/27$ sur $\mathbb{R}$ avec un point de multiplicité 2 en $(1/3, 0)$. La courbe est une cubique nodale.

Exemple II.1.1 (Courbes singulières et elliptiques sur $\mathbb{R}$)**.** La courbe sur $\mathbb{R}$ définie par $y^2 = x^3 - x/3 + 2/27$, tracée en Figure II.1, admet une singularité en $P = (1/3, 0)$. Des équations (II.1.2), la dérivée partielle $\frac{\partial f}{\partial y}$ en P est clairement nulle et l'autre dérivée partielle l'est aussi :

$$\frac{\partial f}{\partial x}(P) = 3\left(\frac{1}{3}\right)^2 - \frac{1}{3} = 0.$$

Cette courbe en Figure II.1 est dite *nodale* (le point P est un *nœud* ou un *point multiple* : au moins deux droites distinctes sont tangentes à la courbe en P), elle n'est pas elliptique.

La courbe sur $\mathbb{R}$ définie par $y^2 = x^3$, tracée en Figure II.2, admet une singularité en $Q = (0, 0)$. Les dérivées partielles s'y annulent trivialement. Cette courbe II.2 est dite *cuspidale* (en l'occurrence, le point Q est un *point de rebroussement de première espèce* : la courbe passe à travers sa « tangente » au point de singularité Q et la matrice hessienne au point Q est de déterminant nul) et, elle non plus, n'est pas elliptique.

Les quatre courbes de la Figure II.3, définies par $E : y^2 = x^3 + ax + b$ sur $\mathbb{R}$ avec $a \in \{-2, 1\}$ et $b \in \{0, 2\}$ sont elliptiques. Remarquons d'ailleurs que la courbe elliptique $E(\mathbb{R})$ a deux composantes connexes lorsque $X^3 + aX + b$ a trois racines réelles, comme celle d'équation $y^2 = x^3 - 2x = x(x - \sqrt{2})(x + \sqrt{2})$, et la courbe $E(\mathbb{R})$ est connexe lorsque

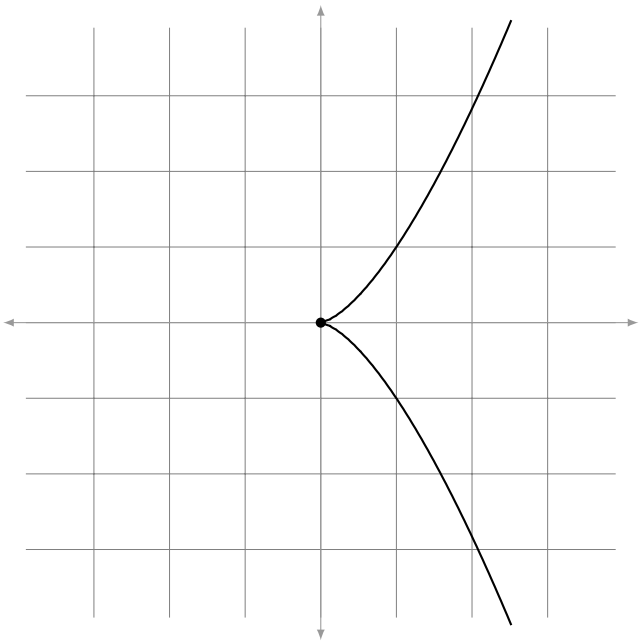

FIGURE II.2 – Courbe d'équation $y^2 = x^3$ sur $\mathbb{R}$ avec un point de rebroussement de première espèce en $(0, 0)$. La courbe est une cubique cuspidale.

le polynôme n'a qu'une seule racine réelle.

Équations de Weierstrass

L'équation (II.1.1), rappelons-la : $y^2 = x^3 + ax + b$, est dite équation de *Weierstrass courte*, ou modèle de Weierstrass court. En effet, toute courbe elliptique, sur un corps de caractéristique nulle ou première à 6, peut s'écrire avec une équation de Weierstrass courte. Lorsque la caractéristique est 2 ou 3, alors les courbes s'écrivent en utilisant le modèle de *Weierstrass long* :

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{II.1.3}$$

où les a_i sont des éléments de K. Si la caractéristique de K n'est ni 2, ni 3, alors il est toujours possible de transformer l'équation longue (II.1.3) en équation courte (II.1.1) à l'aide de transformations affines. Montrons comment. D'abord en complétant le carré du membre gauche de l'équation (II.1.3) :

$$\left(y + \frac{a_1x + a_3}{2}\right)^2 = x^3 + \left(a_2 + \frac{a_1^2}{4}\right)x^2 + \left(a_4 + \frac{a_1a_3}{2}\right)x + a_6 + \frac{a_3^2}{4}.$$

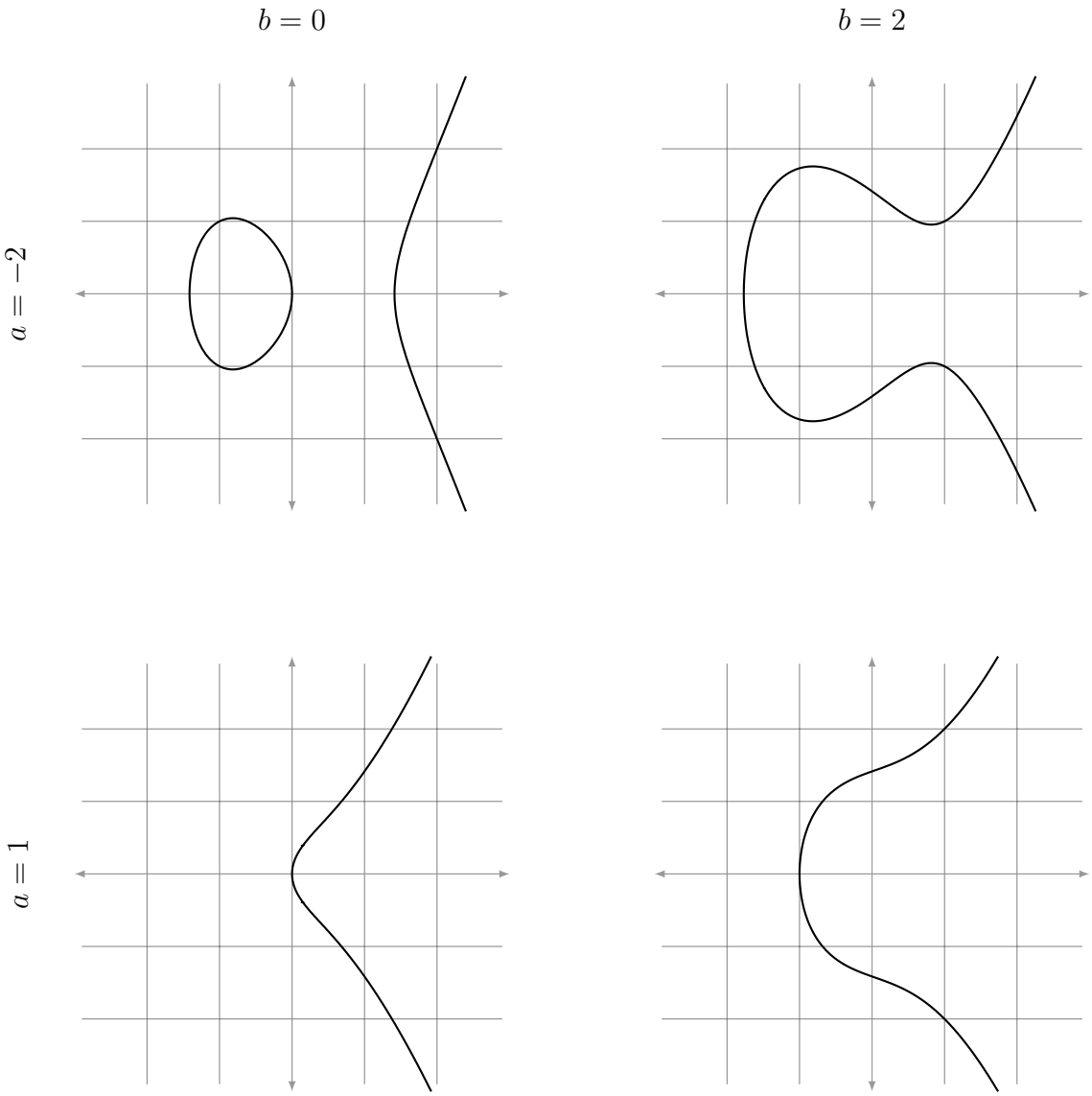

FIGURE II.3 – Courbes elliptiques d'équation $y^2 = x^3 + ax + b$ sur $\mathbb{R}$ avec $a \in \{-2, 1\}$ et $b \in \{0, 2\}$. La courbe d'équation $y^2 = x^3 - 2x$ à deux composantes connexes sur $\mathbb{R}$ alors que les autres n'en ont qu'une.

Puis, nous effectuons le changement de variable $(x, y) \mapsto (x, y+(a_1x+a_3)/2)$. Les nouvelles coordonnées sont alors sur la courbe d'équation

$$y^2 = x^3 + b_2x^2 + b_4x + b_6, \tag{II.1.4}$$

où $b_2 = a_2 + a_1^2/4$, $b_4 = a_4 + a_1a_3/2$ et $b_6 = a_6 + a_3^2/4$. Il reste maintenant à compléter le cube du membre de droite de l'équation (II.1.4) :

$$\begin{aligned} y^2 &= \left(x + \frac{b_2}{3}\right)^3 - \frac{b_2^2}{3}x - \frac{b_2^3}{27} + b_4x + b_6, \\ &= \left(x + \frac{b_2}{3}\right)^3 + \left(b_4 - \frac{b_2^2}{3}\right)\left(x + \frac{b_2}{3}\right) + b_6 + \frac{2b_2^3}{27} - \frac{b_2b_4}{3}. \end{aligned}$$

Le changement de variable $(x, y) \mapsto (x + b_2/3, y)$ envoie les nouvelles coordonnées sur une courbe d'équation $y^2 = x^3 + ax + b$ où $a = b_4 - b_2^2/3$ et $b = b_6 + 2b_2^3/27 - b_2b_4/3$.

Nous ne sommes pas intéressés par les courbes elliptiques sur les corps binaires ou ternaires (la justification de ce choix est donnée plus loin), ainsi nous considérons que K est de caractéristique différente de 2 et de 3 et que toutes les courbes elliptiques sur K sont d'équation de Weierstrass courte (II.1.1).

Exemple II.1.2 (Changement d'équation)**.** Soit $E/\mathbb{R}$ la courbe d'équation

$$E : y^2 + 2xy - y = x^3 + 2x^2 + x + 3,$$

représentée en Figure II.4 (graphe du haut). La courbe E de modèle de Weierstrass long admet une équation courte. Notons E_0 cette courbe d'équation de Weierstrass courte, représentée en Figure II.4 (graphe du bas). Le point de coordonnées (x, y) de E correspond au point $(x_0, y_0) = (x + 1, y + x - 1/2)$ sur la courbe E_0 d'équation

$$E_0 : y^2 = x^3 - 3x + 21/5.$$

L'application affine envoyant E sur E_0 transforme le repère formé des droites $x = -1$ (bleue) et $y = -x + 1/2$ (rouge) en le repère orthogonal « classique » ($x = 0$ et $y = 0$), tout en « redressant » la courbe.

Nous verrons plus tard la notion d'isomorphisme de courbes elliptiques : ce que nous faisons ici est choisir une courbe d'équation de Weierstrass courte dans la classe d'isomorphisme de E.

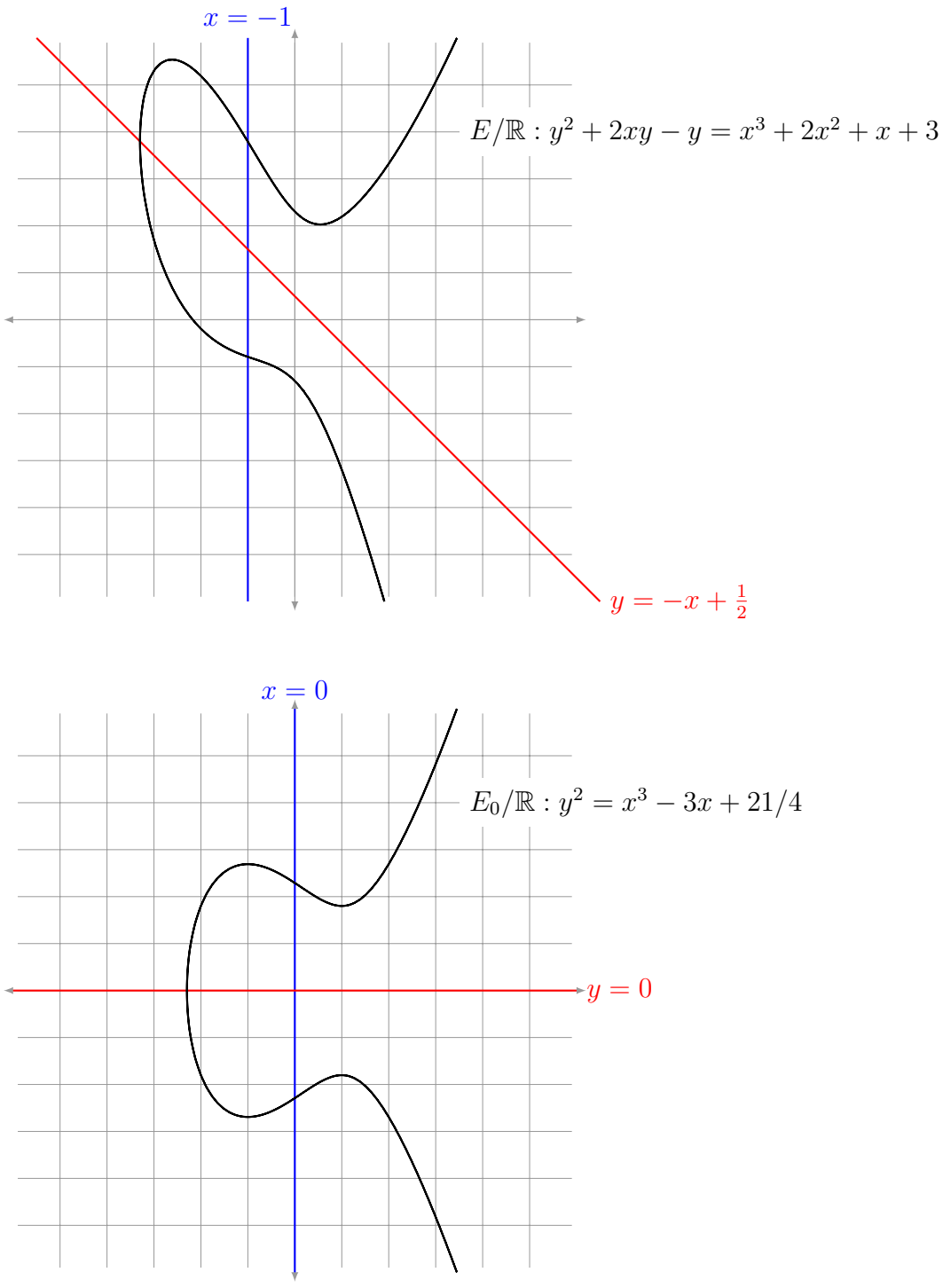

FIGURE II.4 – Transformation de la courbe d'équation $y^2+2xy-y = x^3+2x^2+x+3$ (en haut) en la courbe d'équation $y^2 = x^3 - 3x + 21/4$ (en bas). Le repère formé des droites rouge et bleue subit la même transformation que la courbe.

Point à l'infini

La visualisation du point P_∞ n'est pas chose aisée. Pour cela, il faut considérer la courbe elliptique dans le plan projectif en homogénéisant son équation :

$$Y^2Z = X^3 + aXZ^2 + bZ^3. \tag{II.1.5}$$

Un point affine (x, y) satisfaisant $y^2 = x^3 + ax + b$ est alors identifié au point projectif $[x : y : 1]$. Pour les points à l'infini, il suffit d'intersecter la courbe d'équation (II.1.5) avec la droite à l'infini d'équation $Z = 0$. Cela implique $X = 0$ et ainsi le seul point à l'infini est $[0 : 1 : 0]$. C'est ce point $P_\infty := [0 : 1 : 0]$ que l'on nomme *point à l'infini* de la courbe. Toutes les droites verticales (parallèles, d'équation affine $x = x_0$ pour $x_0 \in \overline{K}$) s'intersectent au point à l'infini P_∞, comme illustré par la Figure II.5 et indiqué dans le préambule Sous-Section I.1.3.

II.1.1 Loi de groupe

L'ensemble des points d'une courbe elliptique E est muni d'une loi de groupe dont P_∞ est l'élément neutre. Cette loi peut être très simplement décrite de façon géométrique sur $\mathbb{R}$ par « la somme de trois points alignés sur une droite est le point à l'infini P_∞ ». De cette description nous déduisons déjà que le groupe E est commutatif et il est donc noté de façon additive.

Élément neutre

L'élément neutre du groupe est le point à l'infini P_∞.

Opposé d'un point

Soit $P \in E$ un point de la courbe. Nous remarquons que le symétrique de P, notons ce point Q, image par la symétrie d'axe les abscisses, est aussi sur la droite verticale passant par P. Le troisième point est P_∞ car toute droite verticale contient ce point. En effet, l'équation projective d'une droite verticale est $X = mZ$ pour $m \in \overline{K}$: clairement $[0 : 1 : 0]$ est sur cette droite.

Ainsi, les trois points P, Q et P_∞ sont alignés, leur somme est l'élément neutre P_∞. En simplifiant cette somme, nous obtenons $P + Q = P_\infty$, *i.e.* l'opposé d'un point est son

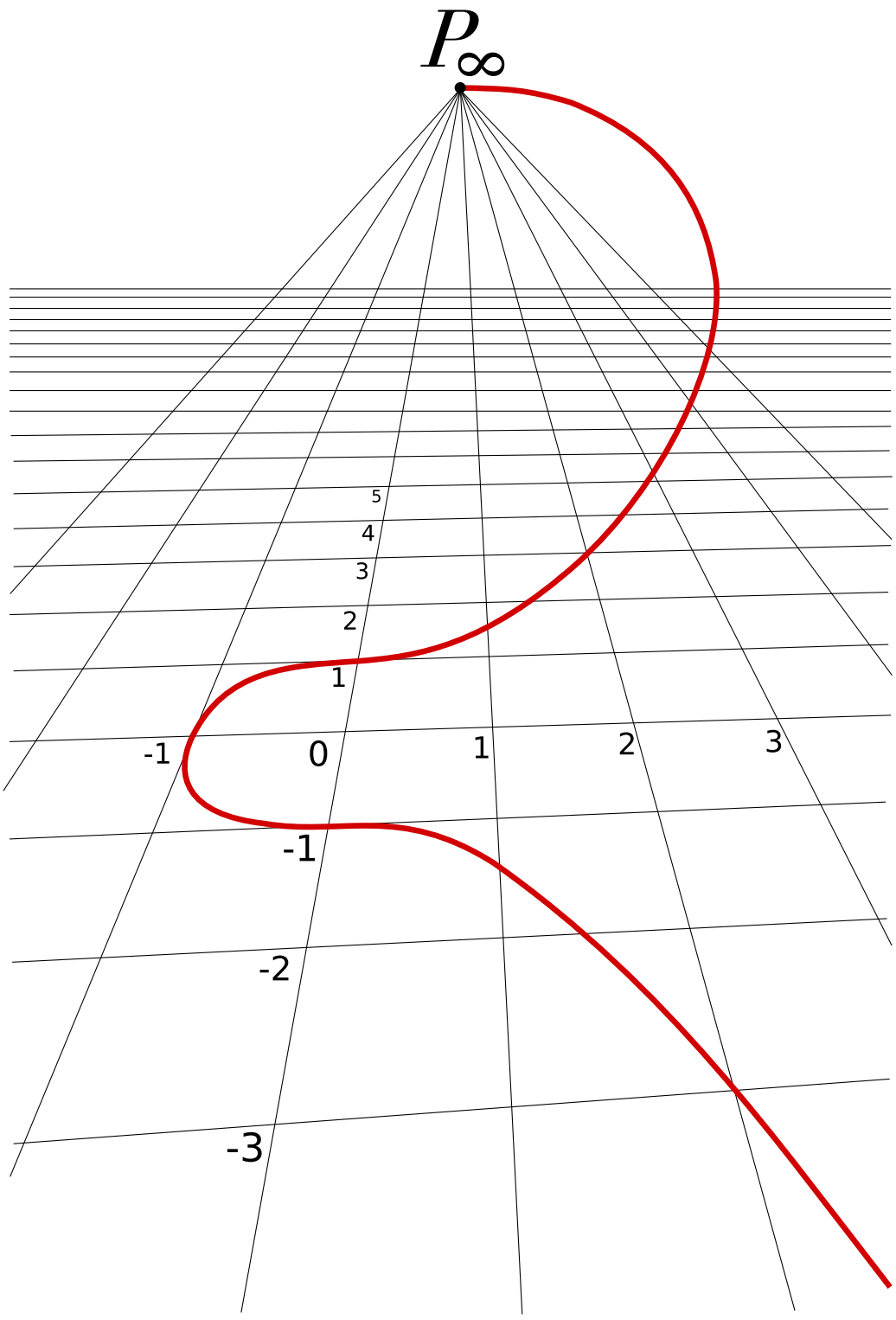

FIGURE II.5 – Illustration projective d'une courbe sur $\mathbb{R}$ d'équation $y^2 = x^3 + 1$. Le point P_∞ est le point d'intersection de la courbe elliptique avec toutes les droites verticales. Il a pour coordonnées projectives $[0 : 1 : 0]$.

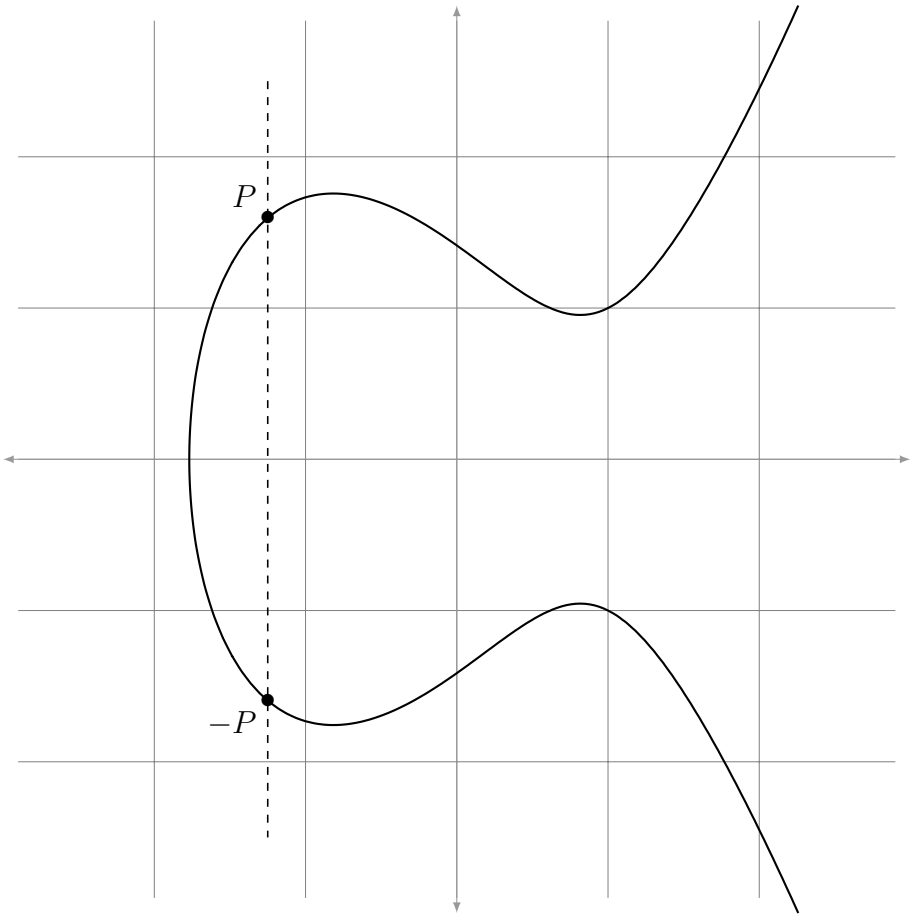

FIGURE II.6 – Un point affine P et son opposé $-P$ sur une courbe elliptique.

symétrique par rapport à l'axe des abscisses et, si $P = (x, y)$ alors $-P = (x, -y)$, comme illustré Figure II.6.

Somme de deux points distincts

Soient $P \in E$ et $Q \in E$ deux points distincts, supposons-les différents de P_∞. Si $Q = -P$ alors leur somme est P_∞. Maintenant, si $Q \neq -P$, la droite passant par les points P et Q intersecte la courbe E en un troisième point R. Ces trois points étant alignés, leur somme est $P + Q + R = P_\infty$, ce qui implique $P + Q = -R$.

Cela se traduit en équations : notons $P = (x_P, y_P)$ et $Q = (x_Q, y_Q)$. La droite passant par P et Q est de coefficient directeur

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

et d'ordonnée à l'origine $\nu = y_P - \lambda x_P$. Pour déterminer les abscisses des points d'intersection entre la courbe E, d'équation $y^2 = x^3 + ax + b$, et la droite, d'équation $y = \lambda x + \nu$, il suffit de factoriser le polynôme $x^3 + ax + b - (\lambda x + \nu)^2$, dont nous connaissons déjà deux racines, *i.e.* x_P et x_Q, et dont nous notons x_0 la troisième :

$$\begin{aligned}(x - x_P)(x - x_Q)(x - x_0) &= x^3 + ax + b - (\lambda x + \nu)^2 \\ &= x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + b - \nu^2.\end{aligned}$$

En identifiant les coefficients du terme en x^2, nous obtenons $x_P + x_Q + x_0 = \lambda^2$. Ainsi, l'abscisse de $P + Q$ est $x_0 = \lambda^2 - x_P - x_Q$ et il en découle que l'ordonnée de $P + Q$ est $y_0 = -(\lambda x_0 + \nu)$, ordonnée de l'opposé du troisième point sur la droite d'équation $y = \lambda x + \nu$.

Dans la littérature, cette construction géométrique de l'addition de deux points distincts s'appelle la *règle de la corde*, dont une illustration est donnée en Figure II.7.

Doublement d'un point

Soit $P \in E$ distinct de P_∞. Pour calculer $P + P$, traçons la tangente à E en P, cette droite intersecte un second point sur la courbe, notons le [2] Q. Le point P est de multiplicité 2, car il appartient à la courbe E et à sa tangente au point P. Ainsi $P + P + Q = P_\infty$, ce qui signifie $P + P = -Q$.

Il est possible, ici aussi, de traduire le doublement de $P = (x_P, y_P)$ en équations. La tangente de E au point P est d'équation

$$\left(\frac{\partial f}{\partial x}(x_P, y_P)\right)(x - x_P) + \left(\frac{\partial f}{\partial y}(x_P, y_P)\right)(y - y_P) = 0,$$

si E est vu comme le lieu des zéros de la fonction $f : (x, y) \mapsto y^2 - x^3 - ax - b$. D'une part $\frac{\partial f}{\partial x}(x_P, y_P) = -3x_P^2 - a$ et d'autre part $\frac{\partial f}{\partial y}(x_P, y_P) = 2y_P$. Ainsi, si on suppose un instant $y_P \neq 0$, la droite tangente à la courbe en P est d'équation

$$y = \frac{3x_P^2 + a}{2y_P}x - \frac{3x_P^2 + a}{2y_P}x_P + y_P,$$

ou bien $y = \lambda x + \nu$, en notant $\lambda = (3x_P^2 + a)/(2y_P)$ et $\nu = y_P - \lambda x_P$. Comme pour la somme de points distincts, nous substituons l'équation de la droite dans celle de la

2. Dans le cas où P est un point d'ordre 3, *i.e.* $[3]P = P_\infty$, le point Q est en fait le point P.

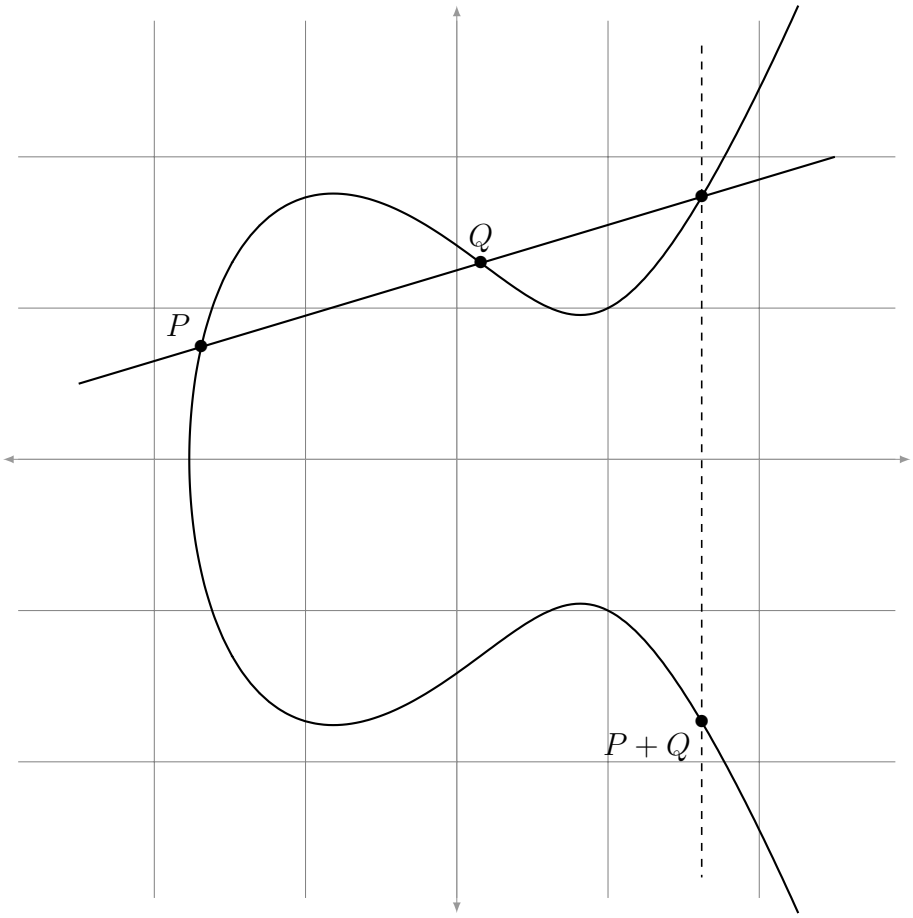

FIGURE II.7 – Deux points affines distincts P et Q ainsi que leur somme $P + Q$.

courbe, sachant qu'ici l'abscisse du point P est de multiplicité 2. En effet, en remarquant $2\lambda y_P = 3x_P^2 + a$, l'abscisse x_P est racine du polynôme dérivé $\frac{\mathrm{d}}{\mathrm{d}x}(x^3 + ax + b - (\lambda x + \nu)^2)$:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}x}[x^3 + ax + b - (\lambda x + \nu)^2](x_P, y_P) &= 3x_P^2 - 2\lambda^2 x_P + a - 2\lambda\nu \\ &= 3x_P^2 + a - 2\lambda(\lambda x_P + \nu) \\ &= 3x_P^2 + a - 2\lambda y_P \\ &= 0.\end{aligned}$$

En notant x_0 l'autre racine du polynôme $x^3 + ax + b - (\lambda x + \nu)^2 = (x - x_P)^2(x - x_0)$ et en identifiant les coefficients des polynômes, l'abscisse de la somme $P + P$ est $x_0 = \lambda^2 - 2x_P$. Il en découle que le point $P + P$ est d'ordonnée $y_0 = -(\lambda x_0 + \nu)$. Dans le cas où y_P est nul, le point P est son propre inverse (la tangente est verticale en P), donc $P + P = P_\infty$.

Dans la littérature, cette construction géométrique du doublement d'un point s'appelle la *règle de la tangente*, dont une illustration est donnée en Figure II.8.

Algorithme additionnant deux points

Résumons par l'Algorithme 1 le calcul de la somme de deux points sur la courbe elliptique $E/K : y^2 = x^3 + ax + b$. Si l'un des deux points est l'infini alors la somme vaut simplement l'autre point. Si les points sont opposés l'un de l'autre, leur somme est nulle, *i.e.* P_∞. Remarquons qu'un point d'ordre deux est son propre inverse et que son abscisse est alors nécessairement nulle. Dans les autres cas, on trace la droite passant par les deux points s'ils sont distincts, ou la tangente sinon, et la somme est alors l'opposé de la dernière intersection avec la courbe.

II.1.2 Corps de fonctions et applications entre courbes

Une courbe elliptique E sur K peut être vue comme l'ensemble des zéros du polynôme $f(X, Y) = Y^2 - X^3 - aX - b \in \overline{K}[X, Y]$ dans un plan affine. L'idéal engendré par $f(X, Y)$ étant premier dans $\overline{K}[X, Y]$, le quotient

$$\overline{K}[E] := \frac{\overline{K}[X, Y]}{\langle f(X, Y)\rangle}$$

est un anneau intègre, appelé *anneau des coordonnées* de E/K.

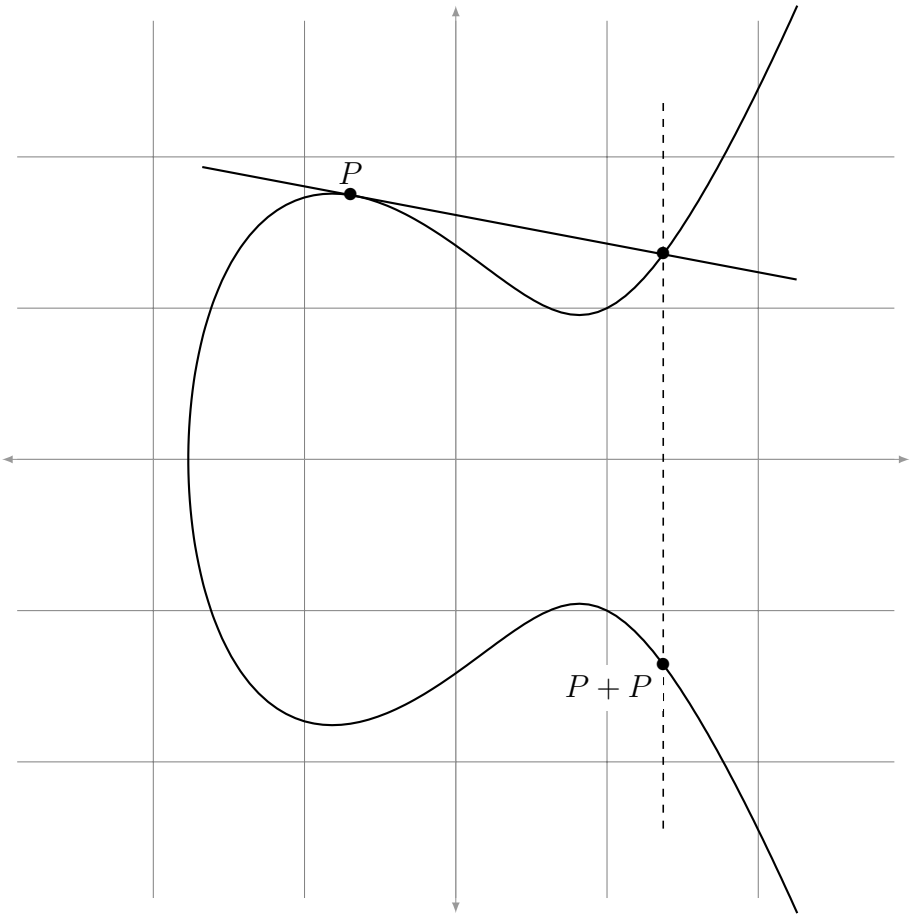

FIGURE II.8 – Un point affine P d'ordonnée non nulle et son double $P + P$.

Algorithme 1 Addition de deux points sur une courbe elliptique

Entrée : $E/K : y^2 = x^3 + ax + b$, $P = (x_P, y_P) \in E$ et $Q = (x_Q, y_Q) \in E$
Sortie : $P + Q \in E$

1: **si** $P = P_\infty$ **alors retourner** Q
2: **si** $Q = P_\infty$ **alors retourner** P
3: **si** $Q = -P$ **alors retourner** P_∞
4: **si** $Q = P$ **alors**
5: $\quad \lambda = (3x_P^2 + a)/(2y_P)$
6: **sinon**
7: $\quad \lambda = (y_Q - y_P)/(x_Q - x_P)$
8: **fin si**
9: $x_0 = \lambda^2 - x_P - x_Q$ **et** $y_0 = \lambda(x_P - x_0) - y_P$
10: **retourner** $P + Q = (x_0, y_0)$

Le corps des fractions de $\overline{K}[E]$ est appelé *corps des fonctions* de E/K, noté $\overline{K}(E)$. Les éléments de $\overline{K}(E)$ sont des quotients, modulo $f(X,Y)$, de polynômes en X et Y. Soit $F(X,Y) \in \overline{K}(E)$, *i.e.* il existe $g(X,Y) \in \overline{K}[E]$ et $h(X,Y) \in \overline{K}[E]$ non nul, tels que $F(X,Y) = g(X,Y)/h(X,Y)$. Il est toujours possible de choisir $g(X,Y)$ et $h(X,Y)$ avec un degré en Y égal à 0 ou 1, car $Y^2 \equiv X^3 + aX + b \pmod{f(X,Y)}$. Ainsi, on peut supposer qu'il existe $g_0(X), g_1(X), h_0(X), h_1(X)$ dans $\overline{K}[X]$ tels que

$$\begin{cases} g(X,Y) = g_0(X) + Y g_1(X), \\ h(X,Y) = h_0(X) + Y h_1(X). \end{cases}$$

Notons $\hat{h}(X,Y) := h_0(X) - Y h_1(X)$ et remarquons

$$\begin{aligned} h(X,Y)\hat{h}(X,Y) &= (h_0(X) + Y h_1(X))(h_0(X) - Y h_1(X)), \\ &= h_0(X)^2 - Y^2 h_1(X), \\ &\equiv h_0(X)^2 - (X^3 + aX + b)h_1(X) \pmod{f(X,Y)}, \end{aligned}$$

donc $h(X,Y)\hat{h}(X,Y) \in \overline{K}[X]$. Cela signifie que pour tout $F(X,Y) \in \overline{K}(E)$, il existe $R(X)$ et $S(X)$ dans $\overline{K}(X)$ deux fractions rationnelles à une indéterminée, telles que

$$F(X,Y) = R(X) + Y S(X) \in \overline{K}(E).$$

Nous définissons de la même façon l'anneau $K[E]$ et le corps $K(E)$.

Zéros et pôles d'une fonction rationnelle

Comme pour les fractions rationnelles, il est possible de définir les *zéros* et les *pôles* d'une fonction rationnelle de $\overline{K}(E)$. Intuitivement, un zéro de $F \in \overline{K}(E)$ est un point $P = (x,y) \in E$ tel que $F(P) = F(x,y) = 0$. Alors qu'un pôle de la fonction F est un point Q en lequel elle n'est pas définie (on dit aussi qu'elle n'est pas *régulière* au point Q), par exemple lorsque le numérateur est non nul alors que le dénominateur l'est ; dans ce cas, on note $F(Q) = \infty$.

Formalisons. Pour chaque point $P \in E$, on définit l'idéal $\mathfrak{m}_P$ de $\overline{K}[E]$ par

$$\mathfrak{m}_P := \{f \in \overline{K}[E] : f(P) = 0\}.$$

On remarquons que l'idéal $\mathfrak{m}_P$ est maximal grâce à l'isomorphisme $\overline{K}[E]/\mathfrak{m}_P \to \overline{K}$ défini

par $f \mapsto f(P)$.

L'idéal maximal $\mathfrak{m}_P$ étant premier, l'ensemble $S := \overline{K}[E] \setminus \mathfrak{m}_P$ est une *partie multiplicative* de $A := \overline{K}[E]$ (la notation avec A et S permet d'alléger l'écriture). Cela veut dire que $1_A \in S \subseteq A \setminus \{0_A\}$ et pour tous $x \in S$ et $y \in S$, le produit xy est aussi dans S, car $\mathfrak{m}_P$ est un idéal premier (*i.e.* un produit est dans $\mathfrak{m}_P$ si, et seulement si, un des deux facteurs est dans $\mathfrak{m}_P$). Une pratique courante dans ce cas est de *localiser* l'anneau A en S, c'est à dire que l'on « rend » inversibles les éléments de S. Ce localisé est un anneau, noté $S^{-1}A$ dans la littérature, mais que nous notons $\overline{K}[E]_P$ et appelons *anneau local de E en P*. En d'autres mots,

$$\overline{K}[E]_P := \left\{ F \in \overline{K}(E) : F = \frac{f}{g} \text{ pour } f, g \in \overline{K}[E] \text{ avec } g(P) \neq 0 \right\},$$

c'est-à-dire que $\overline{K}[E]_P$ est l'anneau des fonctions rationnelles régulières en P. L'idéal $\mathfrak{m}_P$ est encore maximal dans $\overline{K}[E]_P$, mais il est l'unique (c'est la définition d'un anneau local : un anneau commutatif possédant un unique idéal maximal). La particularité supplémentaire est que, E étant de dimension 1 (*i.e.* c'est une courbe), $\overline{K}[E]_P$ est principal : c'est un *anneau de valuation discrète* [Sil09, Proposition II.1.1].

La valuation (normalisée discrète) sur $\overline{K}[E]_P$ est $\mathrm{ord}_P : \overline{K}[E]_P \to \mathbb{N}$ définie par

$$\mathrm{ord}_P(f) := \sup\{d \in \mathbb{N} : f \in \mathfrak{m}_P^d\}.$$

En utilisant $\mathrm{ord}_P(f/g) = \mathrm{ord}_P(f) - \mathrm{ord}_P(g)$, on étend $\mathrm{ord}_P : \overline{K}(E) \to \mathbb{Z}$. L'idéal $\mathfrak{m}_P$ de $\overline{K}[E]_P$ est alors formé de tous les éléments de valuation strictement positive, *i.e.* des fonctions F telles que $F(P) = 0$, et sont donc non-inversibles (dans $\overline{K}[E]_P$), alors que les éléments de valuation nulle sont tous les éléments inversibles : ce sont les unités de $\overline{K}[E]_P$.

Définition II.1.2 (Uniformisante)**.** Une fonction $u \in \overline{K}(E)$ qui engendre l'idéal $\mathfrak{m}_P$ est appelée *uniformisante* de E au point P.

Pour une courbe elliptique $E : y^2 = x^3 + ax + b$, une uniformisante u au point $P \in E$, de coordonnées affines (x_P, y_P) le cas échéant, est donnée par

$$u(x,y) = \begin{cases} x - x_P & \text{si } y_P \neq 0 \text{ et } P \neq P_\infty, \\ y & \text{si } y_P = 0, \\ x/y & \text{si } P = P_\infty. \end{cases} \tag{II.1.6}$$

Proposition II.1.1. *Une fonction u est uniformisante au point P si, et seulement si,* $\mathrm{ord}_P(u) = 1$. *De plus, pour un point $P \in E$ et une fonction $F \in \overline{K}(E)$ telle que $F(P) = 0$ ou ∞, il existe une fonction $G \in \overline{K}(E)$ telle que P ne soit ni un zéro, ni un pôle de G, et une uniformisante u en P telles que*

$$F = u^{\mathrm{ord}_P(F)} G.$$

Nous pouvons alors conclure en explicitant notre intuition de départ.

Définition II.1.3 (Zéro et pôle d'une fonction)**.** Soient E une courbe elliptique et P un point de E. Soit $f \in \overline{K}(E)$, l'*ordre de f en P* est $\mathrm{ord}_P(f)$. Si $\mathrm{ord}_P(f) > 0$ alors f a un *zéro* en P et si $\mathrm{ord}_P(f) < 0$ alors f a un *pôle* en P. Si $\mathrm{ord}_P(f) \geqslant 0$ alors f est *régulière* (ou *définie*) en P et on peut évaluer $f(P)$. Sinon, f a un pôle en P et on écrit $f(P) = \infty$.

Proposition II.1.2 ([Sil09, Proposition II.1.2])**.** *Une fonction non nulle de $\overline{K}(E)$ a un nombre fini de zéros et de pôles, et ils sont en nombre égaux (comptés avec multiplicité). De plus, si elle n'admet aucun pôle, alors elle est constante.*

Isogénie et isomorphisme

Les applications entre courbes elliptiques qui conservent la structure de groupe sont appelées des isogénies.

Définition II.1.4 (Isogénie)**.** Soient E_1/K et E_2/K deux courbes elliptiques. Une *isogénie* ϕ de E_1 vers E_2 est de la forme $\phi(x, y) = (F(x, y), G(x, y))$, avec $F(x, y)$ et $G(x, y)$ deux fonctions rationnelles de $\overline{K}(E_1)$ telles que $(F(P), G(P)) \in E_2$ pour tout point $P \in E_1$, et envoyant le point à l'infini de E_1 sur le point à l'infini de E_2. De plus, ϕ est dite *définie sur K* lorsque $F(x, y)$ et $G(x, y)$ sont dans $K(E_1)$.

Théorème II.1.1. *Une isogénie est un morphisme de groupes soit constant, soit surjectif de noyau fini.*

Définition II.1.5 (Isomorphisme)**.** Un *isomorphisme* entre courbes elliptiques est une isogénie de noyau $\{P_\infty\}$. Deux courbes elliptiques E_1/K et E_2/K sont dites isomorphes sur K si un des deux isomorphismes $E_1 \to E_2$ ou $E_2 \to E_1$ est défini sur K (l'autre le sera alors automatiquement).

Les isomorphismes qui conservent le modèle de Weierstrass (court) sont de la forme

$$\psi : (x, y) \mapsto (u^2 x, u^3 y)$$

pour $u \in \overline{K}$ non nul, entre des courbes d'équation $y^2 = x^3 + ax + b$ et $y^2 = x^3 + u^4 ax + u^6 b$. Une quantité utile, qui permet de déterminer lorsque deux courbes sont isomorphes, est le j-invariant.

Définition II.1.6 (j-invariant)**.** Le *j-invariant* d'une courbe $E/K : y^2 = x^3 + ax + b$ est l'élément $j(E)$ défini par

$$j(E) := -1728 \frac{(4a)^3}{\Delta(E)} = 1728 \frac{4a^3}{4a^3 + 27b^2}.$$

Théorème II.1.2. *Deux courbes elliptiques définies sur K sont isomorphes sur $\overline{K}$ si, et seulement si, elles ont le même j-invariant. De plus, pour tout $j_0 \in \overline{K}$, il existe une courbe elliptique définie sur l'extension $K(j_0)$ de j-invariant égal à j_0.*

Soit $j_0 \in \overline{K}$. Donnons des courbes de j-invariant j_0 [Was08, Section 2.7].

— Si $j_0 = 0$, alors la courbe $y^2 = x^3 + 1$ est de j-invariant 0.

— Si $j_0 = 1728$, alors la courbe $y^2 = x^3 + x$ est de j-invariant 1728.

— Si $j_0 \neq 0, 1728$, alors la courbe

$$y^2 = x^3 + \frac{3j_0}{1728 - j_0} x + \frac{2j_0}{1728 - j_0}$$

est de j-invariant j_0.

Remarquons que toute courbe $y^2 = x^3 + ax + b$ est de j-invariant 0 (respectivement 1728), si, et seulement si, $a = 0$ (respectivement $b = 0$).

Pour une courbe elliptique E, l'ensemble des isogénies $E \to E$ est l'*anneau d'endomorphisme* de E, noté $\mathrm{End}(E)$, pour l'addition ($+$) et la composition ($\circ$), et l'ensemble des isomorphismes $E \to E$ est le *groupe des automorphismes*, noté $\mathrm{Aut}(E)$, pour la composition ($\circ$).

Multiple d'un point

L'une des opérations les plus utilisées en cryptographie est le calcul d'un multiple d'un point. Cela correspond à une exponentiation dans un groupe générique, noté multiplicativement. Des méthodes classiques, comme le *square-and-multiply*, permettent de calculer

ces multiples de point. Soit $m \in \mathbb{Z}$ non nul. La *multiplication-par-m* est un endomorphisme de la courbe elliptique E, noté $[m] : E \to E$. L'Algorithme 2 calcule $[m]P$, pour $P \in E$, en imitant l'exponentiation rapide.

Algorithme 2 *Double-and-add*

Entrée : $P \in E$ et $m = (m_{n-1} \dots m_1 m_0)_2$ avec $m_{n-1} = 1$
Sortie : $[m]P \in E$

1: **Si** $m = 0$ *alors retourner* P_∞
2: $R \leftarrow P$
3: **pour** i **allant de** $n-2$ **à** 0 **faire**
4: $\quad R \leftarrow [2]R$
5: $\quad$ **si** $m_i = 1$ **alors**
6: $\quad\quad R \leftarrow P + R$
7: $\quad$ **fin si**
8: **fin pour**
9: **retourner** $[m]P = R$

Le noyau de l'application $[m]$ est appelé *groupe de m-torsion*, noté

$$E[m] := \mathrm{Ker}[m] = \{P \in E : [m]P = P_\infty\}.$$

Théorème II.1.3 (Classification des groupes de torsion [Sil09, Corollaire III.6.4])**.** *Soient E/K une courbe elliptique et $m \geqslant 1$ un entier. Si $m \neq 0$ dans K alors*

$$E[m] \simeq \frac{\mathbb{Z}}{m\mathbb{Z}} \times \frac{\mathbb{Z}}{m\mathbb{Z}},$$

Sinon, K est de caractéristique $p > 0$ et p divise m : de deux choses l'une,

— *soit pour tout $e \geqslant 1$, les torsions sont triviales :*

$$E[p^e] = \{P_\infty\};$$

— *soit pour tout $e \geqslant 1$, les torsions sont cycliques d'ordre p^e :*

$$E[p^e] \simeq \frac{\mathbb{Z}}{p^e\mathbb{Z}}.$$

L'application $[\ ] : \mathbb{Z} \to \mathrm{End}(E)$, qui à un entier $m \neq 0$ associe l'endomorphisme $[m]$ et à 0 associe l'endomorphisme constant $[0] : P \mapsto P_\infty$, induit une inclusion de $\mathbb{Z}$ dans $\mathrm{End}(E)$, énoncée par le théorème suivant. Nous verrons plus loin (Section II.4) une

description plus détaillée de End(E).

Théorème II.1.4 ([Sil09, Proposition III.4.2])**.** *Pour une courbe elliptique E, l'anneau d'endomorphisme* End(E) *est un anneau de caractéristique nulle sans diviseur de zéro.*

La multiplication-par-m étant un endomorphisme, il existe $F(x,y)$ et $G(x,y)$ dans $K(E)$ vérifiant

$$[m](x,y) = (F(x,y), G(x,y)) .$$

Les fractions $F(x,y)$ et $G(x,y)$ sont définies à l'aide des polynômes de division.

Définition II.1.7 (Polynôme de division)**.** Soit $E/K : y^2 = x^3 + ax + b$ une courbe elliptique. Notons $\mathcal{A} = \mathbb{Z}[a,b]$ l'anneau de coefficients. Les *polynômes de division* $(\psi_n(X,Y))_{n\geqslant 0}$ sont des polynômes de $\mathcal{A}[X,Y]$ définis par la récurrence (omettons les indéterminées) :

$$\begin{aligned}
\psi_0 &= 0;\\
\psi_1 &= 1;\\
\psi_2 &= 2Y;\\
\psi_3 &= 3X^4 + 6aX^2 + 12bX - a^2;\\
\psi_4 &= 4Y(X^6 + 5aX^4 + 20bX^3 - 5a^2X^2 - 4abX - 8b^2 - a^3);\\
\psi_{2n+1} &= \psi_{n+2}\psi_n^3 - \psi_{n-1}\psi_{n+1}^3 \quad \text{pour } n \geqslant 2;\\
\psi_{2n} &= \frac{\psi_n}{2Y}\left(\psi_{n+2}\psi_{n-1}^2 - \psi_{n-2}\psi_{n+1}^2\right) \quad \text{pour } n \geqslant 3.
\end{aligned}$$

Le polynôme $\psi_n(X,Y)$ est appelé le *n-ième polynôme de division.*

À l'aide de la définition des polynômes de division $\psi_n(X,Y) \in \mathcal{A}[X,Y]$, nous constatons que, pour $n \geqslant 0$, $\psi_{2n+1} \in \mathcal{A}[X]$ et $\psi_{2n} \in 2Y\mathcal{A}[X]$; ce qui implique $\psi_n^2 \in \mathcal{A}[X]$ et $\psi_{2n}/Y \in \mathcal{A}[X]$.

Théorème II.1.5. *Soit $m \geqslant 1$. L'endomorphisme multiplication-par-m est donné par*

$$[m](x,y) = \left(\frac{\phi_m(x)}{\psi_m(x)^2}, \frac{\omega_m(x,y)}{\psi_m(x,y)^3}\right),$$

où les polynômes $\phi_m(X)$ et $\omega_m(X,Y)$ sont définis par

$$\begin{aligned}
\phi_m(X) &= X\psi_m^2 - \psi_{m+1}\psi_{m-1},\\
\omega_m(X,Y) &= \left(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2\right)/4Y.
\end{aligned}$$

Proposition II.1.3. *Soit $E/K : y^2 = x^3 + ax + b$ une courbe elliptique et soit $m \geqslant 1$. Les racines dans $\overline{K}$ du polynôme de division ψ_{2m+1}, respectivement ψ_{2m}/Y, donnent les abscisses des points de $E[2m+1]\backslash\{P_\infty\}$, respectivement de $E[2m]\backslash E[2]$.*

Dans la proposition précédente, le Y dans ψ_{2m} correspond à $E[2]$. En effet, nous avons déjà vu que les points P d'ordre 2, *i.e.* $P \in E[2]\backslash\{P_\infty\}$, sont d'ordonnée nulle. Le groupe $E[2]$ est donc formé des points P_∞, $(x_1, 0)$, $(x_2, 0)$ et $(x_3, 0)$ où x_1, x_2 et x_3 sont les trois racines distinctes de $x^3 + ax + b$ dans $\overline{K}$.

II.1.3 Courbe elliptique sur un corps fini

Soit $\mathbb{F}_q$ un corps fini à q éléments de caractéristique $p > 3$, *i.e.* il existe e entier tel que $q = p^e$ et $p \notin \{2, 3\}$. Le nombre de points $\mathbb{F}_q$-rationnels et la structure de $E(\mathbb{F}_q)$ sont des données importantes pour déterminer l'usage d'une courbe en cryptographie. Un encadrement de $\#E(\mathbb{F}_q)$ est donné par la borne de Hasse.

Théorème II.1.6 (Borne de Hasse [Sil09, Théorème V.1.1])**.** *Soit $E/\mathbb{F}_q$ une courbe elliptique. Alors*

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

Cette borne indique que le nombre de points sur une courbe est du même ordre que la taille du corps, c'est-à-dire $\#E(\mathbb{F}_q) \sim q$. La structure du groupe $E(\mathbb{F}_q)$ est déduit du théorème de classification des groupes abéliens finis (Théorème I.1.2).

Théorème II.1.7. *Soit $E/\mathbb{F}_q$ une courbe elliptique. Alors il existe deux entiers n_1 et n_2 tels que n_1 divise n_2 et n_1 divise $q-1$ et*

$$E(\mathbb{F}_q) \simeq \frac{\mathbb{Z}}{n_1\mathbb{Z}} \times \frac{\mathbb{Z}}{n_2\mathbb{Z}}.$$

La courbe $E(\mathbb{F}_q)$ est donc générée par au plus deux points.

Endomorphisme de Frobenius

Tout comme sur un corps fini, le Frobenius joue un rôle important comme endomorphisme de courbe elliptique. Il est défini par

$$\begin{array}{rccc} \pi : & E & \to & E \\ & (x, y) & \mapsto & (x^q, y^q). \end{array}$$

Le Frobenius π fixe $E(\mathbb{F}_q)$. Cela signifie que l'endomorphisme $\pi - [1]$ est de noyau $E(\mathbb{F}_q)$ et

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

Le polynôme minimal de π est $X^2 - tX + q$, où t est appelé la *trace du Frobenius*. Cela signifie que $\pi^2 - [t] \circ \pi + [q] = 0$ dans $\operatorname{End}(E)$, ou encore

$$(x^{q^2}, y^{q^2}) - [t](x^q, y^q) + [q](x, y) = P_\infty, \quad \text{pour tout point } (x, y) \in E.$$

Il s'avère que les racines complexes du polynôme $X^2 - tX + q$ permettent de calculer $\#E(\mathbb{F}_{q^n})$ pour un entier $n \geqslant 1$.

Théorème II.1.8 (Cardinal de $E(\mathbb{F}_{q^n})$ [Sil09, Théorème V.2.3.1])**.** *Soit t la trace du Frobenius sur $E/\mathbb{F}_q$. Soient α et β les deux racines conjuguées complexes du polynôme minimal du Frobenius $X^2 - tX + q$. Alors $|\alpha| = |\beta| = \sqrt{q}$ et pour tout $n \geqslant 1$*

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

La définition de la forme trace trouve aussi son équivalent sur les courbes elliptiques. Pour un point $P = (x, y) \in E(\mathbb{F}_{q^n})$, la *trace de P* est définie par

$$\operatorname{Tr}(P) := \sum_{i=0}^{n-1} \pi^i(P) = \sum_{i=0}^{n-1} (x^{q^i}, y^{q^i}),$$

où $\pi^i := \pi \circ \cdots \circ \pi$ est la composée i fois du morphisme de Frobenius. Par ailleurs, la théorie de Galois indique que l'image de Tr est dans $E(\mathbb{F}_q)$.

Groupe de torsion et degré de plongement

Le groupe de torsion est central en cryptographie à couplage sur courbe elliptique. En effet, pour calculer un couplage, nous avons besoin de deux points $P \in E$ et $Q \in E$ d'ordre premier ℓ tels que $\langle P \rangle \cap \langle Q \rangle = \{P_\infty\}$. Décrivons donc $E[\ell]$ pour $E/\mathbb{F}_q$.

Soit ℓ un entier premier ne divisant pas q. Comme nous l'avons vu au Théorème II.1.3, le groupe de ℓ-torsion $E[\ell]$ est d'ordre ℓ^2 isomorphe à

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

Cela implique que $E[\ell]$ consiste en $\ell + 1$ sous-groupes d'ordre ℓ et, car $E[\ell]$ est fini, qu'il

existe un entier k tel que $E[\ell] \subset E(\mathbb{F}_{q^k})$. Le plus petit de ces entiers k est appelé *degré de plongement* de q par rapport à ℓ, il vérifie les conditions équivalentes suivantes :

- k est le plus petit entier positif tel que ℓ divise $q^k - 1$, *i.e.* k est l'ordre de $q \pmod{\ell}$;
- k est le plus petit entier positif tel que $\mathbb{F}_{q^k}$ contienne toutes les racines ℓ-ième de l'unité de $\overline{\mathbb{F}_q}$, *i.e.* $\mu_\ell \subset \mathbb{F}_{q^k}$;
- k est le plus petit entier positif tel que $E[\ell] \subset E(\mathbb{F}_{q^k})$.

Si ℓ divise $\#E(\mathbb{F}_q)$ une fois, *i.e.* ℓ^2 ne divise pas $\#E(\mathbb{F}_q)$, alors $k > 1$ et il n'y a qu'un seul sous-groupe d'ordre ℓ dans $E(\mathbb{F}_q)$. Cela implique que pour obtenir d'autres points d'ordre ℓ, nous sommes obligés de regarder la courbe sur une extension, et en faisant cela, lorsque l'on trouve un point de $E[\ell] \backslash E(\mathbb{F}_q)$, en fait on trouve toute la ℓ-torsion $E[\ell]$. Dans la suite du manuscrit, on suppose que l'on est toujours dans ce cas là : ℓ divise $\#E(\mathbb{F}_q)$ une fois.

La trace $\mathrm{Tr} : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_q)$ envoie alors tous les points de la torsion $E[\ell]$ dans $E[\ell] \cap \mathrm{Ker}(\pi - [1]) = E(\mathbb{F}_q)[\ell]$. Du polynôme minimal $X^2 - tX + q$ de π, on déduit que la seconde valeur propre du morphisme de Frobenius est q. Ainsi $\mathrm{Ker}(\pi - [q])$ est non-vide. Regardons la trace sur ce noyau : soit $P \in \mathrm{Ker}(\pi - [q])$,

$$\begin{aligned}
\mathrm{Tr}(P) &= P + \pi(P) + \cdots + \pi^{k-1}(P), \\
&= P + [q]P + \cdots + [q^{k-1}]P, \\
&= [1 + q + \cdots + q^{k-1}]P, \\
&= \left[\frac{q^k - 1}{q - 1}\right] P.
\end{aligned}$$

Si $P \in E[\ell] \cap \mathrm{Ker}(\pi - [q])$ alors $\mathrm{Tr}(P) = P_\infty$ car ℓ est premier et $q \pmod{\ell}$ est d'ordre k, donc ℓ divise $(q^k - 1)/(q - 1)$. Ce sous-groupe $E[\ell] \cap \mathrm{Ker}(\pi - [q])$ de $E[\ell]$ est appelé le *sous-groupe de trace nulle* et est d'ordre ℓ ([BSS05, Lemme IX.16]). Il existe une application aTr qui à $P \in E[\ell]$ associe $\mathrm{aTr}(P) \in E[\ell] \cap \mathrm{Ker}(\pi - [q])$, c'est l'application appelée *anti-trace* $\mathrm{aTr} : P \mapsto [k]P - \mathrm{Tr}(P)$. En remarquant que $\mathrm{Tr}(P) = [k]P$ pour tout $P \in E(\mathbb{F}_q)$, lorsque $\mathrm{Tr} : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_q)$, on se convainc facilement que $\mathrm{Tr} \circ \mathrm{aTr}$ envoie tous les points de $E(\mathbb{F}_{q^k})$ sur P_∞ : soit $P \in E(\mathbb{F}_{q^k})$,

$$\begin{aligned}
\mathrm{Tr} \circ \mathrm{aTr}(P) &= \mathrm{Tr}([k]P - \mathrm{Tr}(P)), \\
&= \mathrm{Tr}([k]P) - \mathrm{Tr}(\mathrm{Tr}(P)), \\
&= [k]\,\mathrm{Tr}(P) - [k]\,\mathrm{Tr}(P), \\
&= P_\infty.
\end{aligned}$$

Nous avons ainsi exhibé deux sous-groupes particuliers de la torsion $E[\ell]$. Le premier $E[\ell] \cap \mathrm{Ker}(\pi - [1]) = E(\mathbb{F}_q)[\ell]$ est l'unique sous-groupe d'ordre ℓ de E formé de points $\mathbb{F}_q$-rationnels. Le second $E[\ell] \cap \mathrm{Ker}(\pi - [q])$ est le sous-groupe de trace nulle défini sur $\mathbb{F}_{q^k}$. Deux applications, les restrictions de la trace Tr et l'anti-trace aTr, envoient les éléments de la torsion dans ces deux sous-groupes : la trace Tr envoie les points de $E[\ell]$ sur $E[\ell] \cap \mathrm{Ker}(\pi - [1])$ et l'anti-trace aTr envoie les points de $E[\ell]$ sur $E[\ell] \cap \mathrm{Ker}(\pi - [q])$.

Courbe supersingulière

On a vu plus tôt que $E[p^e]$, pour $e > 0$, était soit $\{P_\infty\}$ soit cyclique d'ordre p^e lorsque E est définie sur un corps fini de caractéristique p. Cette distinction est en fait plus fondamentale : elle différencie les *courbes ordinaires* des *courbes supersingulières*.

Définition II.1.8 (Courbe supersingulière [Sil09, Théorème V.3.1])**.** Soit $E/\mathbb{F}_q$ une courbe définie sur un corps fini de caractéristique première p. La courbe elliptique E est *supersingulière* si pour tout $e > 0$,

$$E[p^e] = \{P_\infty\}.$$

Dans le cas contraire, E est dite *ordinaire* et $E[p^e] \simeq \mathbb{Z}/p^e\mathbb{Z}$.

Il y a d'autres caractérisations des courbes supersingulières, une concernant l'anneau d'endomorphismes de la courbe, que nous énoncerons plus tard, une autre concernant la trace du Frobenius, que nous donnons ici.

Proposition II.1.4. *Une courbe elliptique E définie sur un corps fini de caractéristique première p est supersingulière si, et seulement si, la trace du Frobenius est $t \equiv 0 \pmod{p}$.*

Dans le cas où E est supersingulière et définie sur un corps premier fini $\mathbb{F}_p$, pour $p \geqslant 5$, la trace t du Frobenius est nulle : les conditions $t \equiv 0 \pmod{p}$ et $|t| \leqslant 2\sqrt{p}$ impliquent $t = 0$. Ainsi $\#E(\mathbb{F}_p) = p + 1$ [Wat69].

Pour une courbe supersingulière $E/\mathbb{F}_p$, on constate que $(p-1) \times \#E(\mathbb{F}_p) = p^2 - 1$, donc tout premier ℓ divisant $\#E(\mathbb{F}_p)$ divise aussi $p^2 - 1$, et le degré de plongement [3] de ℓ par rapport à p est 2. Cela signifie que $E[\ell] \subset E(\mathbb{F}_{p^2})$. Nous verrons plus tard que ces courbes supersingulières n'offrent pas une sécurité suffisante, les courbes ordinaires étant alors privilégiées.

3. Toujours dans le cas où ℓ divise $\#E(\mathbb{F}_p)$ une fois.

II.1.4 Tordues

Soient deux courbes elliptiques (ordinaires) E_1/K et E_2/K de même j-invariant, cela signifie qu'elles sont isomorphes sur la clôture algébrique $\overline{K}$. Cependant, il est possible que l'isomorphisme ne soit pas défini sur K, mais sur une extension de K de degré au plus 6 [Sil09, Proposition X.5.4]. Dans ce cas là, on dit que E_2 est une *tordue* de E_1 de degré d, où d est le plus petit degré d'extension $[L : K]$ de K tel que l'isomorphisme soit défini sur L.

Rappelons-nous que, pour $u \in \overline{K}$, un isomorphisme entre $E_1/K : y^2 = x^3 + ax + b$ et $E_2/K : y^2 = x^3 + u^4ax + u^6b$, où u^4a et u^6b sont dans K, envoie le point (x, y) de E_1 sur le point (u^2x, u^3y) de E_2. Notons $\Psi : E_1 \to E_2$ cet isomorphisme et $\Psi^{-1} : E_2 \to E_1$ sa réciproque [4]. Si $u^2 \in K$ et $u^3 \in K$ alors Ψ est défini sur K et $\mathrm{Im}(\Psi) = E_2(K)$, dans le cas contraire E_2 est une tordue de E_1 de degré $d = [K(u^2, u^3) : K]$.

Par simplicité, supposons que K est le corps fini $\mathbb{F}_q$ et donc $u \in \overline{\mathbb{F}_q}$. Distinguons les cas en fonction de $d \in \{2, 3, 4, 6\}$:

Cas $d = 2$: la tordue est dite *quadratique* et u engendre une extension $\mathbb{F}_{q^2}$ de degré deux de $\mathbb{F}_q$ avec $u^2 \in \mathbb{F}_q$. L'isomorphisme Ψ est défini alors sur $\mathbb{F}_{q^2}$, donc pour un point $P \in E_1(\mathbb{F}_q)$, l'image $\Psi(P) \in E_2(\mathbb{F}_{q^2})$.

Ainsi, toute courbe elliptique $E_1/\mathbb{F}_q : y^2 = x^3 + ax + b$ possède une tordue quadratique $E_2/\mathbb{F}_q : y^2 = x^3 + \omega^2ax + \omega^3b$ pour tout $\omega \in \mathbb{F}_q$ non-carré et l'isomorphisme Ψ est défini sur $\mathbb{F}_q(\sqrt{\omega}) = \mathbb{F}_{q^2}$, envoyant un point de $E_1(\mathbb{F}_q)$ dans $E_2(\mathbb{F}_{q^2})$.

Cas $d = 3$: la tordue est dite *cubique* et u engendre une extension $\mathbb{F}_{q^3}$ de degré trois de $\mathbb{F}_q$ avec $u^2 \in \mathbb{F}_{q^3}$ et $u^3 \in \mathbb{F}_q$. L'isomorphisme Ψ est défini alors sur $\mathbb{F}_{q^3}$, donc pour un point $P \in E_1(\mathbb{F}_q)$, l'image $\Psi(P) \in E_2(\mathbb{F}_{q^3})$. La contrainte $u^4a \in \mathbb{F}_q$ implique $a = 0$ car $u^4 \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q$.

Ainsi, toute courbe elliptique $E_1/\mathbb{F}_q : y^2 = x^3 + b$ possède une tordue cubique $E_2/\mathbb{F}_q : y^2 = x^3 + \omega^2b$ pour tout $\omega \in \mathbb{F}_q$ non-cube et l'isomorphisme Ψ est défini sur $\mathbb{F}_q(\sqrt[3]{\omega}) = \mathbb{F}_{q^3}$, envoyant un point de $E_1(\mathbb{F}_q)$ dans $E_2(\mathbb{F}_{q^3})$.

Cas $d = 4$: la tordue est dite *quartique* et u engendre une extension $\mathbb{F}_{q^4}$ de degré quatre de $\mathbb{F}_q$ avec $u^2 \in \mathbb{F}_{q^2}$ et $u^4 \in \mathbb{F}_q$. L'isomorphisme Ψ est défini alors sur $\mathbb{F}_{q^4}$, donc pour un point $P \in E_1(\mathbb{F}_q)$, l'image $\Psi(P) \in E_2(\mathbb{F}_{q^4})$. La contrainte $u^6b \in \mathbb{F}_q$ implique $b = 0$ car $u^6 \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$.

4. En littérature anglaise, pour les courbes à couplage, Ψ^{-1} s'appelle la *twisting map* et sa réciproque Ψ s'appelle la *untwisting map*.

Ainsi, toute courbe elliptique $E_1/\mathbb{F}_q : y^2 = x^3 + ax$ possède une tordue quartique $E_2/\mathbb{F}_q : y^2 = x^3 + \omega a x$ pour tout $\omega \in \mathbb{F}_q$ non-puissance 4 dans $\mathbb{F}_q$ et l'isomorphisme Ψ est défini sur $\mathbb{F}_q(\sqrt[4]{\omega}) = \mathbb{F}_{q^4}$, envoyant un point de $E_1(\mathbb{F}_q)$ dans $E_2(\mathbb{F}_{q^4})$.

Cas $d = 6$: la tordue est dite *sextique* et u engendre une extension $\mathbb{F}_{q^6}$ de degré six de $\mathbb{F}_q$ avec $u^2 \in \mathbb{F}_{q^3}$, $u^3 \in \mathbb{F}_{q^2}$ et $u^6 \in \mathbb{F}_q$. L'isomorphisme Ψ est défini alors sur $\mathbb{F}_{q^6}$, donc pour un point $P \in E_1(\mathbb{F}_q)$, l'image $\Psi(P) \in E_2(\mathbb{F}_{q^6})$. La contrainte $u^4 a \in \mathbb{F}_q$ implique $a = 0$ car $u^4 \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q$.

Ainsi, toute courbe elliptique $E_1/\mathbb{F}_q : y^2 = x^3 + b$ possède une tordue sextique $E_2/\mathbb{F}_q : y^2 = x^3 + \omega b$ pour tout $\omega \in \mathbb{F}_q$ ni un carré, ni un cube dans $\mathbb{F}_q$ et l'isomorphisme Ψ est défini sur $\mathbb{F}_q(\sqrt{\omega}, \sqrt[3]{\omega}) = \mathbb{F}_{q^6}$, envoyant un point de $E_1(\mathbb{F}_q)$ dans $E_2(\mathbb{F}_{q^6})$.

Dans le cas des courbes à couplage, on souhaite tirer avantage de la tordue pour définir $\mathbb{G}_2$ sur une extension plus petite que $\mathbb{F}_{q^k}$, ce qui permet de réduire drastiquement l'aspect calculatoire du couplage. En effet, supposons qu'une courbe E a la bonne équation et que $d \in \{2, 3, 4, 6\}$ divise le degré k de plongement par rapport à un premier ℓ. La courbe E est alors la tordue de degré d d'une courbe E' définie sur $\mathbb{F}_{q^{k/d}}$ et les isomorphismes $\Psi : E' \to E$ et $\Psi^{-1} : E \to E'$ permettent d'envoyer des points du sous-groupe $E[\ell] \cap \mathrm{Ker}(\pi - [q]) \subset E(\mathbb{F}_{q^k}) \setminus E(\mathbb{F}_q)$ dans $E'(\mathbb{F}_{q^{k/d}})[\ell]$, où π est le Frobenius de la courbe E. En pratique, on peut prendre un générateur Q' de $E'(\mathbb{F}_{q^{k/d}})[\ell]$ et on pose $Q = \Psi(Q')$, alors Q engendre un sous-groupe d'ordre ℓ dans $E[\ell]$ qui est le sous-groupe de trace nulle dans $E[\ell]$ [HSV06].

II.2 Diviseurs

Un diviseur D sur une courbe E/K est une somme formelle de points :

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

où un nombre fini des $n_P \in \mathbb{Z}$ sont non nuls. Les deux seules différences cosmétiques entre une somme effective et une somme formelle sont, d'une part l'absence de crochet $[\cdot]$ autour du n_P, et d'autre part la présence de parenthèse $(\cdot)$ autour du point P.

L'ensemble des diviseurs de E, noté $\mathrm{Div}(E)$, est un groupe abélien libre généré par les points de E, dont l'élément neutre est le diviseur avec tous les coefficients $n_P = 0$ et la somme de deux diviseurs se fait naturellement en sommant les coefficients entre eux.

On définit le degré, le support et la somme d'un diviseur $D = \sum n_P(P)$ comme suit : le *degré d'un diviseur* est simplement la somme des coefficients

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

le *support d'un diviseur* est l'ensemble, fini, des points de coefficients non nuls

$$\mathrm{supp}(D) = \{P \in E : n_P \neq 0\} \subset E,$$

et la *somme d'un diviseur* est sa somme effective

$$\mathrm{sum}(D) = \sum_{P \in E} [n_P]P \in E.$$

L'ensemble des diviseurs de degré zéro, noté $\mathrm{Div}^0(E)$, forme un sous-groupe de $\mathrm{Div}(E)$.

Diviseur principal

Pour une fonction f sur E, on rappelle que l'on note $\mathrm{ord}_P(f)$ la multiplicité de f au point P, qui est strictement positive si P est un zéro de f et strictement négative si P est un pôle de f (voir la Définition II.1.3). On écrit $\mathrm{div}(f)$ le *diviseur associé à une fonction* f, défini par

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

Les opérations sur les fonctions se traduisent par des opérations sur les diviseurs, $\mathrm{div}(fg) = \mathrm{div}(f) + \mathrm{div}(g)$ et $\mathrm{div}(f/g) = \mathrm{div}(f) - \mathrm{div}(g)$. De plus, $\mathrm{div}(f) = 0$ si, et seulement si, f est une constante, ce qui implique que si $\mathrm{div}(f) = \mathrm{div}(g)$ alors f est un multiple non nul de g et $\mathrm{div}(f)$ est défini par f à constante non nulle près. Le degré d'un diviseur d'une fonction est toujours nul : $\deg(\mathrm{div}(f)) = 0$.

L'ensemble $\mathrm{Prin}(E)$ des diviseurs de fonction forment un sous-groupe de $\mathrm{Div}^0(E)$, appelés *diviseurs principaux* :

$$D \in \mathrm{Prin}(E) \Leftrightarrow \exists f \in \overline{K}(E), \quad D = \mathrm{div}(f).$$

Ou de façon équivalente ([Sil09, Corollaire III.3.5)]) :

$$D \in \mathrm{Prin}(E) \Leftrightarrow D \in \mathrm{Div}^0(E), \quad \mathrm{sum}(D) = P_\infty.$$

Les inclusions de sous-groupes suivantes sont strictes

$$\mathrm{Prin}(E) \subset \mathrm{Div}^0(E) \subset \mathrm{Div}(E),$$

car, par exemple, pour deux points $P, Q \in E$ tels que $Q \neq -P$, $(P) - (Q)$ est de degré nul mais pas principal et (P) n'est pas de degré nul.

Groupe de Picard

Dans $\mathrm{Div}(E)$, on dit que deux diviseurs D_1 et D_2 sont équivalents si, et seulement si, leur différence est un diviseur principal. Autrement dit, en notant l'équivalence $\sim$,

$$D_1 \sim D_2 \Leftrightarrow \exists f \in \overline{K}(E), \quad D_1 = D_2 + \mathrm{div}(f).$$

Le *groupe de Picard* est défini comme le groupe quotient

$$\mathrm{Pic}^0(E) = \mathrm{Div}^0(E)/\,\mathrm{Prin}(E),$$

c'est-à-dire que le groupe de Picard est le groupe des diviseurs de degré zéro modulo les diviseurs principaux sur E.

II.2.1 Diviseur d'une droite

Parmi les fonctions sur la courbe E, les plus simples (non constantes) sont les droites. Comme vu lors de la définition de la loi de groupe sur une courbe elliptique, on distingue le cas des droites verticales des autres.

Droite verticale

Soit $P = (x_P, y_P) \in E$ et $P \neq P_\infty$. Notons $\nu_P(x, y) = x - x_P$ la droite verticale passant par P. Cette fonction ν_P a un zéro d'ordre 1 en P, un autre en $-P$ et un pôle d'ordre 2 en P_∞. Et si $P = -P$, alors l'ordre du zéro en P est 2. En effet, à l'aide de la définition des uniformisantes (II.1.6) :

- Si P n'est pas sur l'axe des abscisses, alors $P \neq -P$ et l'uniformisante en P et $-P$ est $(x, y) \mapsto x - x_P$. Ainsi, $\mathrm{ord}_P(\nu_P) = \mathrm{ord}_{-P}(\nu_P) = 1$.
- Si P est sur l'axe des abscisses, *i.e.* $y_P = 0$, alors $P = -P$ et l'uniformisante en P est $(x, y) \mapsto y$. Cela signifie que x_P est une des trois racines (distinctes car E est

une courbe elliptique) du polynôme $x^3 + ax + b = (x - x_P)(x - x_0)(x - x_1)$. Ainsi

$$\nu_P(x, y) = \frac{y^2}{(x - x_0)(x - x_1)},$$

et $\mathrm{ord}_P(\nu_P) = 2$. Ce qui correspond bien au fait que dans ce cas ν_P est tangente à E en P.

— En P_∞ l'uniformisante est $(x, y) \mapsto x/y$. Alors ν_P s'écrit

$$\nu_P(x, y) = \left(\frac{x}{y}\right)^{-2} \frac{x^3 - x_P x^2}{y^2},$$

où $(x, y) \mapsto (x^3 - x_P x^2)/y^2$ vaut 1 lorsque évaluée en P_∞. Ainsi P_∞ est un pôle d'ordre 2 de ν_P, *i.e.* $\mathrm{ord}_{P_\infty}(\nu_P) = -2$.

En résumé, pour $P \neq P_\infty$, le diviseur de la droite verticale passant par P est

$$\mathrm{div}(\nu_P) = (P) + (-P) - 2(P_\infty).$$

Droite non-verticale

Soient $P = (x_P, y_P) \in E$ et $Q = (x_Q, y_Q) \in E$ deux points différents de P_∞ tels que $P \neq \pm Q$. Notons $\delta_{P,Q}$ la droite passant par P et Q. De deux choses l'une, soit $\delta_{P,Q}$ est tangente en P (zéro d'ordre 2) et alors $Q = -[2]P$ (zéro d'ordre 1), soit $\delta_{P,Q}$ intersecte la courbe E en un troisième point $-P - Q$ (chacun d'eux est un zéro d'ordre 1). Dans les deux cas, P_∞ est un pôle d'ordre 3. Détaillons :

— Si $\delta_{P,Q}$ est tangente en P alors $Q = -[2]P$ et $y_P \neq 0$. L'équation de la droite est alors $\delta_{P,Q}(x, y) = y - \lambda x - \nu$, où [5] $\lambda = (3x_P^2 + a)/(2y_P)$ et $\nu = y_P - \lambda x_P$.

- Commençons par l'ordre en Q, qui est plus simple à déterminer. L'uniformisante en Q est $(x, y) \mapsto x - x_Q$.

$$\delta_{P,Q}(x, y) = y - \lambda x - \nu = y - y_Q - \lambda(x - x_Q) = (x - x_Q)\left(\frac{y - y_Q}{x - x_Q} - \lambda\right).$$

Pour réduire la fraction, on peut réécrire l'équation de la courbe $y^2 = x^3 + ax + b$ en constatant que $y_Q^2 = x_Q^3 + ax_Q + b$, et alors, en faisant la différence de ces

5. Pour rappel, l'équation de la courbe elliptique E est $y^2 = x^3 + ax + b$, et a est utilisé dans la formule de doublement.

deux équations :

$$\begin{aligned}(y-y_Q)(y+y_Q) &= x^3 - x_Q^3 + a(x-x_Q),\\ &= (x-x_Q)^3 + 3x^2x_Q - 3xx_Q^2 + a(x-x_Q),\\ &= (x-x_Q)^3 + 3x_Q(x^2 - xx_Q) + a(x-x_Q),\\ &= (x-x_Q)^3 + 3x_Q\Big((x-x_Q)^2 + xx_Q - x_Q^2\Big) + a(x-x_Q),\\ &= (x-x_Q)^3 + 3x_Q(x-x_Q)^2 + 3xx_Q^2 - 3x_Q^3 + a(x-x_Q),\\ &= (x-x_Q)^3 + 3x_Q(x-x_Q)^2 + 3x_Q^2(x-x_Q) + a(x-x_Q),\\ &= (x-x_Q)^3 + 3x_Q(x-x_Q)^2 + (3x_Q^2+a)(x-x_Q).\end{aligned}$$

Ce qui implique

$$\frac{y-y_Q}{x-x_Q} = \frac{(x-x_Q)^2 + 3x_Q(x-x_Q) + 3x_Q^2 + a}{y+y_Q} = \frac{(x-x_Q)^2 + 3x_Qx + a}{y+y_Q},$$

et ainsi,

$$\delta_{P,Q}(x,y) = (x-x_Q)\left(\frac{(x-x_Q)^2 + 3x_Qx + a}{y+y_Q} - \lambda\right), \tag{II.2.7}$$

où la fonction entre parenthèse toute à droite a une valeur finie non nulle en Q. Donc Q est un zéro d'ordre 1 de $\delta_{P,Q}$.

- L'uniformisante en P est $(x,y) \mapsto x - x_P$ et si on fait le même raisonnement que précédemment en (II.2.7), on obtient

$$\begin{aligned}\delta_{P,Q}(x,y) &= (x-x_P)\left(\frac{(x-x_P)^2 + 3x_Px + a}{y+y_P} - \lambda\right),\\ &= \frac{x-x_P}{y+y_P}\Big((x-x_P)^2 + 3x_Px + a - \lambda(y+y_P)\Big),\\ &= \frac{x-x_P}{y+y_P}\Big((x-x_P)^2 + (3x_Px - 3x_P^2) + (3x_P^2+a) - \lambda(y+y_P)\Big),\\ &= \frac{x-x_P}{y+y_P}\Big((x-x_P)^2 + 3x_P(x-x_P) + 2\lambda y_P - \lambda(y+y_P)\Big),\\ &= \frac{x-x_P}{y+y_P}\Big((x-x_P)^2 + 3x_P(x-x_P) - \lambda(y-y_P)\Big),\\ &= \frac{(x-x_P)^2}{y+y_P}\left((x-x_P) + 3x_P - \lambda\frac{(x-x_P)^2 + 3x_Px + a}{y+y_P}\right),\end{aligned}$$

ce qui implique que P est un zéro d'ordre 2 de $\delta_{P,Q}$.

- En P_∞, l'uniformisante est $(x, y) \mapsto x/y$. Alors $\delta_{P,Q}$ s'écrit

$$\delta_{P,Q}(x,y) = \left(\frac{x}{y}\right)^{-3} \frac{yx^3 - \lambda x^4 - \nu x^3}{y^3} = \left(\frac{x}{y}\right)^{-3} \frac{yx^3 - \lambda x^4 - \nu x^3}{y(x^3 + ax + b)},$$

où $(x, y) \mapsto (yx^3 - \lambda x^4 - \nu x^3)/(y(x^3 + ax + b))$ vaut 1 quand évaluée en P_∞. Ainsi P_∞ est un pôle d'ordre 3 de $\delta_{P,Q}$.

— Dans le cas où la droite $\delta_{P,Q}$ n'est tangente ni en P ni en Q, elle intersecte la courbe E en un troisième point $-P-Q$. Ce cas est plus simple que le précédent : en réécrivant l'équation de la courbe pour chacun des trois points de sorte à avoir une égalité comme en (II.2.7), on trouve que $\mathrm{ord}_P(\delta_{P,Q}) = \mathrm{ord}_Q(\delta_{P,Q}) = \mathrm{ord}_{-P-Q}(\delta_{P,Q}) = 1$. Idem pour P_∞ : c'est un pôle d'ordre 3 de $\delta_{P,Q}$.

En résumé, pour $P \neq P_\infty$ et $Q \neq P_\infty$ tels que $P \neq \pm Q$, le diviseur de la droite passant par P et Q est

$$\mathrm{div}(\delta_{P,Q}) = (P) + (Q) + (-P - Q) - 3(P_\infty).$$

Les seules droites particulières que l'on n'a pas encore traitées sont celles tangentes aux points d'inflexion : Si P est un point d'inflexion de E alors la tangente à E en P admet seulement P comme zéro d'ordre 3. Le diviseur de cette tangente est $3(P) - 3(P_\infty)$. Il en résulte qu'un point d'inflexion est un point d'ordre 3.

On remarque que pour deux points distincts $P \neq P_\infty$ et $Q \neq P_\infty$, il n'y a pas de droite dont le support du diviseur contient les points P, Q et $P + Q$. Il est cependant facile de trouver une telle fonction (qui ne représentera donc pas une droite) en remarquant $\mathrm{div}(\delta_{P,Q}) = (P) + (Q) + (-P - Q) - 3(P_\infty)$ et $\mathrm{div}(\nu_{P+Q}) = (P + Q) + (-P - Q) - 2(P_\infty)$: la différence de ces diviseurs est

$$\mathrm{div}\left(\frac{\delta_{P,Q}}{\nu_{P+Q}}\right) = \mathrm{div}(\delta_{P,Q}) - \mathrm{div}(\nu_{P+Q}) = (P) + (Q) - (P + Q) - (P_\infty). \qquad \text{(II.2.8)}$$

II.3 Couplages

Un *couplage e d'ordre ℓ* est une application bilinéaire non-dégénérée entre trois groupes cycliques $\mathbb{G}_1$, $\mathbb{G}_2$ et $\mathbb{G}_T$ d'ordre (premier) ℓ

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

Dans la littérature cryptographique, les groupes d'un couplage sont notés de manière multiplicative, c'est ce que l'on va faire ici. Pour simplifier la distinction entre les éléments de $\mathbb{G}_1$ et ceux de $\mathbb{G}_2$, on décore les seconds d'un tilde.

Soient $g \in \mathbb{G}_1$ et $\tilde{g} \in \mathbb{G}_2$ deux générateurs. Dire que e n'est pas dégénérée signifie que $e(g, \tilde{g})$ est d'ordre ℓ dans $\mathbb{G}_T$, *i.e.* est un générateur. La bilinéarité de e se traduit par :

$$\forall a \in \mathbb{Z}, \forall b \in \mathbb{Z}, \quad e(g^a, \tilde{g}^b) = e(g, \tilde{g})^{ab}.$$

Échange de clé tripartite

L'une des premières constructions cryptographiques faite à l'aide des couplages est l'échange de clé tripartite dû à Joux en 2000 [Jou04].

Dans un échange de clé entre deux protagonistes, chaque correspondant envoie sa clé éphémère et les deux obtiennent ainsi un secret partagé : c'est l'échange de clé de Diffie et Hellman. À trois protagonistes, il est possible d'obtenir un secret commun mais il faut alors envoyer deux messages à une des deux autres personnes.

Détaillons : soit $\langle g \rangle$ un groupe d'ordre premier p. Nos trois protagonistes, appelons les Alice, Bob et Charly, tirent chacun un entier modulo p aléatoirement (sous-entendu de façon uniforme), respectivement noté a, b et c. Considérons le cas d'Alice. Alice va envoyer à Bob l'élément g^a et recevoir g^c de Charly. Elle va ensuite envoyer g^{ac} à Bob et recevoir g^{bc} de la part de Charly. Enfin, elle peut calculer g^{abc}, secret commun partagé avec les deux autres.

Avec l'échange de Joux, si le couplage est symétrique, c'est-à-dire $\mathbb{G}_1 = \mathbb{G}_2$, alors on procède comme suit : Alice diffuse g^a à Bob et Charly et récupère g^b de Bob et g^c de Charly. Ensuite elle calcule $e(g^b, g^c)^a$, qui est bien un secret commun aux trois car, par bilinéarité

$$e(g^a, g^b)^c = e(g^c, g^a)^b = e(g^b, g^c)^a = e(g, g)^{abc}.$$

Attaque MOV/Frey-Rück

La première utilisation des couplages en cryptographie a servi à transférer le problème du logarithme discret d'une courbe elliptique vers un corps fini. C'est au début des années 1990 que Menezes, Okamoto et Vanstone détaillent cette attaque [MVO91]. Si $E/\mathbb{F}_q$, où $\mathbb{F}_q$ est de caractéristique $p \geqslant 5$, est un courbe supersingulière, alors son degré de plongement est $k \leqslant 6$ [MVO91]. Soit $P \in E(\mathbb{F}_q)$ d'ordre ℓ premier divisant $\#E(\mathbb{F}_q)$. Il existe un couplage e_ℓ (le couplage de Weil que nous abordons juste après) de $E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell]$ dans $\mu_\ell \subset \mathbb{F}_{q^k}$, le groupe des racines ℓ-ième de l'unité. Pour transférer la recherche du logarithme discret, admettons $Q = [x]P$, il suffit de fixer $R \in E(\mathbb{F}_{q^k})[\ell]$ tel que $R \notin \langle P \rangle$ et alors

$$e_\ell(Q, R) = e_\ell([x]P, R) = e_\ell(P, R)^x.$$

Ainsi $x = \log_P Q = \log_{e_\ell(P,R)} e_\ell(Q, R)$. Les auteurs préconisaient donc d'éviter les courbes avec un petit degré de plongement, car le calcul du logarithme discret est en général plus facile dans les corps finis. En réalité, c'est plutôt un équilibre qu'il faut trouver entre la difficulté des divers DLP si on souhaite utiliser l'application de couplage. Mais, ce qui est certain, c'est que les courbes supersingulières ne semblent pas être les plus convenables lorsque l'on veut instancier des couplages sur des courbes.

Une extension de cette attaque a été faite peu de temps après par Frey et Rück [FR94], adaptée au couplage de Tate, couplage que l'on étudie en Sous-Section II.3.2.

II.3.1 Couplage de Weil

Nous définissons ici le couplage de Weil sur les courbes elliptiques définies sur un corps fini. Soit $E/\mathbb{F}_q$ une courbe elliptique. Rappelons qu'un diviseur de E (vu en Section II.2) est une somme formelle, finie, de points de E :

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

De plus, un diviseur D est principal (*i.e.* est le diviseur d'une fonction sur E) si, et seulement si, $\sum_P n_P = 0$ dans $\mathbb{Z}$ et $\sum_P [n_P]P = P_\infty$ sur E.

Définition II.3.1 (Couplage de Weil [Sil09, Section III.8])**.** Soient k, ℓ deux entiers et $E/\mathbb{F}_q$ une courbe elliptique de degré de plongement k par rapport à ℓ. Soient P et Q deux points de $E(\mathbb{F}_{q^k})[\ell]$. Soient D_P et D_Q deux diviseurs de support disjoint vérifiant

$D_P \sim (P) - (P_\infty)$ et $D_Q \sim (Q) - (P_\infty)$. Soient f et g deux fonctions sur E telles que $\mathrm{div}(f) = \ell D_P$ et $\mathrm{div}(g) = \ell D_Q$. Le *couplage de Weil d'ordre* ℓ, noté w_ℓ, est l'application

$$w_\ell : E(\mathbb{F}_{q^k})[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^k}$$

définie par

$$w_\ell(P, Q) = \frac{f(D_Q)}{g(D_P)}.$$

L'application w_ℓ est bilinéaire et non-dégénérée.

Revenons sur cette définition. La construction d'une courbe elliptique $E/\mathbb{F}_q$ de degré de plongement k par rapport à un entier ℓ sera abordée plus loin. Supposons pour l'instant que nous avons une telle courbe E. Les points P et Q sont des points de ℓ-torsion, *i.e.* $[\ell]P = [\ell]Q = P_\infty$. Concernant les diviseurs D_P et D_Q, il n'est pas possible de choisir $D_P = (P) - (P_\infty)$ et $D_Q = (Q) - (P_\infty)$ car ils ne sont pas de support disjoint. Il faut donc au moins en changer un des deux. Changeons D_Q. Pour cela, nous pouvons prendre une fonction h sur E et poser $D_Q = (Q) - (P_\infty) + \mathrm{div}(h)$. La pratique courante est de prendre $D_Q = (Q + R) - (R)$, pour $R \in E(\mathbb{F}_{q^k})$ aléatoire [6]. Cela revient à choisir la fonction h comme le quotient $\nu_{Q+R}/\delta_{Q,R}$ de la verticale ν_{Q+R} passant par $Q + R$ et de la droite $\delta_{Q,R}$ passant par les points Q et R :

$$\begin{aligned}
\mathrm{div}(h) &= \mathrm{div}(\nu_{Q+R}) - \mathrm{div}(\delta_{Q,R}), \\
&= (Q + R) + (-Q - R) - 2(P_\infty) - (Q) - (R) - (-Q - R) + 3(P_\infty), \\
&= D_Q - ((Q) - (P_\infty)).
\end{aligned}$$

Les diviseurs ℓD_P et ℓD_Q sont principaux car leur degré est nul et leur somme vaut P_∞ (les points P et Q sont dans la ℓ-torsion). Ainsi il existe deux fonctions f et g sur E telles que $\mathrm{div}(f) = \ell D_P$ et $\mathrm{div}(g) = \ell D_Q$. Le problème est alors : comment calculer f et g ?

Commençons par le cas de f où $\mathrm{div}(f) = \ell(P) - \ell(P_\infty)$. Ce qu'il suffirait de trouver, c'est une suite de fonctions, disons de terme général $f_{m,P}$, et telle que

$$\mathrm{div}(f_{\ell,P}) = \ell D_P = \ell(P) - \ell(P_\infty).$$

L'idée est de construire cette suite de proche en proche, de « multiplier le terme précédent par P ». Cependant, $[m]P \neq P_\infty$ pour $m < \ell$, donc le diviseur $m(P) - m(P_\infty)$ n'est pas

6. On a de grandes chances que ce R ne pose pas de problème, *e.g.* R est différent de P_∞.

principal ! Il faut donc ajouter (au moins) un point au support de ce diviseur de sorte qu'il soit principal. En ajoutant le point $[m]P$ au support, et en changeant les multiplicités pour avoir un diviseur de degré nul, nous remarquons que, pour $m \geqslant 2$, le diviseur $m(P) - ([m]P) - (m-1)(P_\infty)$ est principal. Donc il existe des fonctions $f_{m,P}$ vérifiant

$$\mathrm{div}(f_{m,P}) = m(P) - ([m]P) - (m-1)(P_\infty)$$

et $\mathrm{div}(f_{\ell,P}) = \ell D_P$. Il reste à voir comment calculer le premier terme $f_{2,P}$ et passer du terme m au terme $m+1$.

Le diviseur de $f_{2,P}$ doit valoir $2(P) - ([2]P) - (P_\infty)$. La tangente $\delta_{P,P}$ en P à la courbe E a un zéro d'ordre 2 en P :

$$\mathrm{div}(\delta_{P,P}) = 2(P) + (-[2]P) - 3(P_\infty).$$

Il reste alors à ajouter $-(-[2]P) - ([2]P) + 2(P_\infty)$, ce qui correspond au diviseur $\mathrm{div}(1/\nu_{[2]P})$ de l'inverse de la droite verticale $\nu_{[2]P}$ passant par $[2]P$. Ainsi, en posant $f_{2,P} = \delta_{P,P}/\nu_{[2]P}$, nous obtenons bien

$$\mathrm{div}(f_{2,P}) = \mathrm{div}(\delta_{P,P}) - \mathrm{div}(\nu_{[2]P}) = 2(P) - ([2]P) - (P_\infty). \tag{II.3.9}$$

Si $f_{m,P}$ et $f_{m+1,P}$ sont construites alors la différence de ces diviseurs est

$$\begin{aligned}
\mathrm{div}(f_{m+1,P}) - \mathrm{div}(f_{m,P}) &= (m+1)(P) - ([m+1]P) - m(P) + ([m]P) - (P_\infty), \\
&= (P) + ([m]P) - ([m+1]P) - (P_\infty), \\
&= \mathrm{div}(\delta_{P,[m]P}) - \mathrm{div}(\nu_{[m+1]P}),
\end{aligned}$$

où $\delta_{P,[m]P}$ est la droite passant par les points P et $[m]P$ et $\nu_{[m+1]P}$ est la droite verticale passant par $[m+1]P$ (on s'est aidé ici de l'Équation (II.2.8) en y substituant Q par $[m]P$). Ainsi, sachant $f_{m,P}$, on peut calculer $f_{m+1,P}$ comme

$$f_{m+1,P} = f_{m,P}\frac{\delta_{P,[m]P}}{\nu_{[m+1]P}}. \tag{II.3.10}$$

Cependant, lorsque $m+1 = \ell$, il n'est pas possible de tracer la verticale passant par $[\ell]P = P_\infty$. La formule précédente (II.3.10) est donc valable pour $2 \leqslant m < \ell - 1$. Pour

voir comment passer de $f_{\ell-1,P}$ à $f_{\ell,P}$, nous calculons la différence de diviseurs :

$$\begin{aligned}\ell(P) - \ell(P_\infty) - \mathrm{div}(f_{\ell-1,P}) &= \ell(P) - \ell(P_\infty) - (\ell-1)(P) + ([\ell-1]P) + (\ell-2)(P_\infty),\\ &= (P) + ([\ell-1]P) - 2(P_\infty),\\ &= (P) + (-P) - 2(P_\infty),\\ &= \mathrm{div}(\nu_P),\end{aligned}$$

où ν_P est la droite verticale passant par P. Il en résulte

$$f_{\ell,P} = f_{\ell-1,P}\nu_P \quad \Longrightarrow \quad \mathrm{div}(f_{\ell,P}) = \ell(P) - \ell(P_\infty). \tag{II.3.11}$$

Résumons notre réflexion : nous prenons $f = f_{\ell,P}$, où la suite $(f_{m,P})_{2\leqslant m\leqslant \ell}$ est définie [7] par

$$f_{m,P} = \begin{cases} \delta_{P,P}/\nu_{[2]P} & \text{si } m = 2, \quad \text{vu en (II.3.9)}\\ f_{m-1,P}\delta_{P,[m-1]P}/\nu_{[m]P} & \text{si } 2 < m < \ell, \quad \text{vu en (II.3.10)}\\ f_{\ell-1,P}\nu_P & \text{si } m = \ell. \quad \text{vu en (II.3.11)}\end{cases} \tag{II.3.12}$$

Pour le cas de g où $\mathrm{div}(g) = \ell D_Q$, nous avons vu plus haut qu'il est possible de prendre $D_Q = (Q) - (P_\infty) + \mathrm{div}(\nu_{Q+R}/\delta_{Q,R})$, pour un $R \in E(\mathbb{F}_{q^k})$ aléatoire. Cela donne :

$$\begin{aligned}\ell D_Q &= \ell(Q) - \ell(P_\infty) + \ell\,\mathrm{div}(\nu_{Q+R}/\delta_{Q,R}),\\ &= \mathrm{div}(f_{\ell,Q}) + \mathrm{div}((\nu_{Q+R}/\delta_{Q,R})^\ell),\\ &= \mathrm{div}(f_{\ell,Q}(\nu_{Q+R}/\delta_{Q,R})^\ell),\end{aligned}$$

où la fonction $f_{\ell,Q}$ est calculée comme précédemment, en remplaçant P par Q dans la définition de la suite (II.3.12). Ainsi, nous prenons

$$g = f_{\ell,Q}\left(\frac{\nu_{Q+R}}{\delta_{Q,R}}\right)^\ell. \tag{II.3.13}$$

La dernière étape est le calcul du couplage :

$$w_\ell(P,Q) = \frac{f(D_Q)}{g(D_P)}.$$

7. Remarquons que si P est d'ordre 2, alors la fonction ν_P est de diviseur $2(P) - 2(P_\infty)$.

Évaluer une fonction f sur un diviseur $D = \sum n_P(P)$, c'est calculer

$$f(D) := \prod f(P)^{n_P}.$$

Ici, les supports de D_P et D_Q sont disjoints, ainsi aucun point du support de D_Q n'est un zéro ou un pôle de la fonction f : l'élément $f(D_Q)$ n'est ni nul ni infini ; il en est de même pour $g(D_P)$. Pour évaluer une fonction en P_∞, nous pouvons utiliser son homogénéisation projective et l'évaluer en $P_\infty = [0 : 1 : 0]$. Ou bien, nous pouvons aussi changer D_P en tirant un $S \in E(\mathbb{F}_{q^k})$ aléatoire et en posant $D_P = (S + P) - (S)$. La construction de la fonction f de diviseur $\ell D_P = \ell(S + P) - \ell(S)$ se fait alors de la même façon que pour g en (II.3.13).

Le résultat important qui fait fonctionner la bilinéarité de ce couplage est le *théorème de réciprocité de Weil*.

Théorème II.3.1 (Réciprocité de Weil)**.** *Soient f et g deux fonctions sur une courbe telles que* $\mathrm{div}(f)$ *et* $\mathrm{div}(g)$ *soient disjoints. Alors*

$$f(\mathrm{div}(g)) = g(\mathrm{div}(f)).$$

II.3.2 Couplage de Tate-Lichtenbaum

Donnons maintenant la définition d'un autre couplage : celui de Tate-Lichtenbaum.

Définition II.3.2 (Couplage de Tate-Lichtenbaum [Sil09, Section XI.9])**.** Soient k, ℓ deux entiers et $E/\mathbb{F}_q$ une courbe elliptique de degré de plongement k par rapport à ℓ. Soient $P \in E(\mathbb{F}_{q^k})[\ell]$ et $Q \in E(\mathbb{F}_{q^k})$. Soit f une fonction sur E de diviseur $\ell(P) - \ell(P_\infty)$ et soit D_Q un diviseur équivalent à $(Q) - (P_\infty)$ et de support disjoint de $\mathrm{div}(f)$. Le *couplage de Tate-Lichtenbaum d'ordre* ℓ, noté t_ℓ, est l'application

$$t_\ell : E(\mathbb{F}_{q^k})[\ell] \times E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^\times/(\mathbb{F}_{q^k}^\times)^\ell$$

définie par

$$t_\ell(P, Q) = f(D_Q).$$

L'application t_ℓ est bilinéaire et non-dégénérée.

Nous avons déjà vu plus haut comment trouver une fonction f de diviseur $\ell(P) - \ell(P_\infty)$. Remarquons de la définition du couplage de Tate-Lichtenbaum que le point Q est vu

comme un représentant de sa classe d'équivalence dans $E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k})$.

L'ensemble $E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k})$ est un groupe (quotient) où $\ell E(\mathbb{F}_{q^k})$ est défini comme

$$\ell E(\mathbb{F}_{q^k}) = \{[\ell]P : P \in E(\mathbb{F}_{q^k})\}.$$

Dans nos cas d'étude, nous supposons que ℓ divise $\#E(\mathbb{F}_q)$ une fois et ℓ^2 divise $\#E(\mathbb{F}_{q^k})$ une fois aussi. Dans ces conditions,

— il y a $\#E(\mathbb{F}_{q^k})/\ell^2$ éléments dans $\ell E(\mathbb{F}_{q^k})$, car deux points dont la différence est dans la ℓ-torsion ont le même multiple ℓ,

$$\forall P, Q \in E(\mathbb{F}_{q^k}), \quad P - Q \in E[\ell] \Leftrightarrow [\ell]P = [\ell]Q;$$

— il y a ℓ^2 classes d'équivalence dans $E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k})$, deux points sont dans la même classe s'ils diffèrent d'un multiple de ℓ,

$$\forall P, Q \in E(\mathbb{F}_{q^k}), \quad P \equiv Q \Leftrightarrow P - Q \in \ell E(\mathbb{F}_{q^k});$$

— l'intersection $\ell E(\mathbb{F}_{q^k}) \cap E[\ell]$ est $\{P_\infty\}$, donc chaque point de $E[\ell]$ représente une classe de $E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k})$.

Ainsi, dans la définition du couplage de Tate-Lichtenbaum, nous choisissons pour les points du second groupe des éléments de $E[\ell]$, chacun représentant sa classe d'équivalence dans le quotient $E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k})$. Cela ressemble plus à la définition du couplage de Weil.

Exactement comme le second groupe de ce couplage, les éléments de l'image du couplage $\mathbb{F}_{q^k}^\times/(\mathbb{F}_{q^k}^\times)^\ell$ sont des classes d'équivalence. Le groupe $(\mathbb{F}_{q^k}^\times)^\ell$ est défini par

$$(\mathbb{F}_{q^k}^\times)^\ell = \{a^\ell : a \in \mathbb{F}_{q^k}^\times\}.$$

Ainsi, deux éléments de $\mathbb{F}_{q^k}^\times$ sont dans la même classe d'équivalence dans le groupe quotient $\mathbb{F}_{q^k}^\times/(\mathbb{F}_{q^k}^\times)^\ell$ si, et seulement si, leur différence est une puissance ℓ-ième.

Le couplage de Tate-Lichtenbaum a cette propriété indésirable que son image représente une classe d'équivalence, au lieu d'être une valeur unique. Un attribut nécessaire d'un couplage pour être utile en cryptographie est que les différentes parties calculent le même élément grâce à la bilinéarité, et pas seulement des éléments dans une même classe d'équivalence. Pour surmonter cet écueil, il suffit de composer t_ℓ avec l'élévation à la puissance $\#E(\mathbb{F}_{q^k})/\ell = (q^k - 1)/\ell$ afin de simplifier les puissances ℓ-ième et d'envoyer l'image du couplage sur une valeur exacte : une racine ℓ-ième de l'unité. Ce nouveau couplage est

alors dit *réduit*.

Définition II.3.3 (Couplage de Tate-Lichtenbaum réduit)**.** Avec les mêmes notations que la définition II.3.2, le *couplage de Tate-Lichtenbaum réduit d'ordre* ℓ, noté T_ℓ, est l'application

$$T_\ell : E(\mathbb{F}_{q^k})[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^k}$$

définie par

$$T_\ell(P,Q) = t_\ell(P,Q)^{\frac{q^k-1}{\ell}} = f(D_Q)^{\frac{q^k-1}{\ell}}.$$

L'application T_ℓ est bilinéaire et non-dégénérée.

II.3.3 Algorithme de Miller

Aussi bien pour le couplage de Weil que le couplage de Tate, on a besoin de construire des fonctions de diviseur $\ell(P) - \ell(P_\infty)$, pour un point P dans la ℓ-torsion. Pour cela, on a vu qu'on génère une suite de fonctions rationnelles $(f_{m,P})_m$ définie, pour $2 < m < \ell$, par

$$f_{m,P} = f_{m-1,P}\frac{\delta_{P,[m-1]P}}{\nu_{[m]P}}.$$

En passant de $f_{m,P}$ à $f_{m+1,P}$, on ajoute un zéro en P et un pôle en P_∞ à la fonction en la multipliant par des fonctions linéaires (ou leurs inverses), alors que l'on double le nombre de zéros en P et de pôles en P_∞ de la fonction $f_{m,P}$ si on l'élève au carré. En effet, en observant que $\mathrm{div}(f_{m,P}) = m(P) - ([m]P) - (m-1)(P_\infty)$, alors

$$\mathrm{div}(f_{m,P}^2) = 2m(P) - 2([m]P) - 2(m-1)(P_\infty),$$

ce qui est presque le diviseur de $f_{2m,P}$,

$$\mathrm{div}(f_{2m,P}) = 2m(P) - ([2m]P) - (2m-1)(P_\infty).$$

La différence de ces deux diviseurs est $\mathrm{div}(f_{2m,P}) - \mathrm{div}(f_{m,P}^2) = 2([m]P) - ([2m]P) - (P_\infty)$, qui n'est autre que le diviseur $\mathrm{div}(\delta_{[m]P,[m]P}) - \mathrm{div}(\nu_{[2m]P})$ de la tangente $\delta_{[m]P,[m]P}$ en $[m]P$ quotientée par la verticale $\nu_{[2m]P}$ passant par $2[m]P$ (ici aussi, on s'est aidé ici de l'Équation (II.2.8) en y substituant P et Q par $[m]P$). Ainsi, on peut aller de $f_{m,P}$ à $f_{2m,P}$ grâce à

$$f_{2m,P} = f_{m,P}^2\frac{\delta_{[m]P,[m]P}}{\nu_{[2m]P}}.$$

C'est donc en remarquant qu'il est possible de passer de $f_{m,P}$ à $f_{m+1,P}$ ou $f_{2m,P}$ que Miller [Mil04] à décrit un algorithme calculant la fonction $f_{\ell,P}$, imitant la méthode du *square-and-multiply*. Cet algorithme, dit aussi *boucle de Miller*, a une complexité en $O(\log \ell)$. La seconde grande contribution de Miller est de constater que la fonction $f_{\ell,P}$ est le quotient de polynômes de degré ℓ, ce qui est impossible à stocker lorsque ℓ est de taille cryptographique. Il propose donc d'évaluer la fonction au fur et à mesure, gardant ainsi en mémoire un point de $\mathbb{F}_{q^k}$ au lieu d'une fonction rationnelle de $\mathbb{F}_{q^k}(E)$. L'Algorithme 3 résume la boucle de Miller, où on remarque que l'expansion binaire de $\ell = (\ell_{n-1} \dots \ell_1 \ell_0)_2$ régie les étapes de l'algorithme de la même façon qu'un *double-and-add* pour les points de la courbe elliptique (*e.g.* Algorithme 2).

Algorithme 3 Boucle de Miller

Entrée : $\ell = (\ell_{n-1} \dots \ell_1 \ell_0)_2$ avec $\ell_{n-1} = 1$, $P \in E(\mathbb{F}_{q^k})[\ell]$, $D_Q \sim (Q) - (P_\infty)$ pour $Q \in E(\mathbb{F}_{q^k})[\ell]$ tel que D_Q est de support disjoint de tous les $\mathrm{div}(f_{m,P})$ pour $m \leqslant \ell$
Sortie : $f_{\ell,P}(D_Q) \in \mathbb{F}_{q^k}$

1: $R \leftarrow P$ **et** $f \leftarrow 1$
2: **pour** i **allant de** $n-2$ **à** 0 **faire**
3: Calculer les droites $\delta_{R,R}$ et $\nu_{[2]R}$ pour le doublement de R
4: $f \leftarrow f^2 \frac{\delta_{R,R}}{\nu_{[2]R}}(D_Q)$
5: $R \leftarrow [2]R$
6: **si** $\ell_i = 1$ **alors**
7: Calculer les droites $\delta_{P,R}$ et ν_{P+R} pour l'addition de P et R
8: $f \leftarrow f \frac{\delta_{P,R}}{\nu_{P+R}}(D_Q)$
9: $R \leftarrow P + R$
10: **fin si**
11: **fin pour**
12: **retourner** $f_{\ell,P}(D_Q) = f$

II.3.4 Courbes elliptiques à couplage

Pour que les couplages sur courbes elliptiques soient utilisables en pratique, il faut essentiellement deux choses : des courbes appropriées supportant l'opération de couplage et des algorithmes pour calculer efficacement le couplage de deux points.

Remarquons d'abord qu'une implémentation efficace de la loi de groupe de la courbe impacte grandement la vitesse d'exécution d'un couplage. Dans notre cas, pour les courbes à couplage, le système de coordonnées permettant les opérations les plus rapides sur le modèle de Weierstrass est le système de coordonnées Jacobiennes [BL07a]. Ces coordonnées

sont une modification des coordonnées projectives. Un point (x, y) de $E : y^2 = x^3 + ax + b$ est représenté par $[X : Y : Z]$ satisfaisant $Y^2 = X^3 + aXZ^4 + bZ^6$ et $(x, y) = (X/Z^2, Y/Z^3)$. La relation d'équivalence « projective » est ici $[X : Y : Z] = [\lambda^2 X : \lambda^3 Y : \lambda Z]$ pour tout λ non nul. Les formules sont données par Bernstein et Lange [BL07a], et ont été introduites par Chudnovsky et Chudnovsky [CC86].

Choisir une courbe à couplage

Une courbe E compatible avec le couplage, dite *pairing-friendly* en anglais, remplit deux conditions [FST10] :

— il existe un premier $\ell \geqslant \sqrt{q}$ divisant $\#E(\mathbb{F}_q)$;

— le degré de plongement k de la courbe par rapport à ℓ est inférieur à $\log_2(\ell)/8$.

Le premier ℓ est l'ordre d'un sous-groupe de $E(\mathbb{F}_q)$, plus précisément, on souhaite que ℓ ne divise qu'une seule fois $\#E(\mathbb{F}_q)$, comme cela on peut utiliser la théorie développée précédemment. On ne sait qu'utiliser des algorithmes génériques de complexité $O(\sqrt{\ell})$ pour calculer les logarithmes discrets sur la courbe. Ainsi, pour 128 bits de sécurité, il faut que ℓ ait au moins 256 bits et alors $k \leqslant 32$ (par le second point). Nous discutons de la sécurité des courbes à couplage en Section II.6.

Un résultat de Balasubramanian et Koblitz [BK98] indique que pour une courbe elliptique $E/\mathbb{F}_q$ tirée aléatoirement, le degré de plongement k par rapport à un diviseur r de $\#E(\mathbb{F}_q)$ est proportionnel à r, *i.e.* $k \approx r$. Si r est de taille cryptographique (*e.g.* $r \sim 2^{256}$), alors les opérations dans le corps fini $\mathbb{F}_{q^k}$, contenant l'image du couplage, sont impossibles. La stratégie pour obtenir des courbes à couplage est donc de les construire, ce que l'on étudie dans les sections suivantes.

On a vu que la borne de Hasse, au Théorème II.1.6, donne $q \sim \#E(\mathbb{F}_q)$. Une quantité couramment utilisée pour comparer la taille d'un diviseur ℓ de $\#E(\mathbb{F}_q)$ avec celle de q est la valeur-ρ :

$$\rho := \frac{\log q}{\log \ell}.$$

Ce que mesure ce ρ, c'est l'efficacité de la représentation des éléments du groupe d'ordre ℓ, à savoir le groupe $E(\mathbb{F}_q)[\ell]$. En effet, dans le meilleur des cas $\#E(\mathbb{F}_q) = \ell$ et [8] $\rho = 1$, les éléments du groupe d'ordre ℓ sont représentés en mémoire en $\log \ell$ bits. Une des deux conditions pour qu'une courbe soit compatible avec le couplage est exactement $\rho \leqslant 2$, *i.e.* les éléments du groupe d'ordre ℓ sont représentés en mémoire en au plus $2 \log \ell$ bits.

8. En réalité, $q \sim \#E(\mathbb{F}_q) = \ell$ et ainsi $\rho \to 1$ lorsque $q \to +\infty$.

En fait, des méthodes connues [CP01 ; DEM05] permettent de construire des courbes à couplages, où q est un polynôme de degré 2 en ℓ, dans ce cas $\rho = 2$. Les courbes Miyaji-Nakabyashi-Takano [MNT01] et Barreto-Naehrig [BN06] sont aussi construites à l'aide de polynômes et leur ρ est de 1. Malheureusement, ces courbes MNT et BN ne sont pas sûres pour une taille de q raisonnable, comme nous le verrons plus loin. Ainsi, on cherche à faire mieux en construisant des courbes dont ρ est proche de 1 pour le niveau de sécurité souhaitée.

Choisir un algorithme pour calculer le couplage

Nous avons vu plus haut deux couplages sur courbe elliptique, celui de Weil et celui de Tate. Ce qui les différencie, c'est le calcul d'une seconde boucle de Miller pour Weil ou d'une exponentiation dans $\mathbb{G}_T$ pour Tate (plus précisément le couplage de Tate-Lichtenbaum réduit en Définition II.3.3). L'exponentiation dans le couplage de Tate, dite *exponentiation finale*, est moins coûteuse que le calcul d'une boucle de Miller [SCA06] à 128 bits de sécurité et permet aussi diverses améliorations.

Dans le couplage de Tate, il est possible de remplacer le diviseur D_Q par le point directement Q [Bar+02]. En effet, tant que $k > 1$ et que P et Q sont linéairement indépendants,

$$f_{\ell,P}(D_Q)^{\frac{q^k-1}{\ell}} = f_{\ell,P}(Q)^{\frac{q^k-1}{\ell}}.$$

Cela permet de directement évaluer la fonction définie par la récurrence (II.3.12) en Q au lieu de devoir trouver un diviseur D_Q approprié.

Une autre contribution de Barreto *et al.* [Bar+02] est ce que l'on appelle *l'élimination de dénominateur*. Soit $E/\mathbb{F}_q$ une courbe ordinaire de degré de plongement k par rapport à ℓ premier divisant une fois $\#E(\mathbb{F}_q)$ (la situation habituelle dans laquelle on se place). Supposons que $d \in \{2, 4, 6\}$ divise k. Alors, en prenant $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$ et $G_2 = \Psi(E'(\mathbb{F}_{q^{k/d}})[\ell])$, où E' a pour tordue de degré d la courbe E et Ψ est l'isomorphisme de E' vers E, les abscisses des points $Q \in \mathbb{G}_2$ sont dans un sous-corps strict de $\mathbb{F}_{q^k}$. En remarquant que $q^e - 1$ divise $(q^k - 1)/\ell$ pour tout e diviseur propre de k, il est alors possible d'omettre les dénominateur dans la boucle de Miller, car les dénominateurs sont des fonctions de $\mathbb{F}_q(E)$, représentant des droites verticales, évaluées en $x_Q \in \mathbb{F}_{q^{k/2}}$.

Entre 2000 et 2010, plusieurs résultats [DL03 ; Bar+07 ; HSV06] ont indiqué qu'il était possible de raccourcir la longueur de la boucle de Miller. Ces idées ont culminé avec la définition de couplage optimal de Vercauteren [Ver10]. La longueur de la boucle d'un

couplage optimal est au moins $\log_2 \ell/\varphi(k)$, où φ est l'indicatrice d'Euler. Cette technique fait usage des faits que certains endomorphismes de la courbe calculent des multiples $[\lambda]P$ de point P, par exemple le Frobenius sur le sous-groupe de trace nulle avec $\pi(P) = [q]P$, et que, étant donnés deux couplages bilinéaires sur E, leur produit ou leur quotient est encore un couplage bilinéaire sur E. Cela implique d'inverser les rôles de P et Q dans la boucle de Miller et de calculer $f_{T,Q}(P)$ pour un certain T et $Q \in E[\ell] \cap \mathrm{Ker}(\pi - [q])$ et $P \in E(\mathbb{F}_q)[\ell]$.

Une dernière amélioration, qui porte cette fois ci sur l'exponentiation finale, plutôt que sur la boucle de Miller, est l'usage du Frobenius dans le corps fini $\mathbb{F}_{q^k}$ pour décomposer $(q^k - 1)/\ell$. Mais nous détaillons cela pour les courbes que nous exposons au chapitre suivant.

II.3.5 Trois types de couplage

Nous avons décrit plus tôt un protocole d'échange de clé tripartie, en choisissant un couplage dit « symétrique ». Dans la Sous-Section II.1.3, nous avons vu des applications, à savoir la trace Tr et l'anti-trace aTr, envoyant des sous-groupes d'ordre ℓ de la torsion $E[\ell]$ vers des sous-groupes, toujours de $E[\ell]$, correspondant aux sous-espaces propres du Frobenius. Pour clarifier cela et aider les concepteurs de protocoles cryptographiques, Galbraith *et al.* [GPS08] ont défini trois types de couplage et indiqué leurs propriétés.

Pour un couplage $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ entre trois groupes d'ordre premier ℓ, son type est défini comme suit :

Type 1 **:** Un couplage de type 1 est un couplage symétrique, à savoir $\mathbb{G}_1 = \mathbb{G}_2$. Sur les courbes elliptiques, cela ne correspond qu'au cas des courbes supersingulières. En effet, sur une telle courbe E, il existe un morphisme $\psi \in \mathbb{F}_{q^k}(E)$, dit application de distorsion, qui envoie les points de $E(\mathbb{F}_q)$ vers $E(\mathbb{F}_{q^k})$ [BSS05, Chapitre IX]. Ainsi, le couplage est pris comme $e(P, \psi(Q))$ où $P, Q \in E(\mathbb{F}_q)[\ell]$ et $\psi(Q) \in E(\mathbb{F}_{q^k})[\ell] \backslash E(\mathbb{F}_q)$. Verheul note que les courbes ordinaires n'ont pas de telle application de distorsion [Ver01].

Type 2 **:** Un couplage est de type 2 lorsqu'il existe un morphisme $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ facilement calculable et non-trivial. Pour une courbe elliptique $E/\mathbb{F}_q$, on sait que la trace Tr envoie les points de $E[\ell]$ sur $E[\ell] \cap \mathrm{Ker}(\pi - [1])$. Si $\mathbb{G}_1 = E[\ell] \cap \mathrm{Ker}(\pi - [1])$ et que l'on choisit $\mathbb{G}_2 \subset E[\ell]$ d'ordre ℓ différent de $\mathbb{G}_1$ et du sous-groupe de trace nulle, alors Tr est un morphisme de groupes facilement calculable et non-trivial

entre $\mathbb{G}_2$ et $\mathbb{G}_1$.

Type 3 : Un couplage est de type 3 lorsqu'il n'existe pas de morphisme de groupes $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ facilement calculable et non-trivial. Pour une courbe elliptique $E/\mathbb{F}_q$, on sait que l'anti-trace aTr envoie les points de $E[\ell]$ sur $E[\ell] \cap \mathrm{Ker}(\pi - [q])$. Si on prend $\mathbb{G}_1 = E[\ell] \cap \mathrm{Ker}(\pi - [1])$ et que l'on choisit $\mathbb{G}_2 = E[\ell] \cap \mathrm{Ker}(\pi - [q])$, alors on sait qu'il existe un isomorphisme $\psi : \mathbb{G}_2 \to \mathbb{G}_1$, car ces groupes sont tout deux cycliques d'ordre ℓ, mais on n'a aucun moyen de le calculer de façon efficace (cette fois ci, $\mathrm{Tr}(\mathbb{G}_2) = \{P_\infty\}$ par définition).

Suite à la définition des types de couplage par Galbraith *et al.* [GPS08], nombre de protocoles ont été décrits avec des couplages de type 1 ou 2, car le morphisme $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ facilitait généralement les preuves de sécurité. Cependant, ce morphisme ne semble pas avoir un rôle fondamental dans les preuves de sécurité. Il n'y a donc pas de raison pour que les couplages de type 1 ou 2 ne soient pas remplacés par des couplages de type 3 [CM15 ; AHO16], modulo quelques modifications dans la description des protocoles, d'autant plus que ces derniers offrent de meilleures performances [Gui13].

Dans la suite, on peut donc considérer qu'un couplage de type 3 sur une courbe elliptique $E/\mathbb{F}_q$ est choisi tel que $\mathbb{G}_1 = E[\ell] \cap \mathrm{Ker}(\pi - [1])$ et $\mathbb{G}_2 = E[\ell] \cap \mathrm{Ker}(\pi - [q])$.

II.4 Multiplication complexe et anneau d'endomorphismes

L'objet de cette section est l'étude de l'anneau d'endomorphismes de E/K. On a déjà vu que la multiplication-par-m est un endomorphisme de E. Donc $\mathbb{Z} \subseteq \mathrm{End}(E)$, par l'identification $m \mapsto [m]$ (Théorème II.1.4).

Lorsque cet anneau $\mathrm{End}(E)$ est strictement plus grand que $\mathbb{Z}$ alors on dit que E est à *multiplication complexe*. Quel que soit le corps K, il y a trois possibilités pour $\mathrm{End}(E)$ ([Sil09, Corollaire III.9.4]), cet anneau est isomorphe

— à l'anneau des entiers relatifs $\mathbb{Z}$;

— à un ordre $\mathcal{O}$ dans un corps de nombres quadratiques imaginaires ;

— à un ordre maximal dans une algèbre de quaternion définie.

Si K est de caractéristique nulle, la dernière alternative n'est jamais possible. Et si K est de caractéristique non nulle, c'est la première alternative qui ne survient jamais.

Vu que l'on utilise des corps finis en cryptographie, on ne vas pas considérer le cas

$\mathrm{End}(E) \simeq \mathbb{Z}$. De plus, $\mathrm{End}(E)$ est isomorphe à un ordre d'algèbre de quaternion si, et seulement si, $E/\mathbb{F}_q$ est une courbe elliptique supersingulière ([Sil09, Théorème V.3.1]).

Ainsi, pour les courbes ordinaires $E/\mathbb{F}_q$, l'anneau d'endomorphismes $\mathrm{End}(E)$ est isomorphe à un ordre dans un corps de nombres quadratiques imaginaires.

Détaillons. Soit $E/\mathbb{F}_q$ une courbe elliptique ordinaire. L'anneau d'endomorphismes $\mathrm{End}(E)$ est isomorphe à un ordre $\mathcal{O}$ d'un corps de nombres quadratiques imaginaires $K := \mathbb{Q}(\sqrt{-d})$, pour un entier $d > 0$ sans facteur carré et où $\sqrt{-d} \in \mathbb{C}$. Notons

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

Il existe alors $f \in \mathbb{Z}$ tel que $\mathrm{End}(E) \simeq \mathcal{O} = \mathbb{Z} + f\delta\mathbb{Z}$ et tel que le discriminant du polynôme minimal du Frobenius $X^2 - tX + q$ est égal au discriminant de l'ordre $\mathbb{Z} + f\delta\mathbb{Z}$, à savoir $-f^2 d = t^2 - 4q$ si $d \equiv 3 \pmod 4$ ou $-4f^2 d = t^2 - 4q$ sinon.

Deux principales constructions [CP01 ; DEM05] de courbes elliptiques à couplage fixent un degré de plongement k et un $d > 0$ sans facteur carré, et produisent des entiers t, ℓ et q tels qu'il y ait une courbe elliptique $E/\mathbb{F}_q$ dont la trace du Frobenius est t, le nombre de points $\#E(\mathbb{F}_q)$ est divisible par ℓ, la courbe est de degré de plongement k et $\mathrm{End}(E)$ est isomorphe a un ordre de $\mathbb{Q}(\sqrt{-d})$. La méthode CM [9] d'Atkin et Morain [AM93] permet alors de trouver l'équation de la courbe E pourvu que D, le *discriminant CM* de E, ne soit pas trop grand, où $D > 0$ est la partie sans facteur carré de

$$-f^2 D = t^2 - 4q. \tag{II.4.14}$$

L'Équation (II.4.14) est appelée *équation CM* de E et « D pas trop grand » signifie, disons, environ 10^{16} [Sut12].

II.4.1 Endomorphisme GLV lorsque $j(E) = 0$

Gallant *et al.* [GLV01] remarquent que les multiplications sur la courbe $E/\mathbb{F}_q$ peuvent être accélérées s'il existe des endomorphismes facilement calculables dans $\mathrm{End}(E)$.

Illustrons avec un exemple qui nous servira par la suite. Soit $E/\mathbb{F}_q$ une courbe ordinaire de j-invariant nul. Alors E est d'équation $y^2 = x^3 + b$ pour $b \in \mathbb{F}_q^\times$ et $q \equiv 1 \pmod 3$.

En effet, si $j(E) = 0$ et $q \equiv 2 \pmod 3$ alors E est supersingulière. Pour voir cela,

9. CM pour *Complex Multiplication* en anglais.

montrons que $\#E(\mathbb{F}_q) = q + 1$. Comme 3 est inversible modulo $q - 1$, l'application de $\mathbb{F}_q$ dans lui-même $x \mapsto x^3 + b$ est une bijection. Notons $x_0 \in \mathbb{F}_q$ l'unique élément tel que $x_0^3 + b = 0$. La moitié des éléments $x \in \mathbb{F}_q \backslash \{x_0\}$ donne un $x^3 + b$ carré non nul dans $\mathbb{F}_q$, qui ont donc deux racines carrées, appelons-les y et $-y$. Il y a donc $q - 1$ couples $(x, \pm y)$ sur la courbe tels que $x \neq x_0$. Le couple $(x_0, 0)$ est aussi un point de la courbe. Il reste le point à l'infini, ce qui donne bien $\#E(\mathbb{F}_q) = q + 1$.

Comme $q \equiv 1 \pmod 3$, il existe une racine primitive troisième de l'unité, notons-la $\zeta \in \mathbb{F}_q$. Elle est racine du polynôme $X^2 + X + 1$. On remarque que l'application ψ définie par $(x, y) \mapsto (\zeta x, y)$ est un endomorphisme de E, car $\zeta^3 = 1$. Le polynôme minimal de ψ est aussi $X^2 + X + 1$, c'est-à-dire

$$\psi \circ \psi + \psi + [1] = [0], \quad \text{dans } \mathrm{End}(E). \tag{II.4.15}$$

Soit $P \in E(\mathbb{F}_q)$ d'ordre ℓ premier. L'image $\psi(P)$ est un point de $E(\mathbb{F}_q)$ d'ordre ℓ. Cela veut dire qu'il existe $\lambda \pmod \ell$ tel que $\psi(P) = [\lambda]P$ pour tout P de $E(\mathbb{F}_q)[\ell]$. En regardant l'Équation (II.4.15) sur $E(\mathbb{F}_q)[\ell]$, on en déduit

$$[\lambda^2] + [\lambda] + [1] = [0].$$

Donc λ est aussi une racine primitive troisième de l'unité du corps $\mathbb{Z}/\ell\mathbb{Z}$. Supposons maintenant que l'on souhaite calculer le multiple $[m]P$ pour $P \in E(\mathbb{F}_q)[\ell]$. Pour cela, on trouve le couple reste-quotient (m_1, m_2) de la division euclidienne de m par λ, *i.e.* $m = m_1 + m_2\lambda$ avec $0 \leqslant m_1 < \lambda$, et on peut alors calculer $[m]P = [m_1]P + [m_2]\psi(P)$ à l'aide d'un algorithme de multi-exponentiation (voir par exemple [GLV01, Algorithme 1]). Remarquons que m_1 et m_2 sont de taille $\sqrt{\ell}$, ainsi la longueur de la boucle de la multi-exponentiation est en moyenne $(\log_2 \ell)/2$.

Pour les courbes de j-invariant 0, le discriminant CM est $D = 3$, par conséquence l'Équation (II.4.14) donne $-3f^2 = t^2 - 4q$. Comme $\zeta \in \mathbb{F}_q$ est une racine primitive troisième de l'unité, ζ vérifie $\zeta^2 + \zeta + 1 = 0$. On peut donc prendre $\zeta = (\sqrt{-3} - 1)/2$, où $\sqrt{-3} \equiv t/f \pmod q$. Ainsi

$$\zeta \equiv \frac{t - f}{2f} \pmod q.$$

Similairement, comme $\lambda \in \mathbb{Z}/\ell\mathbb{Z}$ satisfait l'équation $\lambda^2 + \lambda + 1 = 0$, il est aussi possible de prendre $\lambda = (\sqrt{-3} - 1)/2$. Cependant, ici on est dans $\mathbb{Z}/\ell\mathbb{Z}$ et plus dans $\mathbb{F}_q$. En se rappelant que ℓ divise $q - t + 1$, le nombre de points $\mathbb{F}_q$-rationnels, on obtient la congruence

$q \equiv t-1 \pmod{\ell}$ que l'on peut utiliser dans d'équation CM (II.4.14) :

$$\begin{aligned} -3f^2 \equiv t^2 - 4q \pmod{\ell} &\Leftrightarrow -3f^2 \equiv t^2 - 4(t-1) \pmod{\ell}, \\ &\Leftrightarrow -3f^2 \equiv t^2 - 4t + 4 \pmod{\ell}, \\ &\Leftrightarrow -3 \equiv \frac{(t-2)^2}{f^2} \pmod{\ell}, \end{aligned}$$

et ainsi $\sqrt{-3} \equiv (t-2)/f \pmod{\ell}$. Donc

$$\lambda \equiv \frac{t-f-2}{2f} \pmod{\ell}.$$

Notons que dans la pratique, des ajustements peuvent être nécessaires pour avoir le bon endomorphisme GLV, vu que $(\omega x, y) = [\lambda](x, y)$ ou $(\omega^2 x, y) = [\lambda](x, y)$.

II.5 Construction Brezing-Weng

La construction de Brezing et Weng permet de construire des familles de courbes elliptiques avec un degré de prolongement prescrit.

Supposons que $E/\mathbb{F}_q$ soit une courbe elliptique ordinaire de degré de plongement k par rapport à ℓ premier. Cela signifie

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

Alors, la première équation $q \equiv t-1 \pmod{\ell}$ implique, avec la seconde équation, que $t-1$ est une racine primitive k-ième de l'unité modulo ℓ, car k est l'ordre de $q \pmod{\ell}$. En notant $\Phi_k(X)$ le k-ième polynôme cyclotomique, la combinaison des deux équations donne donc $\Phi_k(t-1) \equiv 1 \pmod{\ell}$.

Si on regarde l'anneau d'endomorphismes $\mathrm{End}(E)$, il est isomorphe à un ordre $\mathcal{O}$ dans un corps de nombres quadratiques imaginaires $K = \mathbb{Q}(\sqrt{-d})$, pour $d > 0$ sans facteur carré. Le Frobenius π sur $E/\mathbb{F}_q$, de polynôme minimal $X^2 - tX + q$, correspond donc à un élément $\omega \in \mathcal{O}$ de trace t et de norme $\omega\overline{\omega} = q$. En l'occurrence, il existe $b \in \mathbb{Z}$ tel que $\omega = (t + b\sqrt{-d})/2$.

Remarquons alors que $\omega\overline{\omega} - t + 1 \equiv 0 \pmod{\ell}$, ainsi on peut obtenir une congruence

pour l'entier b :

$$\begin{aligned}\omega\overline{\omega} - t + 1 &= (t^2 + b^2 d)/4 - t + 1,\\ &= (t^2 - 4t + 4 + b^2 d)/4,\\ &= ((t-2)^2 + b^2 d)/4,\end{aligned}$$

à savoir $b^2 d \equiv -(t-2)^2 \pmod{\ell}$. Si on note δ une racine carrée de $-d \pmod{\ell}$, alors

$$b \equiv \pm\frac{t-2}{\delta} \pmod{\ell}. \tag{II.5.16}$$

Cela donne lieu à l'algorithme simple suivant, crédité à Cocks et Pinch [CP01]. Étant donné k entier et $d > 0$ entier sans facteur carré, prendre ℓ un entier premier tel que $-d$ soit un carré modulo ℓ (on dit que ℓ se décompose dans $\mathbb{Q}(\sqrt{-d})$) et $\ell \equiv 1 \pmod{k}$, ce qui assure l'existence dans $\mathbb{Q}(\sqrt{-d})$ d'une racine k-ième primitive de l'unité modulo ℓ. Notons ζ_k une telle racine. Soient alors $a \equiv \zeta_k + 1 \pmod{\ell}$ et $b \equiv \pm(a-2)\delta^{-1} \pmod{\ell}$, où $\delta^2 \equiv -d \pmod{\ell}$. Il reste ensuite à vérifier si l'élément $\omega = (a + b\sqrt{-d})/2 \in \mathbb{Q}(\sqrt{-d})$ est de norme un entier premier q (ou puissance de nombre premier). Enfin, on trouve la courbe elliptique $E/\mathbb{F}_q$ correspondante grâce à la méthode CM d'Atkin et Morain [AM93].

Avec cette méthode, a et b sont solutions d'équations modulo ℓ, donc a et b sont de taille $O(\ell)$. L'entier q, qui est la norme de $\omega = (a + b\sqrt{-d})/2$, est alors de taille $O(\ell^2)$.

L'approche générale de Brezing et Weng consiste à construire q, ℓ et t comme des polynômes de $\mathbb{Q}[X]$ grâce aux observations précédentes. Pour cela, étant donnés k entier et $d > 0$ entier sans facteur carré, on construit le corps de nombres $M = \mathbb{Q}(\zeta_n, \sqrt{-d})$, pour n multiple de k et $\zeta_n \in \mathbb{C}$ une racine primitive n-ième de l'unité. Supposons que $M \simeq \mathbb{Q}[X]/(f(X))$, où $f(X) \in \mathbb{Q}[X]$ est un polynôme irréductible de degré $d_f = \varphi(n)$ si $\sqrt{-d} \in \mathbb{Q}(\zeta_n)$, de degré $2\varphi(n)$ sinon.

Pour i entier entre 1 et $\varphi(k)$, notons $g_i(X)$ les polynômes représentant les racines primitives k-ièmes de l'unité, c'est-à-dire que pour tout i la classe $g_i(X) \pmod{f(X)}$ dans M est une racine primitive k-ième de l'unité. Et notons $h_1(X)$ et $h_2(X) = -h_1(X)$ les polynômes représentant les racines carrées de $-d$. Supposons qu'il existe $1 \leqslant i \leqslant \varphi(k)$ et $j \in \{1, 2\}$ tels que $g_i(X)$ et $h_j(X)$ soient tous deux à coefficients entiers.

Comme précédemment, on définit a et b représentant l'élément $\omega = (a + b\sqrt{-d})/2$, mais on souhaite que ces polynômes $a(X)$ et $b(X)$ gardent des coefficients entiers. Or on ne sait pas si l'inverse de $h_j(X) \pmod{f(X)}$ (une racine carrée de $-d \pmod{\ell}$ dans

l'Équation (II.5.16)) est à coefficients entiers. Pour remédier à cela, on remarque dans le calcul de la norme de ω que $db^2 = d((a-2)/\delta)^2 = -(a-2)^2$. Ainsi, on peut prendre $\hat{b} = (a-2)\delta$ et remplacer le db^2 par $d^{-1}\hat{b}^2$ dans le calcul de la norme. Posons alors

$$\begin{aligned} a(X) &\equiv g_i(X) + 1 \pmod{f(X)}, \\ \hat{b}(X) &\equiv (a(X)-2)h_j(X) \pmod{f(X)}. \end{aligned}$$

On teste l'existence d'une classe d'équivalence $x_0 \pmod d$ telle que $\hat{b}(x_0) \equiv 0 \pmod d$. Alors pour tout $x_1 \equiv x_0 \pmod d$, l'élément $\hat{b}(x_1)/d$ est un entier.

Notons $q(X)$ la norme de notre « entier quadratique », *i.e.*

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

Supposons que $q(X) \in \mathbb{Q}[X]$ soit irréductible et prenne des valeurs entières $q(x_1)$ lorsque $x_1 \equiv x_0 \pmod d$. Les polynômes $a(X)$ et $\hat{b}(X)$ étant définis modulo $f(X)$, on souhaite que le premier ℓ soit une image du polynôme $f(X)$. Pour cela, il faut que $f(dX + x_0) \in \mathbb{Z}[X]$ soit irréductible, car prendre $x_1 \equiv x_0 \pmod d$ équivaut à l'existence d'un entier $m \in \mathbb{Z}$ tel que $x_1 = dm + x_0$.

On peut ainsi chercher $x_1 \equiv x_0 \pmod d$ tel que $f(x_1)$ et $q(x_1)$ soient premiers. Si tel est le cas, alors il existe une courbe elliptique sur $\mathbb{F}_{q(x_1)}$ dont l'anneau d'endomorphismes est un ordre $\mathcal{O}$ de $\mathbb{Q}(\sqrt{-d})$. Le Frobenius correspond alors à l'élément

$$\omega_{x_1} = \frac{1}{2}\left(a(x_1) \pm \frac{\hat{b}(x_1)}{d}\sqrt{-d}\right) \in \mathcal{O}.$$

Le nombre de points sur la courbe est alors divisible par $f(x_1)$ par construction :

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

Les degrés de $a(X)$ et $\hat{b}(X)$ sont inférieurs ou égaux à $\deg(f(X)) - 1$. Ainsi, pour un k fixé, la valeur $\rho = \log(q)/\log(\ell)$ tend vers $2 - 2/\deg(f(X))$ pour $\ell \to \infty$, ce qui est strictement inférieur à 2.

La taxonomie de Freeman, Scott et Teske [FST10] donne plusieurs exemples de constructions Brezing-Weng, notamment leur Construction 6.6 qui décrit des familles de courbes

elliptiques à couplage en fonction du degré de plongement k modulo 6, tant que k n'est pas multiple de 18.

II.6 Sécurité d'une courbe elliptique à couplage

La plupart des schémas cryptographiques utilisant des couplages bilinéaires se servent de problèmes en général plus simples que le problème du logarithme discret (DLP [10]). Malheureusement, la difficulté concrète de ces problèmes n'est pas connue. L'approche retenue pour générer un couplage est donc de sélectionner des paramètres tels que le DLP dans les trois groupes $\mathbb{G}_1$, $\mathbb{G}_2$ et $\mathbb{G}_T$ soit supposé dur à résoudre. Apporter une solution au DLP a été intensément étudié durant les quarante dernières années et de multiples algorithmes ont été proposés.

Le DLP sur les courbes elliptiques est appelé ECDLP (où EC signifie *Elliptic Curve*) et il est considéré comme le DLP le plus difficile à résoudre dans le sens où nous ne connaissons que des algorithmes dits génériques [Sho97] pour le résoudre, tels que Pas de Bébé, Pas de Géant de Shanks [Sha71] ou rho (ρ) de Pollard [Pol78]. D'où, pour des raisons d'efficacité et de sécurité, les groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ sont choisis d'ordre premier, *i.e.* le nombre ℓ est premier. Et vu que la meilleure variante de Pollard-rho calcule le logarithme discret en au plus $\sqrt{\pi\ell/4} \approx 0,886\sqrt{\ell}$ étapes [BLS11], il suffit de prendre ℓ avec 2λ bits pour une sécurité de λ bits.

Une valeur, notée ρ, permet de décrire l'efficacité de la représentation des éléments de $\mathbb{G}_1$. Elle est calculée comme $\rho = \log(q)/\log(\ell)$ ou comme $\rho = \deg(q)/\deg(\ell)$ lorsque q et ℓ sont des polynômes de $\mathbb{Q}[X]$. Quand $\rho = 1$, la courbe est d'ordre premier ℓ : c'est le meilleur cas possible pour l'arithmétique des points de $E(\mathbb{F}_q)$, et donc de $\mathbb{G}_1$.

Alors que déterminer la taille de $\mathbb{G}_1$ et $\mathbb{G}_2$ dans la courbe elliptique est assez direct, en faire de même pour $\mathbb{G}_T$ dans le corps fini $\mathbb{F}_{q^k}$ est bien plus complexe ! En effet, les logarithmes discrets de $\mathbb{F}_{q^k}$ peuvent être calculés en temps sous-exponentiel (et parfois même en temps quasi-polynomial) par des algorithmes de crible par corps de nombres (NFS [11]). Dans le cas général, il est difficile d'évaluer la complexité de l'algorithme NFS et de ses variants (voir [GM17] pour plus de détails). Pour se donner une idée de la complexité temporelle des algorithmes NFS, on a recours à la *notation L*, définie en (I.4.2)

10. *Discrete Logarithm Problem* en anglais.
11. *Number Field Sieve* en anglais.

et que l'on rappelle :

$$L_{q^k}[\alpha, c] = \exp\left((c + o(1))\,(\ln q^k)^{\alpha}(\ln\ln q^k)^{1-\alpha}\right)$$

où $\alpha \in [0, 1]$ et $c > 0$. Dans notre cas de figure, à savoir les corps finis contenant l'image d'un couplage, les algorithmes type NFS ont tous pour complexité $L_{q^k}[1/3, c]$ avec c allant de $\sqrt[3]{32/9}$ à $\sqrt[3]{96/9}$. La valeur de c dépend de la variante du NFS choisie et cela conditionne l'estimation du niveau de sécurité. En effet, entre 20 et 30 bits de sécurité peuvent être perdus [BD19]. L'exemple édifiant dans [BD19] est la réévaluation à 100 bits de sécurité les courbes Barreto-Naehrig [BN06], hégémoniques jusqu'alors, en partie due à leur simplicité pour 128 bits de sécurité et à leur valeur ρ de 1.

Deux critères déterminent quelle variante du NFS utiliser : si q est spécial, *i.e.* q est l'image d'un polynôme de degré au moins 3, et si k est composé.

	q non-spécial	q spécial
k premier	NFS [SS16] $(9c^3 \in [64, 96])$	SNFS [JP14 ; Bar+15] $(9c^3 \in [32, 64])$
k composé	exTNFS [KB16] $(9c^3 \in [48, 64])$	SexTNFS [KB16] $(9c^3 = 32)$

TABLE II.1 – Comment choisir la variante NFS pour les courbes à couplage

Jusqu'aux travaux récents de Kim et Barbulescu [KB16], seules les courbes avec q spécial et k composé étaient considérées et déployées car elles offrent les meilleures performances pour le calcul du couplage. Barbulescu et Duquesne ont appliqué les améliorations apportées aux algorithmes NFS dans le cas particulier des corps finis utilisés dans les couplages et ont proposé une mise à jour des estimations des tailles de clé [BD19]. Pour une sécurité de 128 bits, ils incitent à sélectionner une taille de corps d'au moins $k \log_2(q) = 2930$ (respectivement 3618, 5004) si NFS (respectivement exTNFS, SexTNFS) est le meilleur algorithme pour calculer les logarithmes discrets dans $\mathbb{F}_{q^k}$ (l'usage habituel était de choisir $k \log_2(q) = 3072$ [Sma18]). Pour évaluer la sécurité de nos courbes, nous utiliseront l'implémentation faite par Guillevic de l'article [GS21], disponible à l'adresse :

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

Les seules méthodes pour construire des courbes avec une valeur ρ proche de 1, comme la construction Brezing-Weng [BW05], donnent l'ordre du corps fini comme image d'un

polynôme (donc q est spécial) ; les autres constructions ont une valeur ρ proche de 2 comme la construction Cocks-Pinch [CP01] ou la construction Dupont-Enge-Morain [DEM05]. Pour 128 bits de sécurité, nous considérons l'indication donnée par Guillevic [Gui20] et cherchons $k \log_2(q) \in [3072, 5376]$.

Si $\log_2(\ell) = 256$ pour assurer une sécurité de 128 bits du côté de la courbe, nous savons que $\log_2(q) = 256\rho$ et que le corps fini $\mathbb{F}_{q^k}$ est de taille $256\rho k$. L'encadrement donné par Guillevic et $1 \leqslant \rho \leqslant 2$ donnent un intervalle de recherche pour de potentiel degré de plongement k :

$$\left.\begin{aligned} 3072 \leqslant 256\rho k \leqslant 5376 \\ 1 \leqslant \rho \leqslant 2 \end{aligned}\right\} \Rightarrow 6 \leqslant k \leqslant 21.$$

Concrètement, cela signifie que $k < 6$ ne permettrait pas d'avoir une sécurité de 128 bits (pour l'intervalle dans lequel ρ est considéré) et $k > 21$ assurerait amplement la sécurité de 128 bits sur le corps fini, au risque même d'impacter fortement la performance des opérations.

Nous pouvons maintenant expliquer pourquoi nous ne choisissons pas de courbes sur corps binaires ou ternaires, ni de courbes supersingulières.

Il n'est pas possible de choisir le corps $\mathbb{F}_2$ ou $\mathbb{F}_3$ comme corps de base pour une courbe si on veut remplir les critères de sécurité. Il faut donc en prendre des extensions $\mathbb{F}_{p^n}$, pour $p = 2$ ou $p = 3$. Ainsi $\mathbb{G}_1 \subset E(\mathbb{F}_{p^n})$, mais il existe alors des algorithmes de calcul d'indice (de la même idée que NFS) pour résoudre le ECDLP (voir [Bar+14 ; GKZ14]). Ces attaques indiquent même que le meilleur des cas (équilibre entre performance et sécurité) survient lorsque la courbe est définie sur un corps premier.

Après exposition succincte des courbes supersingulières, il est clair qu'elles ne sont pas appropriées pour les couplages. Leur degré de plongement est 2 sur des corps premiers de caractéristique au moins 5 et elles ne supportent que le couplage de type 1. Ces considérations font qu'il est plus judicieux de trouver une courbe ordinaire pour y faire des couplages.

Conclusion du second chapitre

Nous avons abordé dans ce chapitre la définition des courbes elliptiques, en nous intéressant principalement aux couplages et aux courbes compatibles avec cet opération. Nous avons vu comment construire de telles courbes avec la méthode de Brezing-Weng, en paramétrisant une famille avec un degré de plongement k et un discriminant CM D.

Du point de vue de la sécurité, l'équilibre à obtenir est subtil : les manières d'évaluer la sécurité sur la courbe elliptique et sur le corps fini ne sont pas les mêmes et il faut que le choix des paramètres permettent les calculs effectifs. Ainsi, nous avons toujours recours à des constructions qui permettent d'instancier des courbes elliptiques ordinaires sur corps premier avec un degré de plongement préalablement fixé.

Chapitre III

Couplage avec un faible coût d'exponentiation dans $\mathbb{G}_1$

Nous avons vu au chapitre précédent que l'évaluation de la sécurité des courbes à couplages n'est pas facile, notamment à cause des algorithmes de calcul de logarithmes discrets dans le corps fini $\mathbb{F}_{q^k}$. En effet, contrairement à ce qui était pensé jusqu'au début des années 2010, la sécurité dans le corps fini ne dépend pas seulement de la longueur $k \log_2(q)$, mais surtout de la forme des entiers q et k [KB16 ; KJ17]. Il se peut donc qu'un corps à 3000 bits ait la même sécurité qu'un autre à 5000 bits étant donnés k et q ayant des propriétés distinctes.

Les trois groupes $\mathbb{G}_1$, $\mathbb{G}_2$ et $\mathbb{G}_T$ sont affectés par ces considérations, vu que q a aussi un impact direct sur l'arithmétique de $\mathbb{G}_1$ et $\mathbb{G}_2$. Concrètement, il est parfois nécessaire d'augmenter le paramètre q (qui devient alors bien plus grand que le minimum recommandé 2^{256}) pour rester compatible avec des valeurs de k permettant des calculs de couplage performants. Cela est illustré par la courbe BLS [BLS03] de degré de plongement 12 promue par Barbulescu et Duquesne [BD19] pour un niveau de 128 bits de sécurité. Ces courbes sont définies sur un corps premier fini de 460 bits, ce qui est 80% plus grand que les courbes BN [BN06] précédemment utilisées.

Cet exemple est représentatif de la stratégie actuelle pour choisir les paramètres d'une courbe à couplage [BD19]. Le but est en effet de trouver un bon compromis entre les complexités des différents groupes/opérations. Plus précisément, c'est trouver un jeu de paramètres qui ne pénalise aucune opération particulière. Cette stratégie fait donc implicitement l'hypothèse que les protocoles présentent une certaine symétrie dans leur utilisation des trois groupes $\mathbb{G}_1$, $\mathbb{G}_2$ et $\mathbb{G}_T$, mais aussi entre les diverses entités faisant des opérations dans ces groupes. Cela peut être vrai pour certains scénarios mais il y a d'autres cas où ces suppositions sont fausses.

Prenons comme exemple le cas du schéma Enhanced Privacy ID (EPID) introduit par Brickell et Li [BL07b] et maintenant massivement déployé pour sécuriser les enclaves Intel

SGX [AlL16]. L'EPID est une variation de *Direct Anonymous Attestation* (DAA) [BCC04] avec des capacités de révocation améliorées, ce qui veut dire que pour révoquer une clé sk il suffit de simplement ajouter une signature faite avec sk à une liste de révocation. Les signatures suivantes devront alors contenir une preuve qu'elles sont générées grâce à une clé secrète qui n'a rien signée dans la liste de révocation. Cette fonctionnalité implique d'effectuer un nombre élevé d'exponentiations dans $\mathbb{G}_1$ [BL10], linéaire en le nombre n de signatures révoquées. Par exemple, la complexité de la signature est $3\mathbf{e}_1 + 4\mathbf{e}_T + 6n\mathbf{e}_1$, où $\mathbf{e}_i$ représente une exponentiation dans $\mathbb{G}_i$, alors que la vérification d'une signature a pour complexité $4\mathbf{e}_1 + 2\mathbf{e}_2 + 4\mathbf{e}_T + 1\mathbf{P} + 6n\mathbf{e}_1$, où $\mathbf{P}$ représente le calcul d'un couplage. Dans ces cas-là, on remarque que vouloir un q aussi proche du minimum possible 2^{256} (ce qui réduit donc la complexité des opérations dans $\mathbb{G}_1$), et même si cela détériore la performance du calcul du couplage, est un objectif sensé vu qu'il n'y a qu'un couplage à calculer contre environ $6n$ exponentiations dans $\mathbb{G}_1$ pour générer ou vérifier une signature. Cela est d'autant plus vrai que le couplage est calculé par le vérifieur, une entité qui est supposée, dans le contexte des DAA, avoir bien plus de puissance que le signataire (généralement un périphérique aux ressources limitées). En d'autres mots, nous avons besoin ici d'une courbe qui optimise les opérations dans $\mathbb{G}_1$, même si c'est au prix d'un couplage plus coûteux.

Nous nous proposons donc d'examiner les courbes qui offriraient une meilleure performance des opérations dans le groupe $\mathbb{G}_1$, quitte à ce que ce soit au détriment des autres opérations. Ce chapitre correspond à la contribution faite dans l'article *Curves with Fast Computations in the First Pairing Group* [CDS20], co-écrit avec Sylvain Duquesne et Olivier Sanders et accepté à la conférence internationale « Cryptology and Network Security » en décembre 2020.

III.1 Idées

À la fin de la Section II.6, nous n'avons que dégagé des encadrements sur les différents paramètres de courbe. Malheureusement, comme déjà précisé plus haut, il n'y a pas de choix facile à faire car la sécurité et la performance des couplages résultants peuvent drastiquement varier d'un jeu de paramètres à l'autre. En particulier, il n'y a pas de linéarité dans l'évaluation de la sécurité ; par exemple, la sécurité du DLP dans le corps $\mathbb{F}_{q^k}$ est bien plus grande dans le cas « premier » avec $k \in \{11, 13\}$ que dans le cas $k = 12$. Cela signifie que dans le premier cas, il est possible de choisir un q plus petit, améliorant

ainsi la performance des calculs sur $\mathbb{G}_1$. Hélas, le même problème survient lorsque nous considérons les performances du couplage et des exponentiations dans $\mathbb{F}_{q^k}$, mais avec des conclusions contradictoires, vu qu'un k composé (et pair plus particulièrement) permet l'usage d'astuces et de calculs plus efficaces pour le couplage et les opérations dans $\mathbb{G}_T$. Il est donc nécessaire de faire un choix parmi ces paramètres, qui est d'autant plus important lorsque, dans les protocoles cryptographiques, la complexité entre les diverses entités est déséquilibrée.

Le premier groupe d'un couplage $\mathbb{G}_1$ est, comme on l'a vu à la fin de la Section II.3, communément défini comme $E(\mathbb{F}_q)[\ell]$. Nous avons donc un intérêt à réduire la taille de q car cela induit des opérations plus rapides dans $\mathbb{G}_1$. Pour assurer les 128 bits de sécurité sur la courbe, il faut $\log_2(\ell) \geqslant 256$. Si le groupe sur la courbe est d'ordre premier ℓ, alors par la borne de Hasse nous savons que $|q+1-\ell| \leqslant 2\sqrt{q}$. Nous pouvons donc supposer que q a le même nombre de bits que ℓ et que q est un entier de cinq mots machines sur une architecture 64 bits. Cela signifie que q a entre 256 et 320 bits. En effet, le cas q de 256 (exactement) correspond aux courbes d'ordre premier Barreto-Naehrig [BN06] de degré de plongement 12, qui n'offrent pas 128 bits de sécurité, comme l'ont prouvé Barbulescu et Duquesne [BD19]. Ainsi, trouver un q de quatre mots machines sur une architecture 64 bits n'est pas possible. Et un de six mots machines correspond déjà aux nouveaux paramètres proposés dans [BD19].

Nous voudrions aussi pouvoir utiliser la méthode GLV [GLV01] sur notre courbe, il faut donc choisir une courbe avec un petit discriminant CM. Les endomorphismes GLV les plus simples à évaluer sont ceux qui correspondent à une simple multiplication de chaque coordonnée d'un point. Il existe deux cas souvent utilisés :

Invariant $j = 1728$: le discriminant CM est 1, alors la courbe admet pour équation $y^2 = x^3 + ax$ et l'endomorphisme GLV est $(x, y) \mapsto (-x, iy)$, pour $i \in \mathbb{F}_q$ une racine primitive quatrième de l'unité.

Invariant $j = 0$: le discriminant CM est 3, alors la courbe admet pour équation $y^2 = x^3 + b$ et l'endomorphisme GLV est $(x, y) \mapsto (\zeta x, y)$, pour $\zeta \in \mathbb{F}_q$ une racine primitive troisième de l'unité (comme exposé en Sous-Section II.4.1).

Si q est d'au plus 320 bits et ℓ d'au moins 256, alors $1 < \rho \leqslant 1.25$, où la valeur ρ est $\log(q)/\log(\ell)$. En recherchant dans la taxonomie de Freeman, Scott et Teske [FST10], la courbe que nous recherchons à un degré de plongement $k \in \{8, 11, 12, 16, 17, 19\}$.

Le degré de plongement 8 ne donne pas un corps fini $\mathbb{F}_{q^k}$ sûr, car il a au plus $8 \times 1.5 \times 256 = 2560$ bits de sécurité ; il n'est donc pas retenu. Le degré de plongement 16 n'est

pas gardé non plus, car il correspond à la famille des courbes KSS [KSS08] et Barbulescu et Duquesne ont conseillé [BD19] de prendre un premier q d'au moins 330 bits, *i.e.* un nombre de 6 mots machines.

Il nous reste donc plus que $k \in \{11, 13, 17, 19\}$ qui correspond à la Construction 6.6 de la taxonomie [FST10], où le discriminant CM est 3, donc les courbes sont d'équation $y^2 = x^3 + b$, et la trace du Frobenius $t(X)$, le premier $q(X)$ et l'ordre $\ell(X)$ sont des polynômes de $\mathbb{Q}[X]$ tels que $\ell(X)$ est un polynôme cyclotomique divisant $q(X) - t(X) + 1$ et $q(X)^k - 1$.

III.2 Courbes sur un corps de 5 mots machines

La Construction 6.6 de la taxonomie [FST10] dépend du résidu $k \pmod 6$. Vu que $13 \equiv 19 \equiv 1 \pmod 6$ et $11 \equiv 17 \equiv 5 \pmod 6$, nous ne donnons que les cas pertinents de la Construction 6.6.

Dans le cas $k \equiv 1 \pmod 6$, le premier $q(X)$, la trace du Frobenius $t(X)$ et l'ordre $\ell(X)$ sont donnés par :

$$\begin{aligned} q(X) &= \frac{1}{3}(X+1)^2(X^{2k} - X^k + 1) - X^{2k+1}, \\ t(X) &= -X^{k+1} + X + 1 \quad \text{et} \\ \ell(X) &= \Phi_{6k}(X), \end{aligned}$$

dans le cas où $k \equiv 5 \pmod 6$, ils sont donnés par :

$$\begin{aligned} q(X) &= \frac{1}{3}(X^2 - X + 1)(X^{2k} - X^k + 1) + X^{k+1}, \\ t(X) &= X^{k+1} + 1 \quad \text{et} \\ \ell(X) &= \Phi_{6k}(X). \end{aligned}$$

En essayant les différentes valeurs de k, nous pouvons dresser le tableau III.1.

k	$\deg(q)$	$\deg(\ell)$	ρ	$\log_2(\lvert x_0\rvert)$
11	24	20	1.20	$[12, 14[$
13	28	24	1.167	$[10, 12[$
17	36	32	1.125	$[8, 9[$
19	40	36	1.111	$[7, 8[$

TABLE III.1 – Paramètres et intervalles de recherche pour $k \in \{11, 13, 17, 19\}$

Remarquons que pour obtenir une courbe, il faut trouver un $x_0 \in \mathbb{Z}$ tel que $\ell(x_0)$ et $q(x_0)$ soient des entiers premiers. Nous choisissons x_0 satisfaisant $x_0 \equiv 2 \pmod 3$ pour que $q(x_0)$ soit un entier. La dernière colonne du tableau III.1 est un intervalle de recherche pour $\log_2(\lvert x_0\rvert)$ dans $[256/\deg(\ell), 320/\deg(q)[$ impliquant que $q(x_0)$ soit un entier de 5 mots machines et $\ell(x_0)$ soit un entier d'au moins 256 bits (pour l'aisance de lecture, les intervalles sont donnés avec des bornes arrondies à l'entier).

Après une recherche à l'aide d'un ordinateur, nous n'avons pas trouvé de solutions pour $k \in \{11, 17\}$. Pour $k = 13$, nous avons trouvé $x_0 = -2224$ et pour $k = 19$, seulement $x_0 = -145$. En inférant la convention d'attribution de nom utilisée dans la bibliothèque *relic-toolkit* [Ara+], la première courbe est baptisée `BW13-P310` et la seconde `BW19-P286`.

III.2.1 Courbe `BW13-P310`

Utilisons le paramètre $k = 13$ dans la Construction 6.6 [FST10] :

$$\begin{aligned} q(X) &= \frac{1}{3}(X+1)^2(X^{26} - X^{13} + 1) - X^{27}, \\ t(X) &= -X^{14} + X + 1 \quad \text{et} \\ \ell(X) &= \Phi_{78}(X). \end{aligned}$$

La graine $x_0 = -2224$ donne un $q(x_0)$ de 310 bits et un $\ell(x_0)$ de 267 bits, donc $\rho = 1.161$. Le corps fini correspondant $\mathbb{F}_{q^{13}}$ est de 4027 bits.

Pour trouver une constante b, nous pouvons augmenter la valeur de $\lvert b\rvert$ puis vérifier que le nombre de points sur la courbe $y^2 = x^3 + b$ est égal à $q(x_0) - t(x_0) + 1$. Nous trouvons $b = -17$. Définissons donc la courbe `BW13-P310` par l'équation $y^2 = x^3 - 17$.

Cette courbe est la même que celle proposée par Guillevic dans [Gui20]. Dans sa

minutieuse étude de la sécurité de cette courbe, elle a estimé que le coût du DLP dans le corps fini $\mathbb{F}_{q^{13}}$ est de 140 bits. Ainsi, la courbe `BW13-P310` atteint une sécurité d'au moins 128 bits.

III.2.2 Courbe `BW19-P286`

Utilisons le paramètre $k = 19$ dans la construction 6.6 [FST10] :

$$\begin{aligned} q(X) &= \frac{1}{3}(X+1)^2(X^{38} - X^{19} + 1) - X^{39}, \\ t(X) &= -X^{20} + X + 1 \quad \text{and} \\ \ell(X) &= \Phi_{114}(X). \end{aligned}$$

La graine $x_0 = -145$ donne un $q(x_0)$ de 286 bits et un $\ell(x_0)$ de 259 bits, donc $\rho = 1.105$. Le corps fini correspondant $\mathbb{F}_{q^{19}}$ est de 5427 bits.

Pour trouver une constante b, nous procédons comme précédemment. La plus petite valeur $|b|$ est $b = 31$. Définissons donc la courbe `BW19-P286` par l'équation $y^2 = x^3 + 31$.

Pour évaluer le coût du DLP dans $\mathbb{F}_{q^{19}}$, nous reprenons le travail fait par Guillevic [Gui20], exactement comme elle l'a fait pour la courbe précédente `BW13-P310`. Pour obtenir la courbe `BW19-P286` en utilisant Algorithme 3.1 [Gui20] de Guillevic, il faut utiliser les paramètres $k = 19$, $D = 3$, $e_0 = 13$ et la substitution d'indéterminée $X \mapsto -X$.

Avant de lancer le code estimant la sécurité de la courbe sur nos paramètres, nous appliquons la Variante 4 de [Gui20], donnant une meilleure estimation du NFS pour notre courbe. Nous obtenons finalement un coût pour le DLP dans $\mathbb{F}_{q^{19}}$ de 160 bits, ainsi la courbe `BW19-P286` atteint les 128 bits de sécurité.

Endomorphismes GLV sur les courbes `BW13-P310` et `BW19-P286`

Le discriminant du polynôme minimal du Frobenius peut s'écrire $-Df^2 = t^2 - 4q$, où $D > 0$ est le discriminant CM et t est la trace du Frobenius. L'endomorphisme $\phi : (x, y) \mapsto (\zeta x, y)$, avec ζ une racine primitive troisième de l'unité, correspond à une multiplication $(\zeta x, y) = [\lambda](x, y)$ dans $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$ lorsque q et ℓ sont des premiers distincts.

Pour nos courbes, le discriminant CM est $D = 3$ et le j-invariant est nul. On a vu dans la Sous-Section II.4.1 que

$$\zeta \equiv \frac{t-f}{2f} \pmod{q} \quad \text{et} \quad \lambda \equiv \frac{t-f-2}{2f} \pmod{\ell},$$

avec de possibles ajustements, vu que $(\zeta x, y) = [\lambda](x, y)$ ou $(\zeta^2 x, y) = [\lambda](x, y)$.

Dans le cas de `BW13-P310`, λ est de taille 146 bits, alors qu'il est seulement de 137 bits dans le cas de `BW19-P286`.

Commentaires sur `BW13-P310` et `BW19-286`.

Nous donnons dans la suite plusieurs points de comparaison entre notre nouvelle courbe `BW19-P286` avec `BW13-P310` mais aussi d'autres courbes provenant de familles de courbes populaires. Nous remarquons d'ores et déjà que `BW13-P310` et `BW19-P286` correspondent à notre stratégie d'optimisation du groupe $\mathbb{G}_1$ et, a priori, au détriment des autres groupes. Dans cette perspective, notre nouvelle courbe `BW19-P286` va plus loin que `BW13-P310` en réduisant la taille de q d'environ 25 bits. Cette différence est significativement amplifiée dans le contexte des appareils à ressources restreintes car elle utilise moins de mots machines. En effet, sur une architecture 32 bits, `BW19-P286` à un premier q d'un mot machine de moins que `BW13-P310`, ce qui va clairement améliorer ses performances.

III.3 Implémentation et comparaison

Nous avons implémenté l'arithmétique de $\mathbb{G}_1$ pour les deux courbes[1] grâce à *relic-toolkit* [Ara+] et fait quelques comparaisons avec d'autres courbes déjà disponibles dans *relic-toolkit* et visant les 128 bits de sécurité .

Les courbes sélectionnées pour la comparaison sont `BN-P446`, une courbe Barreto-Naehrig [BN06] définie sur un corps à 446 bits ; `K16-P339`, un courbe Kachisa-Schaefer-Scott [KSS08] de degré de plongement 16 définie sur un corps à 339 bits ; `B12-P446`, une courbe Barreto-Lynn-Scott [BLS03] de degré de plongement 12 définie sur un corps à 446 bits ; et `CP8-P544`, une courbe Cocks-Pinch [GMT20] de degré de plongement 8 définie sur un corps à 544 bits. Cette dernière courbe a été choisie car elle provient de récents travaux [GMT20 ; Gui20] la promouvant pour une sécurité de 128 bits. Nous avons inclus les courbes BN, BLS et KSS dans notre tableau car ces familles de courbes sont bien connues et mises à jour par Barbulescu et Duquesne [BD19]. Aussi, hormis nos courbes, seule la courbe `K16-P339` a été implémentée par nos soins, toutes les autres sont disponibles dans la librairie *relic*.

Toutes les courbes bénéficient d'endomorphismes GLV déjà implémentés ou, le cas échéant, que nous avons implémentés.

1. Une implémentation est disponible à l'adresse `https://framagit.org/rclarisse/relic`.

III.3.1 Opérations dans $\mathbb{G}_1$

Dans le tableau III.2, nous comparons le coût d'une exponentiation dans le groupe $\mathbb{G}_1$ lorsque *relic-toolkit* est compilé pour une architecture x64 ou x86 (*i.e.* les mots machines font 32 bits). Les temps sont donnés en microsecondes (le nombre d'itération était de un million) et les calculs ont été effectués sur un ordinateur équipé d'un processeur Intel Core i7-6600u CPU à 2.60 GHz.

	Courbe	`BW19-P286`	`BW13-P310`	`K16-P339`	`B12-P446`	`BN-P446`	`CP8-P544`
64-bit	mots	5	5	6	7	7	9
	temps (μs)	293	304	482	611	855	1058
	ratio	1	1.04	1.65	2.09	2.92	3.61
32-bit	mots	9	10	11	14	14	17
	temps (μs)	1010	1220	1664	2510	3600	4180
	ratio	1	1.21	1.65	2.49	3.56	4.14

TABLE III.2 – Comparaison des temps d'exécution d'une exponentiation dans $\mathbb{G}_1$. Pour les deux architectures, la ligne « mots » indique le nombre de mots machines utilisés pour stocker le premier q, la ligne « temps » indique le temps de calcul observé pour une exponentiation dans $\mathbb{G}_1$ arrondi à la microseconde ($1\ \mu\text{s} = 10^{-6}$ s), et la dernière ligne « ratio » indique le ratio entre le temps observé pour la courbe dans la colonne et la courbe `BW19-P286`.

Ce tableau montre une nette relation, presque quadratique, entre la complexité et le nombre de mots nécessaires pour représenter q. Cela met aussi en valeur l'inconvénient des courbes Barreto-Naehrig qui génèrent des courbes elliptiques d'ordre premier $\ell \sim q$. En effet, ce qui était considéré comme un avantage (les courbes d'ordre premier ont un unique groupe de points $\mathbb{F}_q$-rationnels, rendant les tests d'appartenance triviaux) devient une forte contrainte en forçant ℓ à croître inutilement. Cela ralentit l'exponentiation (vu que les exposants sont environ 75% plus grands que ceux d'autres courbes) et la taille des scalaires.

Dans tous les cas, ce tableau montre que notre courbe `BW19-P286` offre la meilleure performance pour $\mathbb{G}_1$, en particulier pour une architecture plus petite que 64 bits. Elle permet de diminuer de moitié le coût de l'exponentiation dans $\mathbb{G}_1$ comparée aux courbes traditionnelles telles que `B12-P446` et diminue aussi significativement la taille des éléments du groupe, ce qui correspond clairement aux besoins de certains protocoles cryptographiques

tel que celui présenté au début de ce chapitre.

III.3.2 Opérations dans $\mathbb{G}_2$

Les opérations dans $\mathbb{G}_2$ sont malheureusement les plus impactées par notre choix de degré de plongement premier. En effet, pour `BW13-P310` et `BW19-P286`, le groupe $\mathbb{G}_2$ est défini sur les corps $\mathbb{F}_{q^{13}}$ et $\mathbb{F}_{q^{19}}$ respectivement, alors que pour `B12-P446`, `BN-P446` et `CP8-P544` ce groupe $\mathbb{G}_2$ est défini sur $\mathbb{F}_{q^2}$ grâce à une tordue quartique ou sextique, et `K16-P339` a $\mathbb{G}_2$ défini sur $\mathbb{F}_{q^4}$ via une tordue quartique. Comme les courbes sont généralement exprimées avec le même modèle avec le même système de coordonnées, seul le coût d'une multiplication dans l'extension détermine le coût des opérations dans $\mathbb{G}_2$. Par exemple, à l'aide d'implémentation du type Karatsuba [Kar95], la multiplication dans $\mathbb{F}_{q^k}$ est environ $k^{\log_2(3)}$ plus coûteuse que celle dans $\mathbb{F}_q$.

III.3.3 Calcul du couplage

Le calcul du couplage est généralement séparé en deux parties : l'évaluation de la boucle de Miller et l'exponentiation finale. Ici, nous calculons un multiple du couplage ate optimal [Ver10], vu que nous prenons le couplage de Kim, Kim et Cheon [KKC13]. Les valeurs dans le tableau III.3 sont tirés de [Gui20 ; GMT20], agrémentées des nôtres comme suit.

Soient $\mathbf{m}_k$, $\mathbf{s}_k$, $\mathbf{i}_k$, $\mathbf{f}_k$ dénotant respectivement une multiplication, une mise au carré, une inversion, une application du Frobenius (*i.e.* une élévation à la puissance q) dans $\mathbb{F}_{q^k}$ Nous omettons l'indice lorsque l'opération est sur $\mathbb{F}_q$ (*e.g.* $\mathbf{m} = \mathbf{m}_1$). Comme $k \in \{13, 19\}$ est premier, nous estimons un coût $\mathbf{m}_k = \mathbf{s}_k = k^{\log_2(3)}\mathbf{m}$ grâce à une implémentation du type Karatsuba et $\mathbf{f}_k = (k-1)\mathbf{m}$ comme dans [GMT20].

Boucle de Miller

Guillevic [Gui20, Équation (7)] donne une borne inférieure sur le coût de la boucle de Miller. Pour les deux courbes `BW13-P310` et `BW19-P286`, la boucle de Miller du couplage ate optimal est de longueur $u^2 - up + p^2$, un multiple de ℓ [Ver10].

Pour `BW13-P310`, la boucle de Miller a une longueur $u^2 - up + p^2$, où $u = 2224$ est un entier de 12 bits de poids de Hamming 4 et p est un premier de 310 bits. De son Équation (7) [Gui20], Guillevic obtient $949\mathbf{m} + 313\mathbf{m}_{13} + 177\mathbf{s}_{13} + 5\mathbf{f}_{13} + 2\mathbf{i}_{13}$. En

substituant $\mathbf{m}_{13} = \mathbf{s}_{13} = 59\mathbf{m}$ et $\mathbf{f}_{13} = 12\mathbf{m}$ dans cette formule, on obtient une borne inférieure pour le coût de la boucle de Miller du couplage ate optimal, *i.e.* $29919\mathbf{m} + 2\mathbf{i}_{13}$.

Pour `BW19-P286`, la longueur de la boucle est $u^2 - up + p^2$, où $u = 145$ est un entier de 8 bits de poids de Hamming 3 et p est un premier de 286 bits. À l'aide de la même équation de Guillevic [Gui20], nous obtenons $912\mathbf{m} + 212\mathbf{m}_{19} + 115\mathbf{s}_{19} + 5\mathbf{f}_{19} + 2\mathbf{i}_{19}$. Nous substituons $\mathbf{m}_{19} = \mathbf{s}_{19} = 107\mathbf{m}$ et $\mathbf{f}_{19} = 18\mathbf{m}$ dans cette formule, ce qui donne une borne inférieure sur le coût de la boucle de Miller du couplage ate optimal, *i.e.* $35991\mathbf{m} + 2\mathbf{i}_{19}$.

Exponentiation finale

Comme à l'accoutumé, l'exponentiation finale $(q^k - 1)/\ell$ du couplage de Ate optimal est séparée en une partie facile $(q^k - 1)/\Phi_k(q)$ et une partie difficile $\Phi_k(q)/\ell$. Vu que $k \in \{13, 19\}$ est premier, la partie facile est simplement $q-1$, coûtant $\mathbf{f}_k + \mathbf{i}_k$. Pour la partie difficile, Kim, Kim et Cheon [KKC13] (ou alternativement Fuentes-Castañeda, Knapp et Rodríguez-Henríquez [FKR12]) ont remarqué que $\Phi_k(q)/\ell$ peut être décomposé en base q pour tirer usage du Frobenius et que les coefficients peuvent être réduits en cherchant un vecteur court dans un réseau euclidien spécifiquement construit. Cependant, au lieu d'élever à la puissance $\Phi_k(q)/\ell$, cette méthode [KKC13] élève à une puissance multiple $m\Phi_k(q)/\ell$.

Plus précisément, ils écrivent

$$m\frac{\Phi_k(q)}{\ell} = \sum_{i=0}^{k-2} a_i q^i$$

et trouvent les $k-1$ coefficients $(a_i)_{0 \leqslant i \leqslant k-2}$ comme le plus court vecteur du réseau de dimension $k-1$ engendré par les lignes de la matrice

$$\begin{bmatrix} \frac{\Phi_k(q)}{\ell} & 0 & 0 & \cdots & 0 \\ -q & 1 & 0 & \cdots & 0 \\ -q^2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & & \\ -q^{k-2} & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

Puis, ils calculent le Frobenius de l'élément qu'ils veulent élever, jusqu'à la $(k-1)$-ième puissance de q, *i.e.* le Frobenius est itéré $k-1$ fois, et cela coûte $(k-2)\mathbf{f}_k$.

Si les exposants a_i étaient plus longs, nous aurions eu besoin de $(2^{k-1} - k)\mathbf{m}_k$ pour

calculer toutes les combinaisons des $k-1$ puissances du Frobenius. En revanche, ici, nous n'utilisons pas toutes ces combinaisons, seulement environ $\mathcal{O}(\log_2 q)$. La longueur du multi-exposant est alors $\max_i\{\lfloor \log_2 a_i \rfloor\}$, résultant en un coût moyen de l'exponentiation finale de

$$(k-1)\mathbf{f}_k + (\mathcal{O}(\log_2 q) + \max_i\{\lfloor \log_2 a_i \rfloor\})\mathbf{m}_k + \max_i\{\lfloor \log_2 a_i \rfloor\}\mathbf{s}_k + \mathbf{i}_k,$$

où quelques inversions sont omises à cause du signe de certains a_i qui ne sont pas encore calculés.

Pour `BW13-P310`, la valeur de $\max_i\{\lfloor \log_2 a_i \rfloor\}$ est 287 et parmi les 12 a_i, 8 sont négatifs. Seules 191 combinaisons différentes des puissances du Frobenius sont utilisées et cela coûte $341\mathbf{m}_{13}$ pour les calculer. Aussi, il y a 5 positions (dans l'expansion binaire) où tous les a_i ont leur bit à 0, il n'y a donc pas de multiplication en ces positions dans la multi-exponentiation, qui requiert alors $282\mathbf{m}_{13} + 287\mathbf{s}_{13}$. Combinons tous ces résultats, l'exponentiation finale coûte au plus $12\mathbf{f}_{13}+623\mathbf{m}_{13}+287\mathbf{s}_{13}+9\mathbf{i}_{13}$, *i.e.* $53834\mathbf{m}+9\mathbf{i}_{13}$. Remarquons que le facteur multiplicatif est $m = -29590513987728624149724357977601 75$.

Pour `BW19-P286`, la valeur de $\max_i\{\lfloor \log_2 a_i \rfloor\}$ est 271 et 12 des 18 a_i sont négatifs. Seules 222 combinaisons différentes de puissances du Frobenius sont utilisées et leur coût de calcul est $1028\mathbf{m}_{19}$. Il en résulte que la multi-exponentiation requiert $271(\mathbf{m}_{19} + \mathbf{s}_{19})$. Ces résultats donnent une majoration du coût de l'exponentiation finale $18\mathbf{f}_{19} + 1299\mathbf{m}_{19} + 271\mathbf{s}_{19} + 13\mathbf{i}_{19}$, c'est-à-dire $160824\mathbf{m} + 13\mathbf{i}_{19}$. Ici, le facteur multiplicatif est $m = -379212520522736361724291988800021 44$.

	`BW19-P286`	`BW13-P310`	`K16-P339`	`B12-P446`	`BN-P446`	`CP8-P544`
M.	$35991\mathbf{m} + 2\mathbf{i}_{19}$	$29919\mathbf{m} + 2\mathbf{i}_{13}$	$7691\mathbf{m}$	$7805\mathbf{m}$	$11620\mathbf{m}$	$4502\mathbf{m}$
E.	$160824\mathbf{m} + 13\mathbf{i}_{19}$	$53834\mathbf{m} + 9\mathbf{i}_{13}$	$18235\mathbf{m}$	$7723\mathbf{m}$	$5349\mathbf{m}$	$7056\mathbf{m}$
T.	$196815\mathbf{m} + 15\mathbf{i}_{19}$	$83753\mathbf{m} + 11\mathbf{i}_{13}$	$25926\mathbf{m}$	$15528\mathbf{m}$	$16969\mathbf{m}$	$11558\mathbf{m}$

TABLE III.3 – Coût en opérations pour la boucle de Miller (M.), exponentiation finale (E.) et couplage total (T.). Les inversions dans un corps de degré d'extension impair sont indiquées car elles plus coûteuse que lorsque le degré est pair (celles-ci ne coûtent en général qu'une conjugaison dans le sous-corps quadratique).

Nous constatons du tableau III.3 que le coût du couplage pour la courbe `BW19-P286` est approximativement 12 fois plus important que celui pour `B12-P446`. Malgré tout, la comparaison du temps d'exécution d'une multiplication entre les deux corps finis montre

qu'une multiplication est deux fois plus rapide dans le corps de 286 bits (90 ns) par rapport à celui à 446 bits (190 ns). D'où notre estimation que le couplage sur la courbe `BW19-P286` est 6 fois plus lent que celui sur la courbe `B12-P446`.

Conclusion du troisième chapitre

Nous avons motivé un changement de point de vue sur la génération de courbe elliptique à couplage : remplacer la prépondérance de l'équilibre entre les diverses opérations (exponentiation dans les groupes et calcul du couplage) par l'optimisation de quelques unes d'entre elles pour répondre à des besoins cryptographiques précis.

Nous nous sommes concentrés sur des courbes elliptiques avec une exponentiation rapide dans le premier groupe du couplage, après avoir remarqué en introduction du chapitre que certaines instanciations de schéma cryptographique, *e.g.* celles de la famille des signatures de groupe, pourraient bénéficier de telles courbes. Nous avons donc décrit une nouvelle courbe particulièrement adaptée aux protocoles faisant un important usage de l'exponentiation dans le premier groupe du couplage. Cette courbe est en effet deux fois plus rapide dans ce groupe comparée à une courbe BLS définie sur un corps de 446 bits, au prix d'un couplage plus lent.

Chapitre IV

Signatures de groupe

Les signatures de groupe, introduites par David Chaum et Eugène van Heyst [Cv91], permettent à un membre d'un groupe de signer au nom du groupe. L'intérêt d'une telle signature est qu'elle est anonyme, *i.e.* on ne peut pas savoir qui l'a produite, à l'exception d'une entité spécifique, l'autorité d'ouverture, qui peut « ouvrir » toute signature de groupe valide.

Combiner des propriétés apparemment aussi contradictoires que l'anonymat et l'authentification s'est montré délicat, la première solution véritablement pratique ayant été proposée par Ateniese *et al.* [Ate+00]. Quelques années plus tard, Bellare, Micciancio et Warinschi [BMW03] proposèrent le premier modèle de sécurité (modèle BMW) pour les signatures de groupe statique, qui a plus tard été étendu au cas des signatures de groupe dynamique par Bellare, Shi et Zhang [BSZ05] (donnant lieu au modèle BSZ). En plus d'avoir procuré une manière d'évaluer les schémas existants, ces travaux précurseurs ont introduit une construction générique qui est devenue le plan de la plupart des signatures de groupe ultérieures.

De façon informelle, un membre de groupe de cette construction générique reçoit d'un gestionnaire de groupe un certificat (*i.e.* une signature numérique) τ sur sa clé publique pk au moment de rejoindre le groupe. Pour produire une signature de groupe sur un message m, il commence par générer une signature numérique σ sur m (en utilisant la clé de signature sk correspondante) puis chiffre σ et τ. Enfin, il fournit une preuve sans divulgation de connaissance et non-interactive (NIZK [1]) que tous les éléments sont bien formés. Cette approche en trois étapes est connue dans la littérature sous le nom *Sign-Encrypt-Prove* (SEP, Signer-Chiffrer-Prouver en français).

La force du paradigme SEP est qu'il est fondé sur des primitives cryptographiques standards pour lesquelles il existe plusieurs instanciations. Malheureusement, cela conduit à des constructions relativement complexes à cause des exigences de sécurité placées sur chaque bloc, mais principalement à cause de la complexité résultant des preuves NIZK.

1. *Non-Interactive Zero-Knowledge proof* en anglais.

En effet, le signataire doit prouver, sans révéler σ ou τ, que la signature de groupe est un chiffrement valide de la signature σ qui elle-même a été générée grâce à des clés certifiées par le gestionnaire de groupe.

Un tel énoncé est difficile à prouver et cela empire si l'on souhaite prouver la sécurité du schéma sans l'utilisation du modèle de l'oracle aléatoire (ROM [2]). En effet, les preuves NIZK sont bien plus complexes dans cette situation. Même en utilisant la méthodologie de Groth et Sahai [GS08], une signature de groupe contient encore des douzaines d'éléments (voir *e.g.* [Gro07]). Il est toujours possible de substituer les preuves NIZK par des ZK-SNARKs [3] (voir *e.g.* [Gro16]) qui sont assez compacts et permettent une vérification rapide, mais avec le désavantage d'être notamment plus coûteux en calculs pour le prouveur, *i.e.* la personne émettant les signatures.

Une question naturelle découlant de ces observations est de savoir s'il est possible de construire un schéma plus efficace en utilisant un paradigme différent de SEP. Bichsel *et al.* [Bic+10] ont donné une réponse intéressante à cette question. Ils ont en effet proposé une alternative efficace, au prix d'une notion légèrement plus faible d'anonymat. Cela leur a permis de passer outre le résultat d'Abdalla et Warinschi [AW04] et donc d'éviter le chiffrement. Plus précisément, leur idée pour retirer le chiffrement est d'utiliser des certificats τ randomisables [CL04] et de fusionner la signature σ avec la preuve NIZK, donnant ainsi une signature de connaissance [FS87]. Cette construction est plutôt efficace et peut être encore améliorée en l'instanciant avec les signatures randomisables de Pointcheval et Sanders [PS16].

Une autre alternative utilisant des signatures de classe d'équivalence [FHS19] a été proposée par Derler et Slamanig [DS18]. Elle présente des points communs avec [Bic+10], tel que l'absence de chiffrement explicite, mais elle parvient à atteindre l'anonymat total en échange d'une complexité accrue. Malencontreusement, les schémas [DS18] et [Bic+10] reposent tout deux sur des signatures de connaissance et correspondent donc plutôt au modèle de l'oracle aléatoire.

Récemment, Backes *et al.* [Bac+18] ont proposé une méthode de construction différente, faisant appel à une nouvelle primitive nommée signature avec clés publiques flexibles. Cela produit des constructions sûres sans avoir recours au ROM et avec une efficacité améliorée par rapport à l'état de l'art. Cependant, les signatures de groupe produites sont trois fois plus grandes que celles dans le ROM et requièrent plus de calculs

2. *Random Oracle Model* en anglais.
3. *Zero-Knowledge Succinct Non-interactive ARgument of Knowledge* en anglais.

pour être générées.

Plus généralement, les concepteurs de schémas de signature de groupe sont confrontés entre choisir de prouver la sécurité de leur système sans avoir recours au ROM, ou bien d'en favoriser l'efficacité en se plaçant dans le ROM, dont les limites sont connues [CGH04].

Dans ce chapitre, nous nous proposons de détailler la contribution faite dans l'article *Group Signature Without Random Oracles from Randomizable Signatures* [CS20], co-écrit avec Olivier Sanders et publiée à la conférence internationale « Provable and Practical Security » en septembre 2020. Cette contribution est un nouveau schéma de signature de groupe à base de couplage qui évite le ROM et réduit de moitié la taille des signatures et la complexité des calculs comparé à l'état de l'art [Bac+18].

IV.1 Signatures de groupe dynamique

Le modèle de sécurité utilisé est le modèle BSZ [BSZ05] (des auteurs Mihir Bellare, Haixia Shi et Chong Zhang) où le gestionnaire du groupe et l'autorité d'ouverture sont deux entités distinctes (comprendre aussi indépendamment corruptibles), et où les utilisateurs peuvent devenir membre à tout moment à l'aide d'un protocole permettant de rejoindre le groupe (d'où le dynamique).

IV.1.1 Syntaxe

Une signature de groupe est définie par les algorithmes suivants, impliquant trois types d'entités : un gestionnaire de groupe, une autorité d'ouverture et des utilisateurs. Chacun des utilisateurs est identifié par un indice public $i \in \mathbb{N}^*$.

- $\texttt{Setup}(1^\lambda)$: Cet algorithme retourne les paramètres publics pp correspondants à un paramètre de sécurité λ.
- $\texttt{UKeygen}(pp)$: Cet algorithme retourne la paire de clés $(\mathsf{sk}, \mathsf{pk})$ de l'utilisateur par rapport aux paramètres publics pp. On suppose que pk est public et que l'on peut en obtenir un authentique duplicata.
- $\texttt{OKeygen}(pp)$: Cet algorithme retourne la paire de clés $(\mathsf{osk}, \mathsf{opk})$ de l'autorité d'ouverture par rapport à pp.
- $\texttt{GKeygen}(pp)$: Cet algorithme retourne la paire de clés $(\mathsf{gsk}, \mathsf{gpk})$ du gestionnaire de groupe ainsi qu'un registre public $\mathbf{Reg}$, par rapport aux paramètres publics pp.
- `Join` : Ceci est un protocole interactif impliquant le gestionnaire du groupe et un

utilisateur i qui veut devenir membre du groupe. Le gestionnaire a comme entrée du protocole $(\mathsf{gsk}, \mathbf{Reg}, \mathsf{opk}, \mathsf{pk}_i)$ alors que l'utilisateur a $(\mathsf{gpk}, \mathsf{opk}, \mathsf{sk}_i)$. Si le protocole n'échoue pas, l'utilisateur obtient une clé usk_i pour signer au nom du groupe, alors que le gestionnaire met à jour le registre **Reg**.
Sinon, les deux parties retournent $\perp$.

— $\mathtt{Sign}(\mathsf{usk}_i, m)$: Cet algorithme retourne une signature de groupe σ du message m sous la clé de signature usk_i.

— $\mathtt{Verify}(\mathsf{gpk}, \sigma, m)$: Avec comme entrées la clé publique du gestionnaire gpk, une signature de groupe σ et un message m, cet algorithme retourne un bit $b \in \{0, 1\}$.

— $\mathtt{Open}(\mathsf{osk}, \mathsf{gpk}, \mathbf{Reg}, \sigma, m)$: Avec comme entrées la clé privée de l'autorité d'ouverture osk, la clé publique du gestionnaire gpk, le registre **Reg**, une signature de groupe σ et un message m, cet algorithme retourne soit 0, soit $\perp$, soit un indice $i \in \mathbb{N}^*$ avec une preuve $\overline{\pi}$.

— $\mathtt{Judge}(\mathsf{gpk}, \mathbf{Reg}, \sigma, m, i, \overline{\pi})$: Avec comme entrées la clé publique du gestionnaire gpk, le registre **Reg**, une signature de groupe σ, un message m, un indice $i \in \mathbb{N}^*$ et une preuve $\overline{\pi}$, cet algorithme retourne un bit $b \in \{0, 1\}$.

IV.1.2 Modèle de sécurité

Il est attendu d'une signature de groupe qu'elle soit correcte et permette l'anonymat, la traçabilité et la non-diffamation [4] des membres du groupe.

Informellement, la *correction* du schéma [BSZ05] signifie que tout utilisateur ayant rejoint le groupe est capable de produire des signatures de groupe valides σ (*i.e.* telles que l'algorithme `Verify` retourne 1) sur n'importe quel message m. De plus, l'autorité d'ouverture doit être capable d'ouvrir de telles signatures, *i.e.* recouvrir l'identité i du signataire, et produire une preuve publiquement vérifiable que l'utilisateur i est bien celui qui a produit ces signatures.

L'*anonymat* implique que les signatures sont anonymes sauf pour l'autorité d'ouverture. La *traçabilité* demande à ce que personne ne puisse produire des signatures valides qui ne puissent être rattachées à un utilisateur via la procédure `Open`. La *non-diffamation* d'un schéma de signature de groupe signifie que personne ne peut être faussement accusé d'avoir produit une signature. Ces notions correspondent à des jeux de sécurité, décrits dans la Figure IV.1, et font usage des oracles suivants.

4. Traduction de l'anglais *non-frameability* [San15].

— $\mathcal{O}\texttt{Add}(i)$ est un oracle permettant d'ajouter un nouvel utilisateur i. Il lance l'algorithme $\texttt{UKeygen}(pp)$ pour obtenir $(\mathsf{sk}_i, \mathsf{pk}_i)$ et renvoie pk_i. Si i a déjà été utilisé, alors il renvoie $\perp$.
— $\mathcal{O}\texttt{Join}_U(i)$ est un oracle jouant la partie de l'utilisateur dans le protocole `Join`. Il peut être utilisé par un adversaire $\mathcal{A}$ incarnant le rôle d'un gestionnaire de groupe corrompu. Cet oracle renvoie $\perp$ si i a déjà rejoint le groupe ou si i n'existe pas.
— $\mathcal{O}\texttt{Corrupt}(i)$ est un oracle qui renvoie toutes les clés privées de l'utilisateur i. L'utilisateur i est alors dit *corrompu*. Tout utilisateur non-corrompu est considéré *honnête*.
— $\mathcal{O}\texttt{Join}_{GM}()$ [5] est la contrepartie de l'oracle $\mathcal{O}\texttt{Join}_U$ qui peut être utilisé par un utilisateur corrompu pour rejoindre le groupe.
— $\mathcal{O}\texttt{Sign}(i, m)$ est un oracle qui renvoie $\texttt{Sign}(\mathsf{usk}_i, m)$, pourvu que i soit un utilisateur honnête faisant partie du groupe.
— $\mathcal{O}\texttt{Open}(\sigma, m)$ est un oracle qui renvoie $\texttt{Open}(\mathsf{osk}, \mathsf{gpk}, \mathbf{Reg}, \sigma, m)$.
— $\mathcal{O}\texttt{Ch}_b(i_0, i_1, m)$ est un oracle qui prend en entrées les indices de deux utilisateurs honnêtes et renvoie $\texttt{Sign}(\mathsf{usk}_{i_b}, m)$.

Soit $\mathcal{A}$ un adversaire polynomial probabiliste. Un schéma de signature de groupe est

— anonyme si $\texttt{Adv}^{an}(\mathcal{A}) = |\Pr[\texttt{Exp}^{an}_{\mathcal{A}}(1^\lambda) = 1] - 1/2|$ est négligeable pour tout $\mathcal{A}$;
— traçable si $\texttt{Adv}^{tra}(\mathcal{A}) = \Pr[\texttt{Exp}^{tra}_{\mathcal{A}}(1^\lambda) = 1]$ est négligeable pour tout $\mathcal{A}$;
— non-diffamatoire si $\texttt{Adv}^{nf}(\mathcal{A}) = \Pr[\texttt{Exp}^{nf}_{\mathcal{A}}(1^\lambda) = 1]$ est négligeable pour tout $\mathcal{A}$.

Le modèle de sécurité introduit par Bellare, Shi et Zhang [BSZ05] ne pose aucune restriction sur les requêtes $\mathcal{O}\texttt{Corrupt}$ dans l'expérience d'anonymat. Cela signifie que l'adversaire est autorisé à corrompre les utilisateurs « challenge » (*i.e.* ceux qui sont impliqués dans les requêtes $\mathcal{O}\texttt{Ch}$). Cela correspond à la plus forte notion d'anonymat, parfois appelée *full anonymity* [6] ou *CCA-2 anonymity* [7] (voir *e.g.* [DS18]), où l'anonymat persiste même lorsque les clés privées d'un utilisateur sont divulguées.

Remarque IV.1.1. Le modèle BSZ [BSZ05] définit des propriétés de sécurité fortes qui suffisent dans la plupart des cas. Cependant, il est possible dans certaines situations d'assouplir ces propriétés, ce qui permet des constructions plus efficaces. Cela est particulièrement vrai pour la propriété d'anonymat pour laquelle il existe des variantes classiques, comme la

5. Ici, l'indice *GM* signifie *Group Manager* pour gestionnaire de groupe.
6. Traduit par « anonymat total » en français.
7. Traduit par « anonymat CCA-2 » en français.

$\mathtt{Exp}^{an}_{\mathcal{A}}(1^\lambda)$ – Jeu de Sécurité pour l'Anonymat

1. $pp \leftarrow \mathtt{Setup}(1^\lambda)$
2. $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathtt{OKeygen}(pp)$
3. $(\mathsf{gsk}, \mathsf{gpk}) \leftarrow \mathtt{GKeygen}(pp)$
4. $b \xleftarrow{\$} \{0,1\}$
5. $b^* \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{Add},\mathcal{O}\mathtt{Join}_U,\mathcal{O}\mathtt{Corrupt},\mathcal{O}\mathtt{Sign},\mathcal{O}\mathtt{Open},\mathcal{O}\mathtt{Ch}_b}(\mathsf{gsk}, \mathsf{opk})$
6. Si $\mathcal{O}\mathtt{Open}$ est appelé sur la sortie de $\mathcal{O}\mathtt{Ch}_b$, alors retourner 0
7. Retourner $(b = b^*)$

$\mathtt{Exp}^{tra}_{\mathcal{A}}(1^\lambda)$ – Jeu de Sécurité pour la Traçabilité

1. $pp \leftarrow \mathtt{Setup}(1^\lambda)$
2. $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathtt{OKeygen}(pp)$
3. $(\mathsf{gsk}, \mathsf{gpk}) \leftarrow \mathtt{GKeygen}(pp)$
4. $(\sigma, m) \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{Add},\mathcal{O}\mathtt{Join}_{GM},\mathcal{O}\mathtt{Corrupt},\mathcal{O}\mathtt{Sign}}(\mathsf{gpk}, \mathsf{osk})$
5. Si $\perp \leftarrow \mathtt{Open}(\mathsf{osk}, \mathsf{gpk}, \mathbf{Reg}, \sigma, m)$, alors retourner 1
6. Si $(i, \pi) \leftarrow \mathtt{Open}(\mathsf{osk}, \mathsf{gpk}, \mathbf{Reg}, \sigma, m)$ et $0 \leftarrow \mathtt{Judge}(\mathsf{gpk}, \mathbf{Reg}, \sigma, m, i, \pi)$, alors retourner 1
7. Retourner 0

$\mathtt{Exp}^{nf}_{\mathcal{A}}(1^\lambda)$ – Jeu de Sécurité pour la Non-Diffamation

1. $pp \leftarrow \mathtt{Setup}(1^\lambda)$
2. $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathtt{OKeygen}(pp)$
3. $(\mathsf{gsk}, \mathsf{gpk}) \leftarrow \mathtt{GKeygen}(pp)$
4. $(\sigma, m, i, \pi) \leftarrow \mathcal{A}^{\mathcal{O}\mathtt{Add},\mathcal{O}\mathtt{Join}_U,\mathcal{O}\mathtt{Corrupt},\mathcal{O}\mathtt{Sign}}(\mathsf{gsk}, \mathsf{osk})$
5. Si σ a été retournée par $\mathcal{O}\mathtt{Sign}$, alors retourner 0
6. Si i est corrompu, alors retourner 0
7. Retourner $\mathtt{Judge}(\mathsf{gpk}, \mathbf{Reg}, \sigma, m, i, \pi)$

FIGURE IV.1 – Jeux de sécurité pour les signatures de groupe

CPA anonymity[8] [BBS04 ; BW06] ou *selfless anonymity*[9] [BS04 ; Bic+10 ; PS16]. La première notion retire l'oracle $\mathcal{O}\texttt{Open}$ dans le jeu d'anonymat mais les utilisateurs demeurent anonymes même lorsque leurs clés privées fuitent. À l'inverse, l'anonymat désintéressé autorise les requêtes à $\mathcal{O}\texttt{Open}$ mais les utilisateurs ne sont désormais plus anonymes lorsque leurs clés privées fuitent. Ces deux notions ne sont pas comparables et correspondent donc à des contextes différents. La construction que nous décrivons à la section suivante réalise ces deux notions. Il est aussi intéressant de remarquer qu'elle permet l'anonymat total pour le modèle introduit par Bellare, Micciancio et Warinschi [BMW03] (le modèle BMW), où le gestionnaire est aussi l'autorité d'ouverture.

IV.2 Notre signature de groupe

Notre schéma de signature de groupe utilise une signature numérique et un chiffrement à clé publique comme bloc de base, dont nous donnons simplement les prérequis de sécurité. Nous donnons aussi les types d'équation qui peuvent être prouvés par le système de preuve Groth-Sahai [GS08], qui sont utilisés lorsqu'un utilisateur rejoint le groupe. En revanche, les certificats d'appartenance au groupe et les signatures de groupe dans notre schéma sont, respectivement, des signatures de classes d'équivalence Fuchsbauer-Hanser-Slamanig [FHS15] et des signatures Pointcheval-Sanders [PS16], dont nous donnons les définitions en Sous-Section IV.2.1. Nous détaillons l'interaction entre ces deux schémas en Sous-Section IV.2.4.

Dans la suite, l'élément neutre d'un groupe $\mathbb{G}$ est noté $1_{\mathbb{G}}$ et $\mathbb{G}^*$ signifie $\mathbb{G} \setminus \{1_{\mathbb{G}}\}$. Si le groupe $\mathbb{G}$ est d'ordre p, alors on peut dire de façon interchangeable que $a \in \mathbb{Z}/p\mathbb{Z}$ ou que a est un scalaire. Pour un ensemble fini X, la notation $x \xleftarrow{\$} X$ veut dire que x est un élément de X tiré de façon aléatoire en suivant une distribution uniforme.

Les couplages (voir Section II.3) considérés sont des couplages de type 3 d'ordre premier p. Pour un couplage $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ de type 3, il n'existe pas de morphisme de groupes facilement calculable (*i.e.* en temps polynomial) entre $\mathbb{G}_1$ et $\mathbb{G}_2$. Pour marquer la différence entre $\mathbb{G}_1$ et $\mathbb{G}_2$, on note les éléments du second groupe avec un tilde (*e.g.* $\tilde{g}$).

Signature numérique

Un schéma de signature numérique Σ est défini par les quatre algorithmes :

8. Traduit par « anonymat CPA » en français.
9. Traduit par « anonymat désintéressé » [San15] en français.

— $\texttt{Setup}(1^\lambda)$: Cet algorithme retourne les paramètres publics pp correspondant à un paramètre de sécurité λ.
— $\texttt{Keygen}(pp)$: Étant donnés pp, cet algorithme retourne les clés de signature et de vérification $(\mathsf{sk}, \mathsf{pk})$.
— $\texttt{Sign}(\mathsf{sk}, m)$: Cet algorithme retourne une signature σ du message m sous la clé de signature sk.
— $\texttt{Verify}(\mathsf{pk}, m, \sigma)$: Étant donnés la clé de vérification pk, le message m et sa supposée signature σ, cet algorithme retourne 1 si σ est une signature valide m sous pk, et 0 sinon.

La notion de sécurité standard pour une signature est EUF-CMA [10] [GMR88] : cela veut dire qu'il est difficile, même en ayant accès à un oracle de signature, de retourner une paire valide (m, σ), *i.e.* $\texttt{Verify}(\mathsf{pk}, m, \sigma) = 1$, pour un message m n'ayant jamais été demandé à l'oracle de signature.

Chiffrement à clé publique

Nous utilisons de façon marginale un schéma de chiffrement Γ, servant essentiellement à rejoindre le groupe et ouvrir des signatures. Il est défini par les algorithmes suivants :

— $\texttt{Keygen}(1^\lambda)$: Cet algorithme retourne une paire de clés de déchiffrement et chiffrement $(\mathsf{sk}, \mathsf{pk})$ correspondant à un paramètre de sécurité λ.
— $\texttt{Encrypt}(\mathsf{pk}, m)$: Cet algorithme retourne un chiffré c d'un message m sous pk.
— $\texttt{Decrypt}(\mathsf{sk}, c)$: À l'aide de la clé sk, cet algorithme retourne un déchiffrement m ou $\perp$.

Plusieurs notions de sécurité existent pour un schéma à clé publique mais notre signature de groupe n'a besoin que de la sécurité IND-CCA2 [11]. Informellement, cela signifie qu'aucun adversaire, même en ayant accès à un oracle de déchiffrement, n'est capable de distinguer un chiffrement de m_0 d'un chiffrement de m_1, où m_0 et m_1 sont choisis par l'adversaire.

Système de preuve Groth-Sahai

Notre schéma utilise une preuve lors de l'enregistrement auprès du gestionnaire. Nous esquissons ici le système de preuve Groth-Sahai.

10. Pour *existential unforgeability under chosen message attacks* en anglais
11. Pour *indistinguishability under adaptive chosen ciphertext attacks* en anglais

Dans [GS08], Groth et Sahai proposent un système de preuve non-interactive, dans le modèle de la chaîne de référence commune (CRS [12]), qui permet la plupart des relations de « groupes bilinéaires », *i.e.* trois groupes $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ dans une configuration de couplage d'ordre premier p par l'application e bilinéaire. La CRS peut être générée de manière à obtenir soit des preuves vérifiant la propriété de validité, soit des preuves vérifiant celle d'indistingabilité des témoins. Le point important est qu'il est difficile de déterminer quel type de CRS a été généré (dans notre cas, la difficulté repose sur l'hypothèse SXDH, voir Sous-Section IV.2.2). Pour prouver que des variables satisfassent à un ensemble de relations, le prouveur s'engage sur leur valeur (en utilisant des éléments de la CRS) et puis calcule une preuve par relation. Ces preuves peuvent être pour des équations :

— de produits de couplages, pour des variables $\{X_i\}_{i=1}^n \in \mathbb{G}_1$, $\{\tilde{Y}_i\}_{i=1}^n \in \mathbb{G}_2$ et des constantes $t_T \in \mathbb{G}_T$, $\{A_i\}_{i=1}^n \in \mathbb{G}_1$, $\{\tilde{B}_i\}_{i=1}^n \in \mathbb{G}_2$, $\{a_{i,j}\}_{i,j=1}^n \in \mathbb{Z}/p\mathbb{Z}$:

$$\prod_{i=1}^{n} e(A_i, \tilde{Y}_i) \prod_{i=1}^{n} e(X_i, \tilde{B}_i) \prod_{i=1}^{n} \prod_{j=1}^{n} e(X_i, \tilde{Y}_j)^{a_{i,j}} = t_T;$$

— de multi-exponentiations, pour les variables $\{X_i\}_{i=1}^n \in \mathbb{G}_k$, $\{y_i\}_{i=1}^n \in \mathbb{Z}/p\mathbb{Z}$ et les constantes $T \in \mathbb{G}_k$, $\{A_i\}_{i=1}^n \in \mathbb{G}_k$, $\{b_i\}_{i=1}^n \in \mathbb{Z}/p\mathbb{Z}$, $\{a_{i,j}\}_{i,j=1}^n \in \mathbb{Z}/p\mathbb{Z}$:

$$\prod_{i=1}^{n} A_i^{y_i} \prod_{j=1}^{n} X_j^{b_j} \prod_{i=1}^{n} \prod_{j=1}^{n} X_j^{y_i \cdot a_{i,j}} = T, \qquad \text{où } k \in \{1, 2\}.$$

Le système Groth-Sahai permet aussi de générer des preuves NIZK pour des équations de multi-exponentiation ou des équations de produits de couplages telles que $t_T = 1_{\mathbb{G}_T}$ ou $t_T = \prod e(A_i, \tilde{B}_i)$.

IV.2.1 Deux signatures randomisables

On décrit ici les deux schémas de signatures numériques randomisables qui sont au cœur de notre construction.

Signature Pointcheval-Sanders

Pointcheval et Sanders ont proposé dans leur article [PS16] de 2016 une signature randomisable, *i.e.* un schéma permettant de dériver des signatures σ' randomisées à partir

12. De l'anglais *common reference string*.

d'une signature valide σ. Leur schéma adapte les signatures Camenisch-Lysyanskaya [CL03] aux couplages de type 3.

Le schéma de signature PS est donné par les quatre algorithmes suivants :

— $\texttt{Setup}(1^\lambda)$: Étant donné un paramètre de sécurité λ, cet algorithme retourne les paramètres publics du système $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, où $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ est un couplage de type 3 d'ordre p premier.

— $\texttt{Keygen}(pp)$: Cet algorithme choisit $\tilde{g} \xleftarrow{\$} \mathbb{G}_2^*$ et $(x, y) \xleftarrow{\$} (\mathbb{Z}/p\mathbb{Z})^2$, calcule ensuite $(\tilde{X}, \tilde{Y}) \leftarrow (\tilde{g}^x, \tilde{g}^y)$ et définit la clé privée sk comme (x, y) et la clé publique pk comme $(\tilde{g}, \tilde{X}, \tilde{Y})$.

— $\texttt{Sign}(\mathsf{sk}, m)$: Pour signer le message m, cet algorithme choisit $h \xleftarrow{\$} \mathbb{G}_1^*$ et retourne la signature $\sigma \leftarrow (h, h^{x+ym})$.

— $\texttt{Verify}(\mathsf{pk}, m, \sigma)$: Pour vérifier la signature $\sigma = (\sigma_1, \sigma_2)$ sur le message m, cet algorithme s'assure que $\sigma_1 \neq 1_{\mathbb{G}_1}$ et que l'égalité

$$e(\sigma_1, \tilde{X}\tilde{Y}^m) = e(\sigma_2, \tilde{g})$$

est vraie. Si ces conditions sont remplies alors il retourne 1, et 0 sinon.

Pour randomiser une signature $\sigma = (\sigma_1, \sigma_2)$ sur un message m, il suffit de tirer un t aléatoire dans $(\mathbb{Z}/p\mathbb{Z})^\times$ et calculer $\sigma' = (\sigma_1^t, \sigma_2^t)$: c'est toujours une signature valide qui revient à choisir h^t au lieu de h dans l'algorithme de signature.

Ce schéma est sûr sous une hypothèse de sécurité appelée *Assumption* 1 [PS16, Définition 3] dans l'article original, qui est une variation de l'hypothèse LRSW [Lys+99]. Nous la désignerons dans ce qui suit par « hypothèse PS » pour reprendre l'appellation utilisée par les articles qui ont suivi.

Définition IV.2.1 (Hypothèse PS)**.** Soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ un couplage de type 3 d'ordre p premier. Soient $g \in \mathbb{G}_1$ et $\tilde{g} \in \mathbb{G}_2$ des générateurs. Pour $(\tilde{X} = \tilde{g}^x, \tilde{Y} = \tilde{g}^y)$ où x et y sont des scalaires aléatoires dans $\mathbb{Z}/p\mathbb{Z}$, on définit l'oracle $\mathcal{O}(m)$ sur l'entrée $m \in \mathbb{Z}/p\mathbb{Z}$ qui choisit un $h \in \mathbb{G}_1$ aléatoire et retourne la paire (h, h^{x+ym}). Étant donnés $(g, g^y, \tilde{g}, \tilde{X}, \tilde{Y})$ et un accès illimité à l'oracle $\mathcal{O}$, aucun adversaire ne peut efficacement générer une telle paire (h, h^{x+ym^*}), où $h \neq 1_{\mathbb{G}_1}$, et m^* un scalaire jamais demandé à l'oracle $\mathcal{O}$.

Sous l'hypothèse PS, le schéma de signature est EUF-CMA [PS15].

Les auteurs donnent différentes versions de leur schéma de signature. La version qui nous intéresse est celle supportant l'agrégation de signature, car elle permet de réduire la taille de la clé publique (certains éléments de la clé publique passent en paramètre

du système). On ne va cependant pas utiliser spécifiquement la capacité d'agrégation du schéma de signature. Les algorithmes de cette variante sont définis comme suit :

Définition IV.2.2 (Schéma de signature Pointcheval-Sanders)**.**

- $\texttt{Setup}(1^\lambda)$: Étant donné un paramètre de sécurité λ, cet algorithme retourne les paramètres publics du système $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, X, \tilde{g}, \tilde{X})$, où l'application $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ est un couplage de type 3 d'ordre p premier, $g \in \mathbb{G}_1$ et $\tilde{g} \in \mathbb{G}_2$ sont des générateurs et $\log_g(X) = \log_{\tilde{g}}(\tilde{X})$ est un scalaire aléatoire.
- $\texttt{Keygen}(pp)$: Cet algorithme choisit un $y \in \mathbb{Z}/p\mathbb{Z}$ aléatoire et définit la clé privée sk comme $Y = g^y$ et la clé publique pk comme $\tilde{Y} = \tilde{g}^y$.
- $\texttt{Sign}(\mathsf{sk}, m)$: Pour signer le message m, cet algorithme tire r aléatoirement dans $(\mathbb{Z}/p\mathbb{Z})^\times$ et retourne la signature $\sigma \leftarrow (g^r, X^r Y^{rm})$.
- $\texttt{Verify}(\mathsf{pk}, m, \sigma)$: Pour vérifier la signature $\sigma = (\sigma_1, \sigma_2)$ sur le message m, cet algorithme s'assure que $\sigma_1 \neq 1_{\mathbb{G}_1}$ et que l'égalité

$$e(\sigma_1, \tilde{X}\tilde{Y}^m) = e(\sigma_2, \tilde{g})$$

  est vraie. Si ces conditions sont remplies alors il retourne 1, et 0 sinon.

On remarque deux différences par rapport à la description précédente : la première est la présence des générateurs g et $\tilde{g}$ ainsi que les éléments X et $\tilde{X}$ dans les paramètres publics. C'est cela qui permet de réduire la taille des clés publiques. La seconde différence est la clé privée : dans la Définition IV.2.2, sk est g^y, un élément de $\mathbb{G}_1$, au lieu du scalaire y. Cela n'a pas grande importance pour les signatures PS, mais en aura dans notre schéma de signature de groupe (voir la fin de la Sous-Section IV.2.3).

Signature Fuchsbauer-Hanser-Slamanig

Dans leur article [FHS15], les auteurs Fuchsbauer, Hanser et Slamanig décrivent un schéma de signatures randomisables pour des classes d'équivalences, c'est-à-dire qu'une signature porte sur l'ensemble des représentants d'une même classe. Plus précisément, à partir d'une signature valide sur un représentant, il est possible de dériver facilement une signature valide pour un autre représentant.

Les classes d'équivalence sont définies par homothétie, c'est-à-dire que sur $\mathbb{G}_1^n$, deux n-uplets $(M_1, M_2, \ldots, M_n)$ et $(N_1, N_2, \ldots, N_n)$ sont dans la même classe d'équivalence si, et seulement si, il existe un scalaire a tel que, $M_i = N_i^a$ pour tout $1 \leqslant i \leqslant n$. Dans notre

schéma de signature de groupe, on n'utilise que $n = 2$, ainsi on se restreint à décrire ce cas-là.

Définition IV.2.3 (Schéma de signature Fuchsbauer-Hanser-Slamanig)**.**

- $\texttt{Setup}(1^\lambda)$: Étant donné un paramètre de sécurité λ, cet algorithme retourne les paramètres publics du système $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$, où le couplage $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ est de type 3, d'ordre p premier et $g \in \mathbb{G}_1$ et $\tilde{g} \in \mathbb{G}_2$ sont des générateurs.
- $\texttt{Keygen}(pp)$: Cet algorithme tire deux scalaires α_1 et α_2 aléatoirement, calcule $\tilde{A}_1 = \tilde{g}^{\alpha_1}$, $\tilde{A}_2 = \tilde{g}^{\alpha_2}$ et définit la clé privée sk comme (α_1, α_2) et la clé publique pk comme $(\tilde{A}_1, \tilde{A}_2)$.
- $\texttt{Sign}(\mathsf{sk}, (M_1, M_2))$: Pour signer le représentant (M_1, M_2), cet algorithme tire t aléatoirement dans $(\mathbb{Z}/p\mathbb{Z})^\times$ et retourne la signature

$$(\tau_1, \tau_2, \tilde{\tau}) \leftarrow ((M_1^{\alpha_1} M_2^{\alpha_2})^t, g^{1/t}, \tilde{g}^{1/t}).$$

- $\texttt{Verify}(\mathsf{pk}, (M_1, M_2), \tau)$: Pour vérifier la signature $\tau = (\tau_1, \tau_2, \tilde{\tau})$ sur le représentant (M_1, M_2), cet algorithme s'assure que les égalités

$$e(\tau_1, \tilde{\tau}) = e(M_1, \tilde{A}_1) e(M_2, \tilde{A}_2) \quad \text{et} \quad e(\tau_2, \tilde{g}) = e(g, \tilde{\tau})$$

  sont vraies. Si ces conditions sont remplies alors il retourne 1, et 0 sinon.

Pour randomiser une signature $\tau = (\tau_1, \tau_2, \tilde{\tau})$ sur un autre représentant (M_1^s, M_2^s), pour un scalaire s, il suffit de tirer un r aléatoire dans $(\mathbb{Z}/p\mathbb{Z})^\times$ et calculer $(\tau_1^{sr}, \tau_2^{1/r}, \tilde{\tau}^{1/r})$.

IV.2.2 Hypothèses calculatoires

Le Théorème IV.3.1 énonce les conditions pour que notre signature de groupe soit sûre, *i.e.* ait les propriétés d'anonymat, de traçabilité et de non-diffamation. Deux hypothèses calculatoires sont requises, une classique, SXDH, et une autre taillée sur mesure MPS, modifiée de l'hypothèse PS en Définition IV.2.1.

Hypothèse SXDH

Pour $i \in \{1, 2\}$, le problème DDH est difficile dans $\mathbb{G}_i$ si les distributions (g, g^x, g^y, g^z) et (g, g^x, g^y, g^{xy}) sont indistinguables pour $g \in \mathbb{G}_i$ et x, y et z aléatoires.

L'hypothèse SXDH est dite vraie si le problème DDH est difficile dans $\mathbb{G}_1$ et dans $\mathbb{G}_2$.

Hypothèse MPS

Nous aimerions que la sécurité de notre signature de groupe repose directement sur l'hypothèse PS, ce n'est malheureusement pas possible. Nous introduisons donc une nouvelle hypothèse nommée MPS (le M signifie *modified* en anglais).

Définition IV.2.4 (Hypothèse MPS)**.** Soit $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ un couplage de type 3 d'ordre p premier. Soient $g \in \mathbb{G}_1$ et $\tilde{g} \in \mathbb{G}_2$ des générateurs. Soit $h : \{0,1\}^* \to \mathbb{Z}/p\mathbb{Z}$ une fonction, où $\{0,1\}^*$ est l'ensemble des chaînes binaires de longueur finie. Pour des scalaires aléatoires x, y et z, on définit l'oracle $\mathcal{O}(m)$ sur l'entrée $m \in \mathbb{Z}/p\mathbb{Z}$ qui tire aléatoirement $r, s \in \mathbb{Z}/p\mathbb{Z}$ et renvoie le tuple $P = (s, \tilde{g}^{rs}, g^r, g^{rz}, g^{r(x+ty)})$ où $t = h(\tilde{g}^{rs}||g^r||m)$.

L'hypothèse MPS est alors qu'étant donnés $(g, g^y, g^z, g^{zx}, \tilde{g}, \tilde{X}, \tilde{Y})$ pour un scalaire aléatoire z, et un accès illimité à l'oracle $\mathcal{O}$, aucun adversaire ne peut efficacement générer $(t^*, g^r, g^{r(x+t^*y)})$ avec $r \neq 0$ et une valeur de t^* différente de toutes celles faisant parties des réponses de $\mathcal{O}$.

La validité de la réponse de l'adversaire peut facilement être vérifiée grâce au couplage : s'il a une réponse correcte alors l'égalité suivante est vraie,

$$e\left(g^{r(x+t^*y)}, \tilde{g}\right) = e\left(g^r, \tilde{X} \cdot \tilde{Y}^{t^*}\right).$$

On remarque que le but de l'adversaire est toujours de produire une signature PS valide, sauf qu'il a maintenant accès à des éléments additionnels qui ne semblent pas l'aider à créer de contrefaçons. En effet :

— Notre nouvel oracle renvoie toujours une signature PS $(g^r, g^{r(x+ty)})$ mais sur un scalaire $t = h(\tilde{g}^{rs}||g^r||m)$ au lieu de m. Cependant, on définit des conditions de réussite bien plus difficiles pour l'adversaire : il ne peut que réussir si le scalaire t de sa contrefaçon est différent de ceux de $\mathcal{O}$ (en particulier une contrefaçon sur un nouveau message m^* n'est pas valide si cela mène à un t déjà utilisé). Intuitivement, cela exclut toute stratégie ayant recours aux propriétés de h (comme les collisions). D'un point de vue de la sécurité, cela ne change donc rien comparé à l'hypothèse où $\mathcal{O}$ retournerait $(g^r, g^{r(x+my)})$. On introduit cette modification car on a besoin d'une façon d'empêcher de possibles re-randomisations des signatures renvoyées par $\mathcal{O}$.

— Cette légère modification en induit une autre : il nous faut maintenant fournir la paire (g^z, g^{zx}), pour un scalaire z aléatoire, dans l'hypothèse MPS. Dans [PS16],

cette paire est exactement une signature de 0 et est donc directement générée par la réduction dans la preuve de sécurité en appelant $\mathcal{O}$ sur 0. Ce n'est plus possible ici, d'où la nécessité d'ajouter explicitement ces éléments dans la définition de l'hypothèse. Dans tous les cas, cela ne donne pas plus de pouvoir à l'adversaire qu'il n'en a dans l'hypothèse PS.

— L'élément g^{rz} est le seul impliquant le secret z dans P. Cela signifie alors qu'il est inutile de le combiner avec d'autres éléments de P pour obtenir un nouveau n-uplet valide.

— La dernière différence avec l'hypothèse PS est la paire $(s, \tilde{g}^{rs})$ qui doit être ajoutée aux réponses de l'oracle. On remarque cependant que $\tilde{g}^{rs}$ est un élément de $\mathbb{G}_2$ et est ainsi intuitivement inutile pour falsifier une signature PS $(g^r, g^{r(x+ty)}) \in \mathbb{G}_1^2$, grâce à l'asymétrie du couplage. Le même raisonnement est vrai pour s qui n'est pas une des valeurs secrètes utilisées pour calculer la signature PS.

Remarque IV.2.1. On remarque que la difficulté du problème calculatoire sous-jacent à l'hypothèse MPS dépend de la fonction h. Par exemple, si h est constante alors aucun adversaire ne peut réussir dès qu'il fait (au moins) une requête à $\mathcal{O}$. Cependant, nous supposons ici (et prouvons dans le modèle du groupe générique) que l'hypothèse MPS est vraie pour toute fonction $h : \{0,1\}^* \to \mathbb{Z}/p\mathbb{Z}$.

Évaluation de l'hypothèse MPS

Le but d'un adversaire $\mathcal{A}$, contre l'hypothèse MPS, est de contrefaire une nouvelle signature PS, avec l'aide d'un oracle $\mathcal{O}$ qui renvoie plus d'éléments que dans l'hypothèse originale. Comme discuté plus haut, ces nouveaux éléments ne semblent pas donner d'avantage à $\mathcal{A}$. Nous formalisons cela avec une évaluation de l'hypothèse MPS dans le modèle du groupe générique [Sho97].

Proposition IV.2.1. *Dans le modèle du groupe générique, aucun adversaire ne peut réussir contre l'hypothèse* MPS *avec une probabilité plus grande que* $O((4q_{\mathcal{O}} + 7 + q_G)^2/p)$, *où* q_G *est une borne sur le nombre de requêtes faites à l'oracle de groupe et* $q_{\mathcal{O}}$ *est une borne sur le nombre de requêtes faites à l'oracle* $\mathcal{O}$.

Démonstration. L'adversaire a accès à $(g, g^y, g^z, g^{zx}, \tilde{g}, \tilde{X}, \tilde{Y})$ ainsi qu'à ce que l'oracle $\mathcal{O}$ renvoie $(s_i, \tilde{g}^{r_i s_i}, g^{r_i}, g^{r_i z_i}, g^{r_i(x_i + t_i y_i)})$, avec $t_i = h(\tilde{g}^{r_i s_i} || g^{r_i} || m_i)$. On va d'abord prouver que $\mathcal{A}$ ne peut pas symboliquement produire un uplet valide $(t^*, \sigma_1, \sigma_2) \in \mathbb{Z}/p\mathbb{Z} \times \mathbb{G}_1^2$ en

associant chaque élément de groupe avec un polynôme dont les variables formelles sont x, y, z et r_i, pour $i \in [1, q_{\mathcal{O}}]$.

Dans le modèle du groupe générique, tout élément de groupe doit avoir été généré via des requêtes à l'oracle de la loi interne du groupe ou bien à l'oracle $\mathcal{O}$. Vu que l'objectif de $\mathcal{A}$ est de produire des éléments σ_1 et σ_2 de $\mathbb{G}_1$, ils ne peuvent être que combinaisons d'éléments de ce même groupe. Cela signifie qu'il y a des coefficients connus $a, b, c, d, \alpha_i, \beta_i, \gamma_i$ et $a', b', c', d', \alpha'_i, \beta'_i, \gamma'_i$, pour $i \in [1, q_{\mathcal{O}}]$, tels que :

$$\sigma_1 = g^a \cdot (g^y)^b \cdot (g^z)^c \cdot (g^{z \cdot x})^d \cdot \prod_i (g^{r_i})^{\alpha_i} \cdot (g^{r_i \cdot z_i})^{\beta_i} \cdot (g^{r_i(x + t_i \cdot y)})^{\gamma_i};$$
$$\sigma_2 = g^{a'} \cdot (g^y)^{b'} \cdot (g^z)^{c'} \cdot (g^{z \cdot x})^{d'} \cdot \prod_i (g^{r_i})^{\alpha'_i} \cdot (g^{r_i \cdot z_i})^{\beta'_i} \cdot (g^{r_i(x + t_i \cdot y)})^{\gamma'_i}.$$

Un uplet $(t^*, \sigma_1, \sigma_2)$ est valide seulement si $e(\sigma_1, \tilde{X}\tilde{Y}^{t^*}) = e(\sigma_2, \tilde{g})$, $\sigma_1 \neq 1_{\mathbb{G}_1}$ et $t^* \neq t_i$ pour tout $i \in [1, q_{\mathcal{O}}]$. Cela donne la relation suivante :

$$\left(a + by + cz + dzx + \sum_i \alpha_i r_i + \beta_i r_i z + \gamma_i r_i (x + y t_i)\right)(x + y t^*)$$
$$= \left(a' + b'y + c'z + d'zx + \sum_i \alpha'_i r_i + \beta'_i r_i z + \gamma'_i r_i (x + y t_i)\right).$$

Le facteur $(x + yt^*)$ implique que tout monôme du membre de gauche est de degré au moins 1 en x ou y. Nous pouvons donc conclure que $a' = c' = \alpha'_i = \beta'_i = 0$, pour tout $i \in [1, q_{\mathcal{O}}]$, ce qui mène à l'équation suivante :

$$\left(a + by + cz + dzx + \sum_i \alpha_i r_i + \beta_i r_i z + \gamma_i r_i (x + y t_i)\right)(x + y t^*)$$
$$= \left(b'y + d'zx + \sum_i \gamma'_i r_i (x + y t_i)\right).$$

Le membre de droite ne contient aucun monôme de degré 2 en x ou y, ce qui implique que $b = d = \gamma_i = 0$, pour tout $i \in [1, q_{\mathcal{O}}]$, et :

$$\left(a + cz + \sum_i \alpha_i r_i + \beta_i r_i z\right)(x + y t^*) = \left(b'y + d'zx + \sum_i \gamma'_i r_i (x + y t_i)\right).$$

Si nous développons le membre de gauche, on obtient des monômes tels que ax, czy et $\beta_i r_i zx$. Cependant, il n'y a aucun terme similaire dans le membre de droite ce que implique

$a = c = \beta_i = 0$, quel que soit i, et :

$$\left(\sum_i \alpha_i r_i\right)(x + yt^*) = \left(b'y + d'zx + \sum_i \gamma'_i r_i(x + yt_i)\right).$$

Les termes du membre de gauche sont de degré 1 en r_i, ce qui implique $b' = d' = 0$, et ainsi :

$$\left(\sum_i \alpha_i r_i\right)(x + yt^*) = \left(\sum_i \gamma'_i r_i(x + yt_i)\right).$$

Nous obtenons donc, pour chaque $i \in [1, q_{\mathcal{O}}]$,

$$\alpha_i(x + yt^*) = \gamma'_i(x + yt_i),$$

ce qui implique $\alpha_i = \gamma'_i$. Toutefois, on sait aussi que $t^* \neq t_i$ pour tout $i \in [1, q_{\mathcal{O}}]$, dont il résulte $\alpha_i = \gamma'_i = 0$. $\mathcal{A}$ retourne alors un tuple $(t^*, 1_{\mathbb{G}_1}, 1_{\mathbb{G}_1})$, qui est invalide.

Maintenant, évaluons la probabilité d'une validité survenant accidentellement, *i.e.* quand deux polynômes différents renvoyés par l'oracle de groupe s'évaluent en la même valeur. Tous les polynômes considérés dans cette preuve sont au plus de degré 2. Vu qu'il y a au plus $4q_{\mathcal{O}}+7+q_G$ polynômes, il y a au plus $(4q_{\mathcal{O}}+7+q_G)^2/2$ paires qui peuvent donner la même valeur lorsqu'ils sont évalués. Par le lemme Schwartz-Zippel [Zip79 ; Sch80], cela ne peut se produire avec une probabilité supérieure à $(4q_{\mathcal{O}} + 7 + q_G)^2/p$, ce qui est négligeable. □

IV.2.3 Intuition de la construction

Une signature de groupe est généralement constituée de deux types de signature numérique, que nous notons σ et τ. La première, σ, est produite sur le message à signer par l'utilisateur à l'aide de sa paire de clés personnelle $(\mathsf{usk}, \mathsf{upk})$. Intuitivement, l'impossibilité de contrefaire une signature numérique assure qu'aucun adversaire n'est capable de produire une fausse signature de groupe qui pourrait être rattachée à upk (cela correspond à la propriété de non-diffamation). La seconde, τ, est produite par le gestionnaire de groupe sur usk (ou upk) pour différencier les paires de clés des membres du groupe de celles d'utilisateurs non-enregistrés. Ici, la non-contrefaçon permet aux seuls membres de produire des signatures de groupe valides, ce qui est nécessaire pour assurer la traçabilité.

Si la non-diffamation et la traçabilité étaient les deux seules conditions attendues d'une

signature de groupe, alors une telle signature pourrait simplement être $(\tau, \sigma, \mathsf{upk}, m)$. Cela ne peut cependant pas marcher dès lors que l'anonymat est souhaité, aussi la pratique courante est d'au moins chiffrer ou s'engager sur les valeurs τ et upk puis de fournir une preuve sans-divulgation de connaissance que ces éléments sont bien formés.

Les travaux de Bichsel *et al.* [Bic+10] ont montré qu'il est possible de faire mieux quand τ est randomisable. En effet, dans ce cas il est inutile de chiffrer τ, il suffit simplement de le re-randomiser et de l'envoyer sans chiffrement, ce qui améliore grandement l'efficacité du schéma. Leur signature de groupe peut seulement réaliser une plus faible version de l'anonymat, l'anonymat désintéressé, dans le modèle de l'oracle aléatoire (ROM), mais cela semble un prix raisonnable au vu des bénéfices.

Malgré sa nouveauté, le schéma de Bichsel *et al.* [Bic+10] partage encore des similarités avec le modèle BSZ [BSZ05]. Il y a en effet la composition modulaire de deux signatures τ et σ, avec une preuve de connaissance. La signature σ et la preuve de connaissance peuvent être combinées (donnant alors une signature de connaissance) en utilisant l'heuristique de Fiat et Shamir [FS87] dans le ROM, mais l'esprit général reste le même. Les constructions modulaires sont intéressantes car elles peuvent tirer profit des avancées dans l'instanciation de chacun de leurs blocs de base. Par exemple, le schéma de Bichsel *et al.* [Bic+10] peut directement être amélioré en utilisant les signatures PS [PS16] pour instancier τ au lieu des signatures Camenisch-Lysyanskaya [CL03] de la construction originelle. Malheureusement, la complexité d'une construction modulaire est la somme de toutes ses parties, ce qui conduit naturellement à se poser la question suivante : est-il possible d'améliorer l'efficacité du schéma en optimisant la combinaison des différents blocs de base, pour des instanciations spécifiques ?

Dans la suite, nous construisons la signature de groupe la plus efficace sans oracle aléatoire en remarquant que les signatures FHS [FHS15] sur classes d'équivalence interagissent élégamment avec les signatures PS [PS16]. Rappelons succinctement ces dernières, données en Définition IV.2.2 (pour rappel, nous choisissons la version permettant l'agrégation des signatures mais sans utiliser cette propriété spécifiquement). Une signature sur un message m est donnée par $(\sigma_1, \sigma_2) = (g^r, X^r(g^{ym})^r)$ où r est un scalaire aléatoire, $X = g^x$ est un paramètre public et y est la clé privée du signataire. Remarquons qu'il est possible de définir σ_2 comme $\sigma_2^{1/m} = X^{r/m} g^{yr}$: un adversaire capable de falsifier une telle signature peut être transformé en un adversaire contre le schéma originel de signature PS. En conséquence, toute signature produite par un utilisateur sera de la forme $(\sigma_1, \sigma_2) = (g^r, X^{r/m} g^{yr})$.

Si on appliquait la méthodologie classique ici, nous devrions fournir une signature τ sur y (ou g^y) et ensuite prouver sans divulgation de connaissance que τ certifie bien la clé qui a été utilisée pour générer (σ_1, σ_2). Il est cependant possible de faire mieux en faisant directement usage du schéma de signature FHS [FHS15], donné par la Définition IV.2.3.

En effet, pour tout r, si nous ignorons le terme $X^{r/m}$ dans σ_2, il ne reste plus que (g^r, g^{yr}), qui sont les différents représentants d'une même classe d'équivalence. Donc, si nous signons (g^r, g^{ry}) avec le schéma FHS, il est possible de directement vérifier que (σ_1, σ_2) a été générée en utilisant une clé certifiée, sans besoin de preuve de connaissance. L'anonymat de la construction résultante découle simplement de la capacité à re-randomiser la signature FHS tout en changeant le représentant de la classe.

Il nous reste alors à expliquer comment retirer $X^{r/m}$. Rappelons qu'une signature FHS sur (g^r, g^{ry}) est un triplet $(\tau_1, \tau_2, \tilde{\tau})$ tel que

$$e(\tau_1, \tilde{\tau}) = e(g^r, \tilde{A}_1)e(g^{ry}, \tilde{A}_2) \quad \text{et} \quad e(\tau_2, \tilde{g}) = e(g, \tilde{\tau}),$$

où $(\tilde{A}_1, \tilde{A}_2) = (\tilde{g}^{\alpha_1}, \tilde{g}^{\alpha_2})$ est la clé publique. Supposons que nous ajoutons $\tilde{B} = \tilde{X}^{\alpha_2}$ à cette clé publique ($\tilde{X} = \tilde{g}^x$ fait partie de la clé publique du schéma de signature PS). Alors

$$e(\sigma_1, \tilde{A}_1)e(\sigma_2, \tilde{A}_2)e(\sigma_1, \tilde{B}^{-1/m}) = e(g^r, \tilde{A}_1)e(g^{yr}, \tilde{A}_2) = e(\tau_1, \tilde{\tau}),$$

et la seconde équation reste inchangée. Cela signifie que nous pouvons vérifier la validité des signatures FHS et PS en même temps avec essentiellement le même coût de vérification qu'une signature FHS seule. De plus, le fait de fusionner la vérification de ces deux signatures rend inutiles les preuves sans divulgation de connaissance. Concrètement, cela veut dire que notre signature de groupe consiste en seulement $(\sigma_1, \sigma_2, \tau_1, \tau_2, \tilde{\tau})$, *i.e.* quatre éléments de $\mathbb{G}_1$ et un de $\mathbb{G}_2$, et qu'elle peut être vérifiée avec deux équations de couplage.

Curieusement, le fait d'éviter la signature de connaissance classiquement utilisée sur y, permet de réaliser à la fois l'anonymat CPA et l'anonymat désintéressé. En effet, les schémas utilisant des signatures randomisables (comme les signatures CL [CL03] ou PS [PS16]) sont généralement prouvés anonymes sous l'hypothèse DDH dans $\mathbb{G}_1$. Ainsi, pour permettre l'ouverture des signatures, ces schémas forcent les utilisateurs à fournir une sorte de « trappe » $\tilde{g}^y \in \mathbb{G}_2$, donnant la capacité à l'autorité d'ouverture de casser DDH sur leurs signatures en particulier. Quand y fait partie de la clé privée usk de l'utilisateur (ce qui est nécessairement le cas pour la signature de connaissance sur y), la divulgation de celle-ci signifie que l'adversaire peut recouvrer y et donc $\tilde{g}^y$. Dans ce cas, seule la notion

d'anonymat désintéressé peut être attendue du schéma de signature de groupe.

Dans notre construction, nous remarquons que $g^y \in \mathbb{G}_1$ est suffisant pour produire des signatures de groupe, c'est-à-dire que les utilisateurs peuvent « oublier » le scalaire y après avoir généré leurs clés. Si usk venait à fuiter, l'adversaire n'aurait que g^y, qui est inutile pour casser DDH. Dans ce cas, on peut alors garder un certain niveau d'anonymat (à savoir l'anonymat CPA).

IV.2.4 Construction de la signature de groupe

Nous donnons enfin les algorithmes qui constituent notre signature de groupe.

— $\mathtt{Setup}(1^\lambda)$: Soient $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ un couplage de type 3 d'ordre premier p, cet algorithme commence par sélectionner $g \xleftarrow{\$} \mathbb{G}_1$ et $\tilde{g} \xleftarrow{\$} \mathbb{G}_2$, puis calcule $(X, \tilde{X}) \leftarrow (g^x, \tilde{g}^x)$ pour un scalaire aléatoire x. Il génère aussi les paramètres publics pp_Σ d'un schéma de signature numérique Σ et choisit une fonction de hachage $h : \{0,1\}^* \to \mathbb{Z}/p\mathbb{Z}$. Enfin, il génère une chaîne de référence commune $\mathtt{crs}$ pour le système de preuve Groth-Sahai [GS08] dans la configuration SXDH et retourne les paramètres publics du système :

$$pp \leftarrow (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g}, X, \tilde{X}, \mathtt{crs}, pp_\Sigma, h).$$

— $\mathtt{UKeygen}(pp)$: L'utilisateur définit ses clés personnelles $(\mathsf{sk}, \mathsf{pk}) \leftarrow \Sigma.\mathtt{Keygen}(pp_\Sigma)$.

— $\mathtt{OKeygen}(pp)$: L'autorité d'ouverture génère sa paire de clés $(\mathsf{osk}, \mathsf{opk})$ pour un schéma de chiffrement à clé publique Γ.

— $\mathtt{GKeygen}(pp)$: Le gestionnaire du groupe tire aléatoirement deux scalaires α_1 et α_2 puis calcule $(\tilde{A}_1, \tilde{A}_2, \tilde{B}) \leftarrow (\tilde{g}^{\alpha_1}, \tilde{g}^{\alpha_2}, \tilde{X}^{\alpha_2})$. Puis le gestionnaire initialise un registre public $\mathbf{Reg}$ et assigne ses clés $(\mathsf{gsk}, \mathsf{gpk}) \leftarrow ((\alpha_1, \alpha_2), (\tilde{A}_1, \tilde{A}_2, \tilde{B}))$.

— $\mathtt{Join}$: Pour rejoindre le groupe, un utilisateur i choisit d'abord deux scalaires aléatoires, u et y, et calcule (g^u, g^{uy}) ainsi que $C \leftarrow \Gamma.\mathtt{Encrypt}(\mathsf{opk}, \tilde{g}^y)$. Ensuite, il génère une preuve NIZK π assurant que C chiffre un élément $\tilde{g}^y \in \mathbb{G}_2$ tel que $e(g^u, \tilde{g}^y) = e(g^{uy}, \tilde{g})$. Enfin, il génère $\mu \leftarrow \Sigma.\mathtt{Sign}(\mathsf{sk}_i, (g^u || g^{uy} || C || \pi))$ et l'envoie, avec (g^u, g^{uy}, C, π), au gestionnaire.

En recevant ces éléments, le gestionnaire du groupe vérifie la validité de la preuve π et que $\Sigma.\mathtt{Verify}(\mathsf{pk}_i, \mu, (g^u || g^{uy} || C || \pi)) = 1$. Si π et μ sont toutes les deux valides, alors le gestionnaire stocke $(g^u, g^{uy}, C, \pi, \mathsf{pk}_i, \mu)$ dans $\mathbf{Reg}[i]$, génère un scalaire

$t \xleftarrow{\$} \mathbb{Z}/p\mathbb{Z}$ et retourne

$$(\tau_1', \tau_2, \tilde{\tau}) \leftarrow (((g^u)^{\alpha_1}(g^{uy})^{\alpha_2})^t, g^{1/t}, \tilde{g}^{1/t}).$$

Enfin, l'utilisateur calcule $\tau_1 \leftarrow (\tau_1')^{1/u}$ et définit $\mathsf{usk}_i = (\tau_1, \tau_2, \tilde{\tau}, g^y)$.

— $\mathtt{Sign}(\mathsf{usk}_i, m)$: Pour signer un message m, l'utilisateur tire d'abord deux scalaires r et s aléatoirement, puis génère les éléments suivants :

$$\begin{aligned} \tau_1' &\leftarrow \tau_1^{rs}, \\ (\tau_2', \tilde{\tau}') &\leftarrow (\tau_2^{1/s}, \tilde{\tau}^{1/s}), \\ (\sigma_1, \sigma_2) &\leftarrow (g^r, X^{r/h(\tilde{\tau}||\sigma_1||m)}(g^y)^r). \end{aligned}$$

La signature de groupe σ sur m est alors définie comme $\sigma = (\tau_1', \tau_2', \tilde{\tau}', \sigma_1, \sigma_2)$.

— $\mathtt{Verify}(\mathsf{gpk}, \sigma, m)$: Pour vérifier la validité de la signature σ sur m, on commence par s'assurer qu'aucun de ses éléments n'est $1_{\mathbb{G}_1}$ ou $1_{\mathbb{G}_2}$ et que les égalités suivantes sont vraies :

$$e(\sigma_1, \tilde{A}_1 \tilde{B}^{-1/h(\tilde{\tau}||\sigma_1||m)}) e(\sigma_2, \tilde{A}_2) = e(\tau_1, \tilde{\tau}) \quad \text{et} \quad e(\tau_2, \tilde{g}) = e(g, \tilde{\tau}),$$

dans ce cas, la signature est acceptée (l'algorithme retourne 1). Sinon, elle est rejetée (l'algorithme retourne 0).

— $\mathtt{Open}(\mathsf{osk}, \mathsf{gpk}, \sigma, m)$: Avant d'ouvrir une signature, l'autorité d'ouverture vérifie que celle-ci est valide. Sinon, elle retourne 0. En utilisant sa clé privée osk, l'autorité d'ouverture peut déchiffrer n'importe lequel des C_i stockés dans $\mathbf{Reg}[i]$ et ainsi recouvrer les éléments $\tilde{g}^{y_i} \in \mathbb{G}_2$ quel que soit l'utilisateur préalablement enregistré par le gestionnaire de groupe. L'autorité d'ouverture peut alors vérifier, pour chacun des $\tilde{g}^{y_i}$, si l'égalité suivante est vraie :

$$e(\sigma_2, \tilde{\tau}) e(\sigma_1, \tilde{X}^{-1/h(\tilde{\tau}||\sigma_1||m)}) = e(\sigma_1, \tilde{g}^{y_i}).$$

S'il n'y a pas de correspondance, alors l'autorité d'ouverture renvoie $\perp$. Sinon, notons j l'utilisateur correspondant. L'autorité d'ouverture recouvre les données de l'utilisateur $(g^{u_j}, g^{u_j y_j}, C_j, \pi_j, \mathsf{pk}_j, \mu_j)$ stockées dans $\mathbf{Reg}[j]$, s'engage sur la valeur

$\tilde{g}^{y_j}$, puis retourne j accompagné d'une preuve Groth-Sahai $\overline{\pi}$ que :

$$e(\sigma_2, \tilde{\tau}) e(\sigma_1, \tilde{X}^{-1/h(\tilde{\tau}||\sigma_1||m)}) = e(\sigma_1, \tilde{g}^{y_j}) \quad \text{et} \quad e(g^{u_j y_j}, \tilde{\tau}) = e(g^{u_j}, \tilde{g}^{y_j}).$$

— $\texttt{Judge}(\mathsf{gpk}, \sigma, m, i, \overline{\pi})$: Pour vérifier une ouverture, on vérifie que $\overline{\pi}$ est valide, $\texttt{Verify}(\mathsf{gpk}, \sigma, m) = 1$ et $\Sigma.\texttt{Verify}(\mathsf{pk}_i, \mu_i, (g^{u_i}||g^{u_i y_i}||C_i||\pi_i)) = 1$. Si toutes les conditions sont satisfaites, alors on retourne 1. Sinon, on retourne 0.

Correction du schéma

Remarquons qu'à la fin du protocole `Join`, l'utilisateur reçoit une signature FHS sur la classe d'équivalence contenant le représentant (g, g^y). Car en effet,

$$(\tau_1, \tau_2, \tilde{\tau}) = ((g^{\alpha_1} g^{y\alpha_2})^t, g^{1/t}, \tilde{g}^{1/t}).$$

Pour produire une signature de groupe sur un message m, l'utilisateur commence par re-randomiser $(\tau_1, \tau_2, \tilde{\tau})$ en utilisant l'aléa s, tout en changeant de représentant dans la classe grâce à l'aléa r, *i.e.* (g^r, g^{yr}). Le triplet $(\tau_1', \tau_2', \tilde{\tau}')$ qui en résulte est toujours une signature FHS sur la même classe d'équivalence. Il génère ensuite une paire (σ_1, σ_2) telle que $(\sigma_1, \sigma_2^{m'})$ soit une signature PS sur $m' = h(\tilde{\tau}'||\sigma_1||m)$ en utilisant le même aléas r. Ainsi, une telle signature de groupe satisfait :

$$\begin{aligned} e\left(\sigma_1, \tilde{A}_1 \tilde{B}^{-1/m'}\right) e(\sigma_2, \tilde{A}_2) &= e\left(g^r, \tilde{g}^{\alpha_1 - x\alpha_2/m'}\right) e\left(g^{r(x/m'+y)}, \tilde{g}^{\alpha_2}\right), \\ &= e(g, \tilde{g})^{r(\alpha_1 + y\alpha_2)}, \\ &= e(\tau_1', \tilde{\tau}'), \end{aligned}$$

et $e(\tau_2', \tilde{g}) = e(g^{1/(ts)}, \tilde{g}) = e(g, \tilde{g}^{1/(ts)}) = e(g, \tilde{\tau}')$.

Remarque IV.2.2. Les signatures de groupe suivant le modèle de construction SEP ont généralement une procédure d'ouverture efficace. En effet, l'autorité d'ouverture connaît le clé de déchiffrement correspondante et peut donc déchiffrer le message inclus dans la signature de groupe et identifier le signataire. Il n'y a malheureusement pas d'équivalent pour les constructions sans chiffrement et en particulier, il n'y a plus de clé « maîtresse » que l'autorité d'ouverture peut utiliser pour lever l'anonymat.

Les schémas utilisant les signatures randomisables [Bic+10 ; DS18] surmontent ce problème en forçant chaque utilisateur à fournir à l'autorité d'ouverture un moyen lui per-

mettant d'ouvrir leurs signatures. Concrètement, durant l'exécution du protocole `Join`, chaque utilisateur doit transmettre des éléments dépendant de sa clé secrète à cette autorité. Malheureusement, ce prérequis ne correspond pas au modèle BSZ [BSZ05] où `Join` est un protocole à deux parties entre l'utilisateur et le gestionnaire de groupe. Il y a alors deux façons de régler ce problème. Soit l'autorité d'ouverture prend part au protocole `Join`, soit on exige que l'utilisateur envoie ces éléments identifiants au gestionnaire. La première solution est conceptuellement la plus simple mais modifie le modèle original BSZ, ce qui n'est pas le cas de la seconde approche, bien qu'elle demande des primitives supplémentaires pour assurer la sécurité du schéma. En effet, il n'est pas possible de transmettre de tels éléments en clair (sinon le gestionnaire pourrait lever l'anonymat), il faut donc les chiffrer et prouver que (sans divulguer d'information) le chiffré est bien conforme.

On a choisi ici de décrire la solution la plus complexe (la seconde), vu qu'il est aisé de dériver un schéma de signature de groupe correspondant à la première option (celle où l'autorité d'ouverture prend part à l'enregistrement des utilisateurs). On a donc besoin d'un schéma de chiffrement à clé publique IND-CCA2 compatible avec les preuves NIKZ. En pratique, il est possible de choisir n'importe laquelle des instanciations de [Lib+14] qui interagissent bien avec les preuves Groth-Sahai [GS08]. Remarquons que l'efficacité n'est pas vraiment une préoccupation ici car l'étape d'enregistrement avec le protocole `Join` n'a pas d'impact sur la génération des signatures de groupe elles-mêmes.

Remarque IV.2.3. On note que le modèle de sécurité de Bellare *et al* [BSZ05] suppose déjà que la phase de mise en place (`Setup`) est fiable, donc notre construction est parfaitement ajustée à ce modèle sur ce point. Malgré tout, cela n'explique pas comment générer des paramètres publics dans des conditions de la vie réelle. En pratique, il serait naturel de confier à l'autorité d'ouverture leur génération. Concernant la sécurité, ce serait problématique seulement pour la non-diffamation si cette entité est corrompue *avant* le `Setup`, mais cela est exclu par le modèle [BSZ05]. Nous pouvons aussi minimiser les risques en utilisant une génération de paramètres coopérative, comme dans [Can+15].

IV.2.5 Comparaison avec d'autres schémas

Nous comparons notre algorithme de signature avec celui d'autres constructions de l'état de l'art. Elles sont toutes prouvées sûres sous des hypothèses interactives.

Dans la Figure IV.1, nous comptons le nombre d'opérations coûteuses, *i.e.* les exponentiations dans $\mathbb{G}_1$, $\mathbb{G}_2$ et $\mathbb{G}_T$ (dénotées par e_1, e_2 et e_T respectivement). Concernant le coût

de la signature, notre schéma offre la meilleure performance lorsque calculer $3\mathsf{e}_1 + 1\mathsf{e}_2$ est moins coûteux que calculer $1\mathsf{e}_T$. Nous résumons notre comparaison dans la Figure IV.1, où le modèle BMW est celui de [BMW03] tandis que le modèle BSZ est celui de [BSZ05]. La dernière ligne du tableau correspond à notre construction où l'autorité d'ouverture et le gestionnaire de groupe sont fusionnés, ce qui a un impact sur la sécurité mais pas sur les performances.

Schéma	Taille en bits	Coût en exp. de groupe	ROM ?	Modèle GS	Anonymat
BCNSW [Bic+10]	1664	3 e_1 + 1 e_T	oui	BMW	désintéressé
PS [PS16]	1280	2 e_1 + 1 e_T	oui	BMW	désintéressé
DS [DS18]	2816	5 e_1 + 1 e_2	oui	BSZ	CPA
DS* [DS18]	4608	5 e_1 + 6 e_2	oui	BSZ	total
BHKS [Bac+18]	4992	9 e_1 + 2 e_2	non	BMW	total
Notre schéma	2304	5 e_1 + 1 e_2	non	BSZ	CPA et désintéressé
Notre schéma*	2304	5 e_1 + 1 e_2	non	BMW	total

TABLE IV.1 – Comparaison de sécurité et de performances. La *taille en bits* correspond à la taille d'une signature lorsque le schéma est instancié avec une courbe BLS, de degré de plongement 12, sur un corps de 384 bits [CS20]

Si l'on se concentre sur les constructions sans oracle aléatoire, notre signature de groupe est plus performante que la récente construction de [Bac+18] : elle réduit de moitié la taille de la signature et son coût. On remarque aussi qu'elle est compétitive comparée à la construction [PS16] dans le modèle de l'oracle aléatoire. En effet, bien que la taille des signatures soit plus grande (deux fois plus grande lorsque instanciée avec une courbe BLS sur un corps de 384 bits), le coût est similaire et, plus important, le signataire n'a plus à effectuer des opérations dans $\mathbb{G}_T$, et donc, il n'a pas à implémenter l'arithmétique de $\mathbb{F}_{p^k}$, ce qui est remarquable.

IV.3 Analyse de la sécurité

Théorème IV.3.1. *La signature de groupe décrite en Sous-Section IV.2.4 assure :*

— *la traçabilité sous la sécurité EUF-CMA du schéma de signature FHS ;*

— *la non-diffamation sous l'hypothèse* MPS*, la résistance aux collisions de la fonction de hachage* h *et la sécurité EUF-CMA de la signature numérique* Σ *;*
— *l'anonymat CPA sous l'hypothèse* SXDH *et la sécurité IND-CCA2 du schéma de chiffrement* Γ *;*
— *l'anonymat désintéressé si elle assure la non-diffamation, si* Γ *est IND-CCA2 sûr et si l'hypothèse* SXDH *est vérifiée ;*
— *l'anonymat total, lorsque le gestionnaire du groupe est aussi l'autorité d'ouverture, si elle assure la traçabilité et si l'hypothèse* SXDH *est vérifiée.*

Remarque IV.3.1. Le théorème IV.3.1 montre que notre schéma garde certaines propriétés de sécurité (à savoir la sécurité CPA) même lorsque les clés privées des utilisateurs sont divulguées, contrairement aux schémas [Bic+10 ; PS16]. Le fait que l'anonymat désintéressé dépende de la non-diffamation peut paraître surprenant mais cela est dû au processus d'ouverture spécial utilisé par notre réduction $\mathcal{R}$ dans la preuve de sécurité. L'idée est que $\mathcal{R}$ a la possibilité d'ouvrir toutes les signatures sauf celles générées par l'utilisateur « challengé ». Pour passer outre ce problème, la réduction $\mathcal{R}$ stocke toutes les signatures qu'elle a produites au nom de cet utilisateur pour pouvoir les reconnaître lorsque éventuellement celles-ci seront demandées à l'oracle $\mathcal{O}\texttt{Open}$. Cependant, cela fonctionne tant que l'adversaire est incapable de contrefaire des signatures pour cet utilisateur, d'où l'exigence de la non-diffamation.

La dernière affirmation du Théorème IV.3.1 indique que le schéma de signature de groupe réalise la plus forte forme d'anonymat lorsque nous faisons la supposition supplémentaire que l'autorité d'ouverture est aussi le gestionnaire du groupe, comme dans le modèle de Bellare, Micciancio et Warinschi [BMW03].

IV.3.1 Preuve de la Traçabilité

On montre que toute signature de groupe intraçable peut être utilisée pour construire une contrefaçon d'une signature de classe d'équivalence du schéma FHS. Plus spécifiquement, soit $\mathcal{A}$ un adversaire contre la notion de traçabilité de notre signature de groupe réussissant avec probabilité ε, alors $\mathcal{A}$ peut être converti en un adversaire contre la notion de sécurité EUF-CMA du schéma FHS réussissant avec la même probabilité.

Techniquement, l'attaque de $\mathcal{A}$ peut aboutir en retournant une signature valide σ sur m telle que, soit la procédure d'ouverture échoue, soit l'ouverture réussit mais il est impossible de produire une preuve d'ouverture valide. Nous pouvons exclure la seconde

alternative car le système de preuves Groth-Sahai est correct et sûr.

Notre réduction $\mathcal{R}$ génère les paramètres publics, exception faite qu'elle conserve x après avoir généré X et $\tilde{X}$. Elle récupère ensuite la clé publique $\tilde{A}_1$ et $\tilde{A}_2$ du challenger de la sécurité EUF-CMA et définit la clé publique du gestionnaire de groupe comme $(\tilde{A}_1, \tilde{A}_2, \tilde{A}_2^x)$ En utilisant son oracle de signature, la réduction est capable de gérer toutes les demandes `Join` et ainsi la simulation est parfaite. À la fin du jeu, $\mathcal{A}$ renvoie avec probabilité ε une signature de groupe σ sur m intraçable. Si on écrit σ comme $(\tau_1, \tau_2, \tilde{\tau}, \sigma_1, \sigma_2)$, cela signifie :

(1) $e\left(\sigma_1, \tilde{A}_1 \tilde{B}^{-1/h(\tilde{\tau}||\sigma_1||m)}\right) e(\sigma_2, \tilde{A}_2) = e(\tau_1, \tilde{\tau})$;

(2) $e(\tau_2, \tilde{g}) = e(g, \tilde{\tau})$;

(3) $e(\sigma_2, \tilde{g}) e\left(\sigma_1, \tilde{X}^{-1/h(\tilde{\tau}||\sigma_1||m)}\right) \neq e(\sigma_1, \tilde{g}^{y_i})$ pour tout $\tilde{g}^{y_i}$ stocké (chiffré).

L'équation (1) est équivalente à :

$$e(\sigma_1, \tilde{A}_1) e\left(\sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)}, \tilde{A}_2\right) = e(\tau_1, \tilde{\tau}),$$

ce qui signifie, à l'aide de l'équation (2), que $(\tau_1, \tau_2, \tilde{\tau})$ est une signature FHS sur le message $(\sigma_1, \sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)})$. Cependant, $(\tau_1, \tau_2, \tilde{\tau})$ est considérée comme une contrefaçon valide seulement si $(\sigma_1, \sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)})$ n'appartient pas à la classe d'équivalence d'un message adressé à l'oracle de signature.

Soit $\mathcal{S} = \{(h_i, h_i^{y_i})\}_{i=1}^q$, pour $h_i \in \mathbb{G}_1$, l'ensemble des messages requêtés. Tous ces messages ont été échangés lors d'un protocole `Join` pendant lequel le gestionnaire de groupe a reçu (et stocké) $\tilde{g}^{y_i}$ (chiffré). En conséquence, si (μ_1, μ_2) est dans la même classe d'équivalence qu'un élément de $\mathcal{S}$, alors il existe $1 \leqslant i \leqslant q$ tel que $e(\mu_1, \tilde{g}^{y_i}) = e(\mu_2, \tilde{g})$.

Supposons alors que $\left(\sigma_1, \sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)}\right)$ satisfait à la condition précédente, à savoir qu'il existe un i tel que

$$e(\sigma_1, \tilde{g}^{y_i}) = e\left(\sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)}, \tilde{g}\right).$$

Cela conduit à

$$\begin{aligned} e(\sigma_1, \tilde{g}^{y_i}) &= e\left(\sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)}, \tilde{g}\right), \\ &= e\left(\sigma_2, \tilde{g}\right) e\left(\sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)}, \tilde{g}\right), \\ &= e\left(\sigma_2, \tilde{g}\right) e\left(\sigma_1, \tilde{X}^{-1/h(\tilde{\tau}||\sigma_1||m)}\right), \end{aligned}$$

ce qui contredit l'équation (3). La paire $\left(\sigma_1, \sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)}\right)$ n'a donc pas été signée, ni

aucun représentant de la même classe d'équivalence, ce qui implique que $(\tau_1, \tau_2, \tilde{\tau})$ avec le représentant $\left(\sigma_1, \sigma_2 \sigma_1^{-x/h(\tilde{\tau}||\sigma_1||m)}\right)$ est une contrefaçon valide contre la sécurité EUF-CMA du schéma FHS.

IV.3.2 Preuve de l'Anonymat

Les preuves d'anonymat CPA et anonymat désintéressé sont très similaires et ne diffèrent que par un jeu. Considérons donc un adversaire contre « l'anonymat » sans préciser lequel sauf au Jeu 5 où la distinction sera faite. Nous discutons le cas de l'anonymat total dans la Remarque IV.3.2.

Soit $\mathcal{A}$ un adversaire contre l'anonymat de notre construction réussissant avec probabilité ε. Nous définissons une suite de jeux montrant que cet avantage est négligeable. Pour chaque Jeu i, nous définissons $\mathtt{Adv}_i = |\mathtt{Pr}(S_i) - 1/2|$, où S_i est l'évènement « $\mathcal{A}$ réussit au cours du Jeu i ». Nous définissons aussi $\mathtt{Adv}_{\mathsf{SXDH}}$ comme l'avantage contre le problème SXDH.

Jeu 1. Le premier jeu est exactement celui de l'anonymat Figure IV.1 où la réduction $\mathcal{R}$ génère normalement toutes les valeurs secrètes et est ainsi capable de répondre à toutes les requêtes faites aux oracles. Par définition, le premier avantage est $\mathtt{Adv}_1 = \varepsilon$.

Jeu 2. Dans le second jeu, $\mathcal{R}$ choisit un indice $1 \leqslant i^* \leqslant q_A$ aléatoire, où q_A est une borne sur le nombre de requêtes à $\mathcal{O}\mathtt{Add}$. $\mathcal{R}$ procède comme à l'accoutumé mais abandonne si $\mathcal{A}$ requête (i_0, i_1, m) à l'oracle $\mathcal{O}\mathtt{Ch}$ avec $i_b \neq i^*$, pour $b \in \{0, 1\}$. L'avantage de $\mathcal{A}$ dans ce nouveau jeu est alors au moins

$$\mathtt{Adv}_2 \geqslant \varepsilon / q_A.$$

Jeu 3. Dans le troisième jeu, $\mathcal{R}$ génère une chaine de référence commune `crs` truquée et simule alors toutes les preuves sans divulgation de connaissance. Tout changement de comportement de $\mathcal{A}$ peut alors être utilisé pour distinguer une `crs` truquée d'une `crs` générée normalement, ce qui est impossible sous l'hypothèse SXDH dans notre configuration. Ainsi,

$$\mathtt{Adv}_3 \geqslant \mathtt{Adv}_2 - \mathtt{Adv}_{\mathsf{SXDH}}.$$

Jeu 4. Dans le quatrième jeu, $\mathcal{R}$ fixe opk comme clé publique de l'expérience IND-CCA2. La réduction utilise ensuite l'oracle de déchiffrement pour déchiffrer le C_i stocké

dans $\mathbf{Reg}[i]$ pour tout utilisateur i et ainsi elle peut répondre à toute requête, de façon habituelle. Cependant, en recevant une requête $\mathcal{O}\mathtt{Join}_U$ pour i^* (ce qui arrive forcément à cause du Jeu 2), elle procède normalement sauf qu'elle génère C comme le chiffré d'un élément aléatoire de $\mathbb{G}_2$ et simule la preuve. Un changement dans le comportement de $\mathcal{A}$ impliquerait une attaque contre la sécurité IND-CCA2 de Γ, d'où

$$\mathtt{Adv}_4 \geqslant \mathtt{Adv}_3 - \mathtt{Adv}_{\mathsf{IND-CCA2}}.$$

Jeu 5. Dans le cinquième jeu, $\mathcal{R}$ stocke, dans un registre **Sig**, toutes les signatures qu'elle génère au nom de i^*. En recevant une requête `Open` pour une paire (σ, m), elle vérifie d'abord si $\sigma \in \mathbf{Sig}$ et dans ce cas, elle retourne i^* avec une preuve simulée. Sinon, elle retourne $\mathtt{Open}(\sigma, m)$.

Remarquons que le Jeu 5 est identique au Jeu 4 lorsque l'on considère l'anonymat CPA vu qu'il n'y a pas de requête à $\mathcal{O}\mathtt{Open}$ dans ce cas. Pour l'anonymat désintéressé, une seule différence à lieu lorsque l'adversaire parvient à présenter une signature contrefaite qui peut être rattachée à i^*. Cependant, un tel adversaire peut directement être converti en un adversaire contre la non-diffamation. Ainsi

$$\mathtt{Adv}_5 \geqslant \mathtt{Adv}_4 - \mathtt{Adv}^{nf}.$$

Jeu 6. Dans le sixième jeu, $\mathcal{R}$ procède comme dans le jeu précédent sauf qu'elle répond à la requête $\mathcal{O}\mathtt{Ch}$ en retournant une signature générée en utilisant une clé aléatoire. L'avantage de $\mathcal{A}$ ne peut alors qu'être 0. Nous prouvons plus bas que les Jeux 5 et 6 ne peuvent être distingués sous l'hypothèse SXDH, ce qui implique

$$\mathtt{Adv}_6 \geqslant \mathtt{Adv}_5 - \mathtt{Adv}_{\mathsf{SXDH}}.$$

Preuve d'indistinguabilité entre les jeux d'anonymat 5 *et* 6. La réduction $\mathcal{R}$ reçoit le uplet (g, g^a, g^b, g^z), un challenge DDH dans $\mathbb{G}_1$, et doit alors décider si $z = ab$ ou non. Elle va agir comme si $y = a$ pour la clé secrète usk_{i^*} de l'utilisateur i^*. Ce n'est pas un problème vu que g^a est suffisant pour produire des signatures de groupe et rejoindre le groupe depuis le Jeu 4. De plus, le Jeu 5 assure que $\mathcal{R}$ a la capacité de répondre aux requêtes $\mathcal{O}\mathtt{Open}$, même sans connaître $\tilde{g}^a$.

Pour répondre à la requête $\mathcal{O}\mathtt{Ch}$ pour un message m, elle choisit un scalaire aléatoire

t et calcule une signature de groupe σ comme suit :

$$\begin{aligned} \tau_1 &\leftarrow ((g^b)^{\alpha_1}(g^z)^{\alpha_2})^t; \\ (\tau_2, \tilde{\tau}) &\leftarrow (g^{1/t}, \tilde{g}^{1/t}); \\ (\sigma_1, \sigma_2) &\leftarrow (g^b, (g^b)^{x/h(\tilde{\tau}||\sigma_1||m)}(g^z)). \end{aligned}$$

Dans tous les cas, σ est une signature de groupe valide sur m, *i.e.* $\texttt{Verify}(\mathsf{gpk}, \sigma, m)$ retourne le bit 1. Si $z = ab$, alors σ est un signature valide produite par l'utilisateur i^* et $\mathcal{A}$ est toujours en train de jouer le Jeu 5. Sinon, σ est une signature produite par une clé aléatoire, indépendante de a, et $\mathcal{A}$ est en train de jouer le Jeu 6. Tout changement de comportement de $\mathcal{A}$ entre ces deux jeux peut être utilisé contre l'hypothèse DDH dans $\mathbb{G}_1$ et donc contre l'hypothèse SXDH. □

Au final, nous obtenons le résultat suivant :

- $\varepsilon/q_A \leqslant 2\ \mathtt{Adv}_{\mathsf{SXDH}} + \mathtt{Adv}_{\mathsf{IND-CCA2}}$ pour tout adversaire réussissant contre l'anonymat CPA avec probabilité ε ;
- $\varepsilon/q_A \leqslant 2\ \mathtt{Adv}_{\mathsf{SXDH}} + \mathtt{Adv}_{\mathsf{IND-CCA2}} + \mathtt{Adv}^{nf}$ pour tout adversaire réussissant contre l'anonymat désintéressé avec probabilité ε ;

ce qui prouve l'anonymat CPA et l'anonymat désintéressé de notre construction.

Remarque IV.3.2. Considérons maintenant le cas où le gestionnaire de groupe et l'autorité d'ouverture sont fusionnés, comme dans le modèle BMW [BMW03]. Comme expliqué dans la Remarque IV.2.2, l'usage d'un chiffrement IND-CCA2 durant le protocole `Join` est seulement nécessaire lorsque l'autorité d'ouverture n'est pas impliquée dans ce processus, ce qui n'est désormais plus le cas ici. Nous pouvons donc nous débarrasser de Γ et supprimer le Jeu 4 de la preuve de sécurité.

Dans le Jeu 5, $\mathcal{R}$ procède comme suit. Elle enregistre toujours les signatures générées au nom de i^* dans **Sig** mais répond maintenant aux requêtes $\mathcal{O}\texttt{Open}$ sur (σ, m) comme suit :

— si $\sigma \in \mathbf{Sig}$, alors elle retourne i^* accompagné d'une preuve simulée π ;
— si $\texttt{Open}(\sigma, m)$ renvoie (i, π) ou 0, alors elle retourne cette réponse à l'adversaire ;
— si $\texttt{Open}(\sigma, m)$ renvoie $\perp$, alors elle retourne i^* accompagné d'une preuve simulée π.

Notons qu'un problème peut survenir dans le troisième cas si l'adversaire parvient à présenter une signature de groupe qui ne peut pas être rattachée à un utilisateur enregistré. Cependant, cela voudrait dire que $\mathcal{A}$ est un adversaire contre la traçabilité, ce qui est considéré avoir une probabilité négligeable.

Tous les autres jeux restent inchangés donc $\varepsilon/q_A \leqslant 2\,\mathtt{Adv}_{\mathsf{SXDH}} + \mathtt{Adv}^{tra}$ pour tout adversaire gagnant contre l'anonymat total avec probabilité ε dans le modèle BMW [BMW03].

IV.3.3 Preuve de la Non-Diffamation

Un adversaire $\mathcal{A}$ gagnant contre la non-diffamation de notre schéma est capable de contrefaire une signature de groupe valide $\sigma = (\tau_1, \tau_2, \tilde{\tau}, \sigma_1, \sigma_2)$ sur un message m qui peut être rattachée à un certain utilisateur honnête i. Nous distinguons quatre cas :

— Cas 1 : la paire (g^u, g^{uy}) stockée dans $\mathbf{Reg}[i]$ n'est pas celle envoyée par l'utilisateur i lorsqu'il rejoint le groupe ;
— Cas 2 : l'utilisateur i a produit une signature $(\tau_1', \tau_2', \tilde{\tau}', \sigma_1', \sigma_2')$ sur m' telle que $(\tilde{\tau}', \sigma_1', m') \neq (\tilde{\tau}, \sigma_1, m)$ mais $h(\tilde{\tau}||\sigma_1||m) = h(\tilde{\tau}'||\sigma_1'||m')$;
— Cas 3 : l'utilisateur i a produit une signature $(\tau_1', \tau_2', \tilde{\tau}', \sigma_1', \sigma_2')$ sur m' telle que $(\tilde{\tau}', \sigma_1', m') = (\tilde{\tau}, \sigma_1, m)$;
— Cas 4 : aucun des précédents évènements n'est survenu.

Cas 1 et 2. Rappelons qu'un utilisateur peut être accusé d'avoir produit une signature seulement si son ouverture est valide, *i.e.* seulement si l'algorithme `Judge` retourne le bit 1. Une étape de l'algorithme `Judge` est de vérifier que l'utilisateur i a bien signé les éléments stockés dans $\mathbf{Reg}[i]$, à savoir s'il y a une signature μ dans $\mathbf{Reg}[i]$ telle que $\Sigma.\mathtt{Verify}(\mathsf{pk}_i, \mu, (g^u||g^{u\cdot y}||C_i||\pi_i)) = 1$. Ainsi, le cas 1 implique directement une contrefaçon de Σ, qui est supposée être EUF-CMA.

Idem, le cas 2 implique directement une collision de la fonction de hachage h.

Cas 3. Dans le troisième cas, l'algorithme `Judge` retourne 1, la signature σ est valide, ce qui signifie :

(1) $e(\sigma_1, \tilde{A}_1\tilde{B}^{-1/h(\tilde{\tau}||\sigma_1||m)})e(\sigma_2, \tilde{A}_2) = e(\tau_1, \tilde{\tau})$,
(2) $e(\tau_2, \tilde{g}) = e(g, \tilde{\tau})$.

De plus, comme $\tilde{\tau} = \tilde{\tau}'$, nous déduisons

$$e(\tau_2, \tilde{g}) = e(g, \tilde{\tau}) = e(g, \tilde{\tau}') = e(\tau_2', \tilde{g})$$

et donc $\tau_2 = \tau_2'$. En outre, σ est une signature rattachable à l'utilisateur i :

$$e(\sigma_2, \tilde{g})e\left(\sigma_1, \tilde{X}^{-1/h(\tilde{\tau}||\sigma_1||m)}\right) = e(\sigma_1, \tilde{g}^{y_i}).$$

Nous pouvons réécrire cette équation, sachant $\sigma_1 = \sigma'_1$,

$$e(\sigma_2, \tilde{g}) = e\left(\sigma_1, \tilde{X}^{1/h(\tilde{\tau}||\sigma_1||m)}\tilde{g}^{y_i}\right) = e\left(\sigma'_1, \tilde{X}^{1/h(\tilde{\tau}'||\sigma'_1||m')}\tilde{g}^{y_i}\right) = e(\sigma'_2, \tilde{g}),$$

et donc $\sigma_2 = \sigma'_2$. En combinant cette égalité avec (1), nous trouvons $\tau_1 = \tau'_1$. En conséquence, σ est exactement la signature $(\tau'_1, \tau'_2, \tilde{\tau}', \sigma'_1, \sigma'_2)$ renvoyée par l'oracle $\mathcal{O}\mathtt{Sign}$ et ne constitue donc pas une attaque valide contre la non-diffamation.

Cas 4. Prouvons dans le cas 4 qu'un adversaire gagnant l'expérience de non-diffamation casse l'hypothèse MPS. Soit $(\mathsf{g}, \mathsf{g}^y, \mathsf{g}^z, \mathsf{g}^{zx}, \tilde{\mathsf{g}}, \tilde{\mathsf{g}}^x, \tilde{\mathsf{g}}^y)$ une instance MPS pour la fonction h définie dans pp. La réduction $\mathcal{R}$ génère les paramètres publics pp comme à l'accoutumé sauf pour $(g, X) = (\mathsf{g}^z, \mathsf{g}^{zx})$ et $(\tilde{g}, \tilde{X}) = (\tilde{\mathsf{g}}, \tilde{\mathsf{g}}^x)$. Elle génère ensuite les clés $(\mathsf{osk}, \mathsf{opk})$ et $(\mathsf{gsk}, \mathsf{gpk})$ comme dans la Sous-Section IV.2.4, sélectionne $1 \leqslant i^* \leqslant q_A$ (où q_A est la borne sur le nombre de requêtes à l'oracle $\mathcal{O}\mathtt{Add}$) et traite les requêtes comme suit :

- $\mathcal{O}\mathtt{Add}(i)$: $\mathcal{R}$ répond à toutes les requêtes normalement avec $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \Sigma.\mathtt{Keygen}$.
- $\mathcal{O}\mathtt{Corrupt}(i)$: $\mathcal{R}$ retourne les clés secrètes demandées si $i \neq i^*$ et échoue sinon.
- $\mathcal{O}\mathtt{Join}_U(i)$: $\mathcal{R}$ procède normalement si $i \neq i^*$ en générant les valeurs secrètes nécessaires. Sinon, elle chiffre $\tilde{\mathsf{g}}^y$ comme C et envoie $(\mathsf{g}, \mathsf{g}^y)$ accompagné de la preuve NIZK π correspondante. Notons que $(\mathsf{g}, \mathsf{g}^y)$ est une paire valide pour $\tilde{\mathsf{g}}^y$ (donc la preuve NIZK peut être générée normalement) ce qui définit implicitement u comme z^{-1}.
- $\mathcal{O}\mathtt{Sign}(i, m)$: ici nous considérons seulement le cas où $i = i^*$ vu que $\mathcal{R}$ peut produire normalement toutes les autres signatures. $\mathcal{R}$ transfère le message m à l'oracle $\mathcal{O}$ résolvant le problème MPS qui renvoie $(s, \tilde{\mathsf{g}}^{rs}, \mathsf{g}^r, \mathsf{g}^{rz}, \mathsf{g}^{r(x+ty)})$ où le scalaire $t = h(\tilde{\mathsf{g}}^{rs}||\mathsf{g}^r||m)$. $\mathcal{R}$ construit alors la signature de groupe σ suivante :
  - $(\sigma'_1, \sigma'_2) \leftarrow (\mathsf{g}^r, (\mathsf{g}^{r(x+ty)})^{1/t})$;
  - $(\tau'_2, \tilde{\tau}') \leftarrow ((\mathsf{g}^{rz})^s, (\tilde{\mathsf{g}}^r)^s)$;
  - $\tau'_1 \leftarrow (\mathsf{g}^{\alpha_1}(\mathsf{g}^y)^{\alpha_2})^{1/s})$;

  où $(\alpha_1, \alpha_2) = \mathsf{osk}$. C'est bien une signature de groupe valide car :

$$\begin{aligned} e(\sigma'_1, &\tilde{A}_1\tilde{B}^{-1/h(\tilde{\tau}'||\sigma'_1||m)})e(\sigma'_2, \tilde{A}_2) \\ &= e(\mathsf{g}^r, \tilde{\mathsf{g}}^{\alpha_1 - (\alpha_2 x)/h(\tilde{\mathsf{g}}^{rs}||\mathsf{g}^r||m)})e(\mathsf{g}^{r(x/h(\tilde{\mathsf{g}}^{rs}||\mathsf{g}^r||m)+y)}, \tilde{\mathsf{g}}^{\alpha_2}), \\ &= e(\mathsf{g}^r, \tilde{\mathsf{g}}^{\alpha_1})e(\mathsf{g}^{ry}, \tilde{\mathsf{g}}^{\alpha_2}), \\ &= e(\tau'_1, \tilde{\tau}'), \end{aligned}$$

et $e(\tau_2', \tilde{g}) = e((\mathsf{g}^{rz})^s, \tilde{\mathsf{g}}) = (\mathsf{g}^z, \tilde{\mathsf{g}}^{rs}) = e(g, \tilde{\tau})$. De plus, σ suit la bonne distribution car cette construction définit implicitement r et s de l'algorithme `Sign` comme r/z et $1/rs$ de l'instance MPS, où r et s sont aléatoires.

Nous remarquons que $\mathcal{R}$ est parfaitement capable de répondre à toutes les requêtes tant que sa supposition sur i^* est correcte, évènement qui survient avec probabilité au moins $1/q_A$. Dans ce cas, $\mathcal{A}$ retourne, avec une certaine probabilité ε, une signature valide $\sigma = (\tau_1, \tau_2, \tilde{\tau}, \sigma_1, \sigma_2)$ sur un message m qui permet de remonter jusqu'à i^*. Concrètement, cela signifie que σ satisfait aux différentes conditions :

- $e(\sigma_1, \tilde{A}_1 \tilde{B}^{-1/t}) e(\sigma_2, \tilde{A}_2) = e(\tau_1, \tilde{\tau})$;
- $e(\tau_2, \tilde{g}) = e(g, \tilde{\tau})$;
- $e(\sigma_2, \tilde{g}) e(\sigma_1, \tilde{X}^{-1/t}) = e(\sigma_1, \tilde{g}^y)$;

avec $t = h(\tilde{\tau} || \sigma_1 || m)$. Comme nous considérons le cas 4, nous savons que la valeur de t est différente de celles utilisées par $\mathcal{R}$ pour répondre aux requêtes de $\mathcal{O}$`Sign`. En outre, la troisième équation signifie

$$\begin{aligned} e(\sigma_2^t, \tilde{\mathsf{g}}) &= e(\sigma_2^t, \tilde{g}), \\ &= e(\sigma_1, \tilde{X}(\tilde{g}^y)^t), \\ &= e(\sigma_1, \tilde{\mathsf{g}}^x(\tilde{\mathsf{g}}^y)^t), \end{aligned}$$

ce qui implique que $(t, \sigma_1, \sigma_2^t)$ est une solution valide au problème MPS. Tout adversaire correspondant au cas 4, gagnant avec une probabilité ε, peut alors être converti en un adversaire contre l'hypothèse MPS avec une probabilité de succès au moins ε/q_A. Ainsi, le cas 4 ne survient qu'avec une probabilité négligeable sous l'hypothèse MPS.

Conclusion du quatrième chapitre

Nous avons décrit dans ce chapitre un schéma de signature de groupe efficace, dont la complexité est comparable aux constructions prouvées à l'aide de l'oracle aléatoire. Nous avons accompli cela en adaptant les signatures randomisables PS [PS16] et FHS [FHS19], permettant ainsi de combiner les équations de vérifications de ces deux schémas. Comparée aux meilleures alternatives prouvées sûres en dehors du ROM, cette signature de groupe réduit de moitié le coût calculatoire et la taille des signatures, avec seulement quatre éléments de $\mathbb{G}_1$ et un de $\mathbb{G}_2$.

Conclusion

Dans ce mémoire, nous avons abordé la construction de courbes elliptiques à couplage avec la particularité de partir de besoins cryptographiques précis, afin de proposer les courbes les plus adaptées à certains cas d'usage. Cette approche a été rendue possible par la perte d'hégémonie des courbes Barreto-Naehrig qui étaient quasi optimales jusque-là. Cela nous a conduit à proposer une nouvelle courbe grâce à la construction de Brezing et Weng. Cette courbe est assez inhabituelle par son degré de plongement premier (19), mais offre de bonnes performances pour le signataire dans le contexte des schémas dérivés des signatures de groupe, comme les DAA ou les EPID. La suite de ces travaux serait de dégager d'autres familles de besoins pour divers schémas, comme par exemple des courbes ayant un coût équilibré entre les opérations de $\mathbb{G}_1$ et $\mathbb{G}_2$ (dans le sens où on ne souhaite pas trop pénaliser $\mathbb{G}_2$) pour le système de preuve Groth-Sahai [GS08]. Les résultats de Bos, Costello et Naehrig [BCN14] sur la comparaison de modèles d'équation sont très pertinents dans ce cas, car nous pensons que les courbe BLS de degré de plongement 24 peuvent être des courbes permettant une plus grande adaptation, un plus grand choix de paramètres à affiner.

En seconde partie de ce manuscrit, nous avons vu plus en action les couplages sur courbes elliptiques en proposant un nouveau schéma de signature de groupe. En se démarquant des constructions précédentes qui utilisaient le paradigme SEP (Signer-Chiffrer-Prouver), et ne pouvaient donc pas tirer partie de possibles interactions entre le certificat d'appartenance au groupe et la signature, notre schéma réussit à être le plus efficace (en termes de taille de signature) comparé au autres n'utilisant pas le ROM, et fait même concurrence à ceux prouvés dans le modèle de l'oracle aléatoire. Mais, au-delà de ces simples gains de performance, il illustre à nouveau toute la force des couplages pour la cryptographie. Ces derniers permettent en effet d'éviter d'avoir recours aux coûteuses preuves sans divulgation de connaissance (*zero-knowledge*), outils jusque-là incontournables pour les signatures de groupe, quel que soit l'environnement mathématique considéré (groupes cycliques, réseaux euclidiens, etc).

Plusieurs décennies après leur introduction en cryptographie, les couplages restent encore un outil fondamental pour la conception de protocoles de sécurité, renforçant encore

davantage la suprématie des courbes elliptiques sur ce domaine. Ils n'en restent pas moins un outil encore partiellement inexploré, dont la sécurité n'est pas pleinement appréhendée, et constituent donc toujours un sujet d'étude majeur en cryptographie.

Références

Cette thèse a été rédigée en LaTeX en utilisant le modèle mis à disposition par l'École des Docteurs Bretagne-Loire :

`https://gitlab.inria.fr/ed-mathstic/latex-template`

Plusieurs figures de ce manuscrit sont des adaptations de figures provenant de **TikZ for Cryptographers** disponibles à l'adresse :

`https://www.iacr.org/authors/tikz/`

On y trouvera notamment l'illustration de la méthode Pohlig-Hellman de la page 27 et toutes les illustrations de courbes : les cubiques singulières (Figures II.1 en page 31 et II.2 en page 32), les courbes elliptiques (Figure II.3 en page 33) et leur loi de groupe (Figures II.6, II.7 et II.8 respectivement en pages 38, 40 et 42).

L'illustration projective du point à l'infini (Figure II.5 en page 37) est une adaptation de l'image vectorisée :

`https://commons.wikimedia.org/wiki/File:Torsion_on_cubic_curve.jpg`

La Figure II.4 (page 35) illustrant le changement de modèles de Weierstrass sur un exemple est adaptée du site `trustica.cz` :

`https://trustica.cz/en/2018/02/22/introduction-to-elliptic-curves/`

BIBLIOGRAPHIE

[AHO16] Masayuki ABE, Fumitaka HOSHINO et Miyako OHKUBO. « Design in Type-I, Run in Type-III : Fast and Scalable Bilinear-Type Conversion Using Integer Programming ». In : *Advances in Cryptology – CRYPTO 2016, Part III*. Sous la dir. de Matthew ROBSHAW et Jonathan KATZ. T. 9816. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2016, p. 387-415. DOI : 10.1007/978-3-662-53015-3_14.

[AlL16] Guy ALLEE. *EPID for IoT Identity*. https://img.en25.com/Web/McAfeeE10BuildProd{a6dd7393-63f8-4c08-b3aa-89923182a7e5}_EPID_Overview_Public_2016-02-08.pdf. 2016.

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

[APM04] Vijayalakshmi ATLURI, Birgit PFITZMANN et Patrick MCDANIEL, éd. *ACM CCS 2004 : 11th Conference on Computer and Communications Security*. Washington, DC, USA : ACM Press, oct. 2004.

[Ara+] D. F. ARANHA, C. P. L. GOUVÊA, T. MARKMANN, R. S. WAHBY et K. LIAO. *RELIC is an Efficient LIbrary for Cryptography*. https://github.com/relic-toolkit/relic.

[Ate+00] Giuseppe ATENIESE, Jan CAMENISCH, Marc JOYE et Gene TSUDIK. « A Practical and Provably Secure Coalition-Resistant Group Signature Scheme ». In : *Advances in Cryptology – CRYPTO 2000*. Sous la dir. de Mihir BELLARE. T. 1880. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2000, p. 255-270. DOI : 10.1007/3-540-44598-6_16.

[AW04] Michel ABDALLA et Bogdan WARINSCHI. « On the Minimal Assumptions of Group Signature Schemes ». In : *ICICS 04 : 6th International Conference on Information and Communication Security*. Sous la dir. de Javier LÓPEZ, Sihan QING et Eiji OKAMOTO. T. 3269. Lecture Notes in Computer Science. Malaga, Spain : Springer, Heidelberg, Germany, oct. 2004, p. 1-13.

[Bac+18] Michael Backes, Lucjan Hanzlik, Kamil Kluczniak et Jonas Schneider. « Signatures with Flexible Public Key : Introducing Equivalence Classes for Public Keys ». In : *Advances in Cryptology – ASIACRYPT 2018, Part II*. Sous la dir. de Thomas Peyrin et Steven Galbraith. T. 11273. Lecture Notes in Computer Science. Brisbane, Queensland, Australia : Springer, Heidelberg, Germany, déc. 2018, p. 405-434. doi : 10.1007/978-3-030-03329-3_14.

[Bar+02] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn et Michael Scott. « Efficient Algorithms for Pairing-Based Cryptosystems ». In : *Advances in Cryptology – CRYPTO 2002*. Sous la dir. de Moti Yung. T. 2442. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2002, p. 354-368. doi : 10.1007/3-540-45708-9_23.

[Bar+07] Paulo S. L. M. Barreto, Steven D. Galbraith, Colm O'hEigeartaigh et Michael Scott. « Efficient pairing computation on supersingular Abelian varieties ». In : *Des. Codes Cryptogr.* 42.3 (2007), p. 239-271. doi : 10.1007/s10623-006-9033-6. url : https://doi.org/10.1007/s10623-006-9033-6.

[Bar+14] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux et Emmanuel Thomé. « A Heuristic Quasi-Polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic ». In : *Advances in Cryptology – EUROCRYPT 2014*. Sous la dir. de Phong Q. Nguyen et Elisabeth Oswald. T. 8441. Lecture Notes in Computer Science. Copenhagen, Denmark : Springer, Heidelberg, Germany, mai 2014, p. 1-16. doi : 10.1007/978-3-642-55220-5_1.

[Bar+15] Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic et François Morain. « Improving NFS for the Discrete Logarithm Problem in Non-prime Finite Fields ». In : *Advances in Cryptology – EUROCRYPT 2015, Part I*. Sous la dir. d'Elisabeth Oswald et Marc Fischlin. T. 9056. Lecture Notes in Computer Science. Sofia, Bulgaria : Springer, Heidelberg, Germany, avr. 2015, p. 129-155. doi : 10.1007/978-3-662-46800-5_6.

[BBS04] Dan Boneh, Xavier Boyen et Hovav Shacham. « Short Group Signatures ». In : *Advances in Cryptology – CRYPTO 2004*. Sous la dir. de Matthew Franklin. T. 3152. Lecture Notes in Computer Science. Santa Barbara, CA,

USA : Springer, Heidelberg, Germany, août 2004, p. 41-55. DOI : 10.1007/978-3-540-28628-8_3.

[BCC04] Ernest F. Brickell, Jan Camenisch et Liqun Chen. « Direct Anonymous Attestation ». In : *ACM CCS 2004 : 11th Conference on Computer and Communications Security*. Sous la dir. de Vijayalakshmi Atluri, Birgit Pfitzmann et Patrick McDaniel. Washington, DC, USA : ACM Press, oct. 2004, p. 132-145. DOI : 10.1145/1030083.1030103.

[BCN14] Joppe W. Bos, Craig Costello et Michael Naehrig. « Exponentiating in Pairing Groups ». In : *SAC 2013 : 20th Annual International Workshop on Selected Areas in Cryptography*. Sous la dir. de Tanja Lange, Kristin Lauter et Petr Lisonek. T. 8282. Lecture Notes in Computer Science. Burnaby, BC, Canada : Springer, Heidelberg, Germany, août 2014, p. 438-455. DOI : 10.1007/978-3-662-43414-7_22.

[BD19] Razvan Barbulescu et Sylvain Duquesne. « Updating Key Size Estimations for Pairings ». In : *Journal of Cryptology* 32.4 (oct. 2019), p. 1298-1336. DOI : 10.1007/s00145-018-9280-5.

[BF03] Dan Boneh et Matthew K. Franklin. « Identity Based Encryption from the Weil Pairing ». In : *SIAM Journal on Computing* 32.3 (2003), p. 586-615.

[Bic+10] Patrik Bichsel, Jan Camenisch, Gregory Neven, Nigel P. Smart et Bogdan Warinschi. « Get Shorty via Group Signatures without Encryption ». In : *SCN 10 : 7th International Conference on Security in Communication Networks*. Sous la dir. de Juan A. Garay et Roberto De Prisco. T. 6280. Lecture Notes in Computer Science. Amalfi, Italy : Springer, Heidelberg, Germany, sept. 2010, p. 381-398. DOI : 10.1007/978-3-642-15317-4_24.

[BK98] R. Balasubramanian et Neal Koblitz. « The Improbability That an Elliptic Curve Has Subexponential Discrete Log Problem under the Menezes - Okamoto - Vanstone Algorithm ». In : *Journal of Cryptology* 11.2 (mars 1998), p. 141-145. DOI : 10.1007/s001459900040.

[BL07a] Daniel J. Bernstein et Tanja Lange. *Analysis and optimization of elliptic-curve single-scalar multiplication*. Cryptology ePrint Archive, Report 2007/455. https://eprint.iacr.org/2007/455. 2007.

[BL07b] Ernie Brickell et Jiangtao Li. « Enhanced privacy id : a direct anonymous attestation scheme with enhanced revocation capabilities ». In : *Proceedings of the 2007 ACM Workshop on Privacy in the Electronic Society, WPES 2007, Alexandria, VA, USA, October 29, 2007*. Sous la dir. de Peng Ning et Ting Yu. ACM, 2007, p. 21-30. isbn : 978-1-59593-883-1. doi : 10.1145/1314333.1314337. url : https://doi.org/10.1145/1314333.1314337.

[BL10] Ernie Brickell et Jiangtao Li. « Enhanced Privacy ID from Bilinear Pairing for Hardware Authentication and Attestation ». In : *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2010, Minneapolis, Minnesota, USA, August 20-22, 2010*. Sous la dir. d'Ahmed K. Elmagarmid et Divyakant Agrawal. IEEE Computer Society, 2010, p. 768-775. doi : 10.1109/SocialCom.2010.118. url : https://doi.org/10.1109/SocialCom.2010.118.

[BLS01] Dan Boneh, Ben Lynn et Hovav Shacham. « Short Signatures from the Weil Pairing ». In : *Advances in Cryptology – ASIACRYPT 2001*. Sous la dir. de Colin Boyd. T. 2248. Lecture Notes in Computer Science. Gold Coast, Australia : Springer, Heidelberg, Germany, déc. 2001, p. 514-532. doi : 10.1007/3-540-45682-1_30.

[BLS03] Paulo S. L. M. Barreto, Ben Lynn et Michael Scott. « Constructing Elliptic Curves with Prescribed Embedding Degrees ». In : *SCN 02 : 3rd International Conference on Security in Communication Networks*. Sous la dir. de Stelvio Cimato, Clemente Galdi et Giuseppe Persiano. T. 2576. Lecture Notes in Computer Science. Amalfi, Italy : Springer, Heidelberg, Germany, sept. 2003, p. 257-267. doi : 10.1007/3-540-36413-7_19.

[BLS11] Daniel J. Bernstein, Tanja Lange et Peter Schwabe. « On the Correct Use of the Negation Map in the Pollard rho Method ». In : *PKC 2011 : 14th International Conference on Theory and Practice of Public Key Cryptography*. Sous la dir. de Dario Catalano, Nelly Fazio, Rosario Gennaro et Antonio Nicolosi. T. 6571. Lecture Notes in Computer Science. Taormina, Italy : Springer, Heidelberg, Germany, mars 2011, p. 128-146. doi : 10.1007/978-3-642-19379-8_8.

[BMW03] Mihir Bellare, Daniele Micciancio et Bogdan Warinschi. « Foundations of Group Signatures : Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions ». In : *Advances in Cryptology – EUROCRYPT 2003*. Sous la dir. d'Eli Biham. T. 2656. Lecture Notes in Computer Science. Warsaw, Poland : Springer, Heidelberg, Germany, mai 2003, p. 614-629. doi : 10.1007/3-540-39200-9_38.

[BN06] Paulo S. L. M. Barreto et Michael Naehrig. « Pairing-Friendly Elliptic Curves of Prime Order ». In : *SAC 2005 : 12th Annual International Workshop on Selected Areas in Cryptography*. Sous la dir. de Bart Preneel et Stafford Tavares. T. 3897. Lecture Notes in Computer Science. Kingston, Ontario, Canada : Springer, Heidelberg, Germany, août 2006, p. 319-331. doi : 10.1007/11693383_22.

[BS04] Dan Boneh et Hovav Shacham. « Group Signatures With Verifier-Local Revocation ». In : *ACM CCS 2004 : 11th Conference on Computer and Communications Security*. Sous la dir. de Vijayalakshmi Atluri, Birgit Pfitzmann et Patrick McDaniel. Washington, DC, USA : ACM Press, oct. 2004, p. 168-177. doi : 10.1145/1030083.1030106.

[BSS05] Ian F Blake, Gadiel Seroussi et Nigel P Smart. *Advances in elliptic curve cryptography*. T. 317. Cambridge University Press, 2005.

[BSZ05] Mihir Bellare, Haixia Shi et Chong Zhang. « Foundations of Group Signatures : The Case of Dynamic Groups ». In : *Topics in Cryptology – CT-RSA 2005*. Sous la dir. d'Alfred Menezes. T. 3376. Lecture Notes in Computer Science. San Francisco, CA, USA : Springer, Heidelberg, Germany, fév. 2005, p. 136-153. doi : 10.1007/978-3-540-30574-3_11.

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

[BW06] Xavier Boyen et Brent Waters. « Compact Group Signatures Without Random Oracles ». In : *Advances in Cryptology – EUROCRYPT 2006*. Sous la dir. de Serge Vaudenay. T. 4004. Lecture Notes in Computer Science. St. Petersburg, Russia : Springer, Heidelberg, Germany, mai 2006, p. 427-444. doi : 10.1007/11761679_26.

[Can+15] Sébastien Canard, David Pointcheval, Olivier Sanders et Jacques Traoré. « Divisible E-Cash Made Practical ». In : *PKC 2015 : 18th International Conference on Theory and Practice of Public Key Cryptography*. Sous la dir. de Jonathan Katz. T. 9020. Lecture Notes in Computer Science. Gaithersburg, MD, USA : Springer, Heidelberg, Germany, mars 2015, p. 77-100. doi : 10.1007/978-3-662-46447-2_4.

[CC86] David V Chudnovsky et Gregory V Chudnovsky. « Sequences of numbers generated by addition in formal groups and new primality and factorization tests ». In : *Advances in Applied Mathematics* 7.4 (1986), p. 385-434.

[CDS20] Rémi Clarisse, Sylvain Duquesne et Olivier Sanders. « Curves with Fast Computations in the First Pairing Group ». In : *CANS 20 : 19th International Conference on Cryptology and Network Security*. Sous la dir. de Stephan Krenn, Haya Shulman et Serge Vaudenay. T. 12579. Lecture Notes in Computer Science. Vienna, Austria : Springer, Heidelberg, Germany, déc. 2020, p. 280-298. doi : 10.1007/978-3-030-65411-5_14.

[CGH04] Ran Canetti, Oded Goldreich et Shai Halevi. « On the Random-Oracle Methodology as Applied to Length-Restricted Signature Schemes ». In : *TCC 2004 : 1st Theory of Cryptography Conference*. Sous la dir. de Moni Naor. T. 2951. Lecture Notes in Computer Science. Cambridge, MA, USA : Springer, Heidelberg, Germany, fév. 2004, p. 40-57. doi : 10.1007/978-3-540-24638-1_3.

[CGP03] Stelvio Cimato, Clemente Galdi et Giuseppe Persiano, éd. *SCN 02 : 3rd International Conference on Security in Communication Networks*. T. 2576. Lecture Notes in Computer Science. Amalfi, Italy : Springer, Heidelberg, Germany, sept. 2003.

[CL03] Jan Camenisch et Anna Lysyanskaya. « A Signature Scheme with Efficient Protocols ». In : *SCN 02 : 3rd International Conference on Security in Communication Networks*. Sous la dir. de Stelvio Cimato, Clemente Galdi et Giuseppe Persiano. T. 2576. Lecture Notes in Computer Science. Amalfi, Italy : Springer, Heidelberg, Germany, sept. 2003, p. 268-289. doi : 10.1007/3-540-36413-7_20.

[CL04] Jan Camenisch et Anna Lysyanskaya. « Signature Schemes and Anonymous Credentials from Bilinear Maps ». In : *Advances in Cryptology – CRYPTO 2004*. Sous la dir. de Matthew Franklin. T. 3152. Lecture Notes

in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2004, p. 56-72. DOI : 10.1007/978-3-540-28628-8_4.

[CM15] Sanjit Chatterjee et Alfred Menezes. « Type 2 Structure-Preserving Signature Schemes Revisited ». In : *Advances in Cryptology – ASIACRYPT 2015, Part I*. Sous la dir. de Tetsu Iwata et Jung Hee Cheon. T. 9452. Lecture Notes in Computer Science. Auckland, New Zealand : Springer, Heidelberg, Germany, nov. 2015, p. 286-310. DOI : 10.1007/978-3-662-48797-6_13.

[Coh+05] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen et Frederik Vercauteren. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2005.

[Con] Keith Conrad. URL : https://kconrad.math.uconn.edu/blurbs/.

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

[CP01] C Cocks et RG Pinch. « ID-based cryptosystems based on the Weil pairing, 2001 ». In : *Unpublished manuscript* (2001).

[CS20] Rémi Clarisse et Olivier Sanders. « Group Signature Without Random Oracles from Randomizable Signatures ». In : *ProvSec 2020 : 14th International Conference on Provable Security*. Sous la dir. de Khoa Nguyen, Wenling Wu, Kwok-Yan Lam et Huaxiong Wang. T. 12505. Lecture Notes in Computer Science. Singapore : Springer, Heidelberg, Germany, nov. 2020, p. 3-23. DOI : 10.1007/978-3-030-62576-4_1.

[Cv91] David Chaum et Eugène van Heyst. « Group Signatures ». In : *Advances in Cryptology – EUROCRYPT'91*. Sous la dir. de Donald W. Davies. T. 547. Lecture Notes in Computer Science. Brighton, UK : Springer, Heidelberg, Germany, avr. 1991, p. 257-265. DOI : 10.1007/3-540-46416-6_22.

[DEM05] Régis Dupont, Andreas Enge et François Morain. « Building Curves with Arbitrary Small MOV Degree over Finite Prime Fields ». In : *Journal of Cryptology* 18.2 (avr. 2005), p. 79-89. DOI : 10.1007/s00145-004-0219-7.

[DH76] Whitfield Diffie et Martin E. Hellman. « New Directions in Cryptography ». In : *IEEE Transactions on Information Theory* 22.6 (1976), p. 644-654.

[DL03] Iwan M. Duursma et Hyang-Sook Lee. « Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$ ». In : *Advances in Cryptology – ASIACRYPT 2003*. Sous la dir. de Chi-Sung Laih. T. 2894. Lecture Notes in Computer Science. Taipei, Taiwan : Springer, Heidelberg, Germany, nov. 2003, p. 111-123. doi : 10.1007/978-3-540-40061-5_7.

[DS18] David Derler et Daniel Slamanig. « Highly-Efficient Fully-Anonymous Dynamic Group Signatures ». In : *ASIACCS 18 : 13th ACM Symposium on Information, Computer and Communications Security*. Sous la dir. de Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López et Taesoo Kim. Incheon, Republic of Korea : ACM Press, avr. 2018, p. 551-565.

[FHS15] Georg Fuchsbauer, Christian Hanser et Daniel Slamanig. « Practical Round-Optimal Blind Signatures in the Standard Model ». In : *Advances in Cryptology – CRYPTO 2015, Part II*. Sous la dir. de Rosario Gennaro et Matthew J. B. Robshaw. T. 9216. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2015, p. 233-253. doi : 10.1007/978-3-662-48000-7_12.

[FHS19] Georg Fuchsbauer, Christian Hanser et Daniel Slamanig. « Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials ». In : *Journal of Cryptology* 32.2 (avr. 2019), p. 498-546. doi : 10.1007/s00145-018-9281-4.

[FKR12] Laura Fuentes-Castañeda, Edward Knapp et Francisco Rodríguez-Henríquez. « Faster Hashing to $\mathbb{G}_2$ ». In : *SAC 2011 : 18th Annual International Workshop on Selected Areas in Cryptography*. Sous la dir. d'Ali Miri et Serge Vaudenay. T. 7118. Lecture Notes in Computer Science. Toronto, Ontario, Canada : Springer, Heidelberg, Germany, août 2012, p. 412-430. doi : 10.1007/978-3-642-28496-0_25.

[FR94] Gerhard Frey et Hans-Georg Rück. « A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves ». In : *Mathematics of computation* 62.206 (1994), p. 865-874.

[Fra04] Matthew Franklin, éd. *Advances in Cryptology – CRYPTO 2004*. T. 3152. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2004.

[FS87] Amos Fiat et Adi Shamir. « How to Prove Yourself : Practical Solutions to Identification and Signature Problems ». In : *Advances in Cryptology – CRYPTO'86*. Sous la dir. d'Andrew M. Odlyzko. T. 263. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 1987, p. 186-194. doi : 10.1007/3-540-47721-7_12.

[FST10] David Freeman, Michael Scott et Edlyn Teske. « A Taxonomy of Pairing-Friendly Elliptic Curves ». In : *Journal of Cryptology* 23.2 (avr. 2010), p. 224-280. doi : 10.1007/s00145-009-9048-z.

[Fuc15] L. Fuchs. *Abelian Groups*. Springer Monographs in Mathematics. Springer International Publishing, 2015. isbn : 9783319194226.

[GKZ14] Robert Granger, Thorsten Kleinjung et Jens Zumbrägel. « Breaking '128-bit Secure' Supersingular Binary Curves - (Or How to Solve Discrete Logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$) ». In : *Advances in Cryptology – CRYPTO 2014, Part II*. Sous la dir. de Juan A. Garay et Rosario Gennaro. T. 8617. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2014, p. 126-145. doi : 10.1007/978-3-662-44381-1_8.

[GLV01] Robert P. Gallant, Robert J. Lambert et Scott A. Vanstone. « Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms ». In : *Advances in Cryptology – CRYPTO 2001*. Sous la dir. de Joe Kilian. T. 2139. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2001, p. 190-200. doi : 10.1007/3-540-44647-8_11.

[GM17] Aurore Guillevic et François Morain. « Discrete logarithms ». In : *Guide to pairing-based cryptography*. Chapman et Hall/CRC, 2017, p. 9-1.

[GMR88] Shafi Goldwasser, Silvio Micali et Ronald L. Rivest. « A Digital Signature Scheme Secure Against Adaptive Chosen-message Attacks ». In : *SIAM Journal on Computing* 17.2 (avr. 1988), p. 281-308.

[GMT20] Aurore Guillevic, Simon Masson et Emmanuel Thomé. « Cocks-Pinch curves of embedding degrees five to eight and optimal ate pairing computation ». In : *Des. Codes Cryptogr.* 88.6 (2020), p. 1047-1081. doi : 10.1007/s10623-020-00727-w. url : https://doi.org/10.1007/s10623-020-00727-w.

[Gor93] Daniel M. Gordon. « Discrete Logarithms in *GF(P)* Using the Number Field Sieve ». In : *SIAM J. Discret. Math.* 6.1 (1993), p. 124-138. doi : 10.1137/0406010. url : https://doi.org/10.1137/0406010.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson et Nigel P. Smart. « Pairings for cryptographers ». In : *Discret. Appl. Math.* 156.16 (2008), p. 3113-3121. doi : 10.1016/j.dam.2007.12.010. url : https://doi.org/10.1016/j.dam.2007.12.010.

[GR15] Rosario Gennaro et Matthew J. B. Robshaw, éd. *Advances in Cryptology – CRYPTO 2015, Part II*. T. 9216. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2015.

[Gro07] Jens Groth. « Fully Anonymous Group Signatures Without Random Oracles ». In : *Advances in Cryptology – ASIACRYPT 2007*. Sous la dir. de Kaoru Kurosawa. T. 4833. Lecture Notes in Computer Science. Kuching, Malaysia : Springer, Heidelberg, Germany, déc. 2007, p. 164-180. doi : 10.1007/978-3-540-76900-2_10.

[Gro16] Jens Groth. « On the Size of Pairing-Based Non-interactive Arguments ». In : *Advances in Cryptology – EUROCRYPT 2016, Part II*. Sous la dir. de Marc Fischlin et Jean-Sébastien Coron. T. 9666. Lecture Notes in Computer Science. Vienna, Austria : Springer, Heidelberg, Germany, mai 2016, p. 305-326. doi : 10.1007/978-3-662-49896-5_11.

[GS08] Jens Groth et Amit Sahai. « Efficient Non-interactive Proof Systems for Bilinear Groups ». In : *Advances in Cryptology – EUROCRYPT 2008*. Sous la dir. de Nigel P. Smart. T. 4965. Lecture Notes in Computer Science. Istanbul, Turkey : Springer, Heidelberg, Germany, avr. 2008, p. 415-432. doi : 10.1007/978-3-540-78967-3_24.

[GS21] Aurore Guillevic et Shashank Singh. « On the Alpha Value of Polynomials in the Tower Number Field Sieve Algorithm ». In : *Mathematical Cryptology* 1.1 (fév. 2021), p. 39. url : https://hal.inria.fr/hal-02263098.

[Gui13] Aurore Guillevic. « Comparing the Pairing Efficiency over Composite-Order and Prime-Order Elliptic Curves ». In : *ACNS 13 : 11th International Conference on Applied Cryptography and Network Security*. Sous la dir. de Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel et Reihaneh Safavi-Naini. T. 7954. Lecture Notes in Computer Science. Banff,

AB, Canada : Springer, Heidelberg, Germany, juin 2013, p. 357-372. DOI : 10.1007/978-3-642-38980-1_22.

[Gui20] Aurore GUILLEVIC. « A Short-List of Pairing-Friendly Curves Resistant to Special TNFS at the 128-Bit Security Level ». In : *PKC 2020 : 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II*. Sous la dir. d'Aggelos KIAYIAS, Markulf KOHLWEISS, Petros WALLDEN et Vassilis ZIKAS. T. 12111. Lecture Notes in Computer Science. Edinburgh, UK : Springer, Heidelberg, Germany, mai 2020, p. 535-564. DOI : 10.1007/978-3-030-45388-6_19.

[Hin93] Sir Harry HINSLEY. *The Influence of ULTRA in the Second World War*. 1993. URL : http://www.cix.co.uk/~klockstone/hinsley.htm.

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

[Int16] INTEL. *A Cost-Effective Foundation for End-to-End IoT Security, white paper*. 2016. URL : https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/intel-epid-iot-security-white-paper.pdf.

[Jou04] Antoine JOUX. « A One Round Protocol for Tripartite Diffie-Hellman ». In : *Journal of Cryptology* 17.4 (sept. 2004), p. 263-276. DOI : 10.1007/s00145-004-0312-y.

[JP14] Antoine JOUX et Cécile PIERROT. « The Special Number Field Sieve in $\mathbb{F}_{p^n}$ - Application to Pairing-Friendly Constructions ». In : *PAIRING 2013 : 6th International Conference on Pairing-based Cryptography*. Sous la dir. de Zhenfu CAO et Fangguo ZHANG. T. 8365. Lecture Notes in Computer Science. Beijing, China : Springer, Heidelberg, Germany, nov. 2014, p. 45-61. DOI : 10.1007/978-3-319-04873-4_3.

[JP16] Antoine JOUX et Cécile PIERROT. « Technical history of discrete logarithms in small characteristic finite fields - The road from subexponential to quasi-polynomial complexity ». In : *Des. Codes Cryptogr.* 78.1 (2016), p. 73-85. DOI : 10.1007/s10623-015-0147-6. URL : https://doi.org/10.1007/s10623-015-0147-6.

[Kar95] Anatolii Alexeevich Karatsuba. « The complexity of computations ». In : *Proceedings of the Steklov Institute of Mathematics-Interperiodica Translation* 211 (1995), p. 169-183.

[KB16] Taechan Kim et Razvan Barbulescu. « Extended Tower Number Field Sieve : A New Complexity for the Medium Prime Case ». In : *Advances in Cryptology – CRYPTO 2016, Part I*. Sous la dir. de Matthew Robshaw et Jonathan Katz. T. 9814. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2016, p. 543-571. doi : 10.1007/978-3-662-53018-4_20.

[KJ17] Taechan Kim et Jinhyuck Jeong. « Extended Tower Number Field Sieve with Application to Finite Fields of Arbitrary Composite Extension Degree ». In : *PKC 2017 : 20th International Conference on Theory and Practice of Public Key Cryptography, Part I*. Sous la dir. de Serge Fehr. T. 10174. Lecture Notes in Computer Science. Amsterdam, The Netherlands : Springer, Heidelberg, Germany, mars 2017, p. 388-408. doi : 10.1007/978-3-662-54365-8_16.

[KKC13] Taechan Kim, Sungwook Kim et Jung Hee Cheon. « On the Final Exponentiation in Tate Pairing Computations ». In : *IEEE Trans. Inf. Theory* 59.6 (2013), p. 4033-4041. doi : 10.1109/TIT.2013.2240763. url : https://doi.org/10.1109/TIT.2013.2240763.

[Kob87] Neal Koblitz. « Elliptic curve cryptosystems ». In : *Mathematics of computation* 48.177 (1987), p. 203-209.

[Kob89] Neal Koblitz. « Hyperelliptic Cryptosystems ». In : *Journal of Cryptology* 1.3 (oct. 1989), p. 139-150. doi : 10.1007/BF02252872.

[Kob90] Neal Koblitz. « A Family of Jacobians Suitable for Discrete Log Cryptosystems ». In : *Advances in Cryptology – CRYPTO'88*. Sous la dir. de Shafi Goldwasser. T. 403. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 1990, p. 94-99. doi : 10.1007/0-387-34799-2_8.

[KSS08] Ezekiel J. Kachisa, Edward F. Schaefer et Michael Scott. « Constructing Brezing-Weng Pairing-Friendly Elliptic Curves Using Elements in the Cyclotomic Field ». In : *PAIRING 2008 : 2nd International Conference on Pairing-based Cryptography*. Sous la dir. de Steven D. Galbraith et Kenneth

G. Paterson. T. 5209. Lecture Notes in Computer Science. Egham, UK : Springer, Heidelberg, Germany, sept. 2008, p. 126-135. DOI : 10.1007/978-3-540-85538-5_9.

[Lan05] S. Lang. *Algebra*. Graduate Texts in Mathematics. Springer New York, 2005. ISBN : 9780387953854.

[Lib+14] Benoît Libert, Thomas Peters, Marc Joye et Moti Yung. « Non-malleability from Malleability : Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures ». In : *Advances in Cryptology – EUROCRYPT 2014*. Sous la dir. de Phong Q. Nguyen et Elisabeth Oswald. T. 8441. Lecture Notes in Computer Science. Copenhagen, Denmark : Springer, Heidelberg, Germany, mai 2014, p. 514-532. DOI : 10.1007/978-3-642-55220-5_29.

[LPY15] Benoît Libert, Thomas Peters et Moti Yung. « Short Group Signatures via Structure-Preserving Signatures : Standard Model Security from Simple Assumptions ». In : *Advances in Cryptology – CRYPTO 2015, Part II*. Sous la dir. de Rosario Gennaro et Matthew J. B. Robshaw. T. 9216. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 2015, p. 296-316. DOI : 10.1007/978-3-662-48000-7_15.

[Lys+99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai et Stefan Wolf. « Pseudonym Systems ». In : *SAC 1999 : 6th Annual International Workshop on Selected Areas in Cryptography*. Sous la dir. d'Howard M. Heys et Carlisle M. Adams. T. 1758. Lecture Notes in Computer Science. Kingston, Ontario, Canada : Springer, Heidelberg, Germany, août 1999, p. 184-199. DOI : 10.1007/3-540-46513-8_14.

[Mil04] Victor S. Miller. « The Weil Pairing, and Its Efficient Calculation ». In : *Journal of Cryptology* 17.4 (sept. 2004), p. 235-261. DOI : 10.1007/s00145-004-0315-8.

[Mil86] Victor S. Miller. « Use of Elliptic Curves in Cryptography ». In : *Advances in Cryptology – CRYPTO'85*. Sous la dir. d'Hugh C. Williams. T. 218. Lecture Notes in Computer Science. Santa Barbara, CA, USA : Springer, Heidelberg, Germany, août 1986, p. 417-426. DOI : 10.1007/3-540-39799-X_31.

[MNT01] Atsuko Miyaji, Masaki Nakabayashi et Shunzo Takano. « Characterization of Elliptic Curve Traces under FR-Reduction ». In : *ICISC 00 : 3rd International Conference on Information Security and Cryptology*. Sous la dir. de Dongho Won. T. 2015. Lecture Notes in Computer Science. Seoul, Korea : Springer, Heidelberg, Germany, déc. 2001, p. 90-108.

[Moe+06] Bodo Moeller, Nelson Bolyard, Vipul Gupta, Simon Blake-Wilson et Chris Hawk. *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*. RFC 4492. Mai 2006. doi : 10.17487/RFC4492. url : https://rfc-editor.org/rfc/rfc4492.txt.

[MVO91] Alfred Menezes, Scott A. Vanstone et Tatsuaki Okamoto. « Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field ». In : *23rd Annual ACM Symposium on Theory of Computing*. New Orleans, LA, USA : ACM Press, mai 1991, p. 80-89. doi : 10.1145/103418.103434.

[MvV97] Alfred J. Menezes, Paul C. van Oorschot et Scott A. Vanstone. *Handbook of Applied Cryptography*. The CRC Press series on discrete mathematics and its applications. 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA : CRC Press, 1997, p. xxviii + 780. isbn : 0-8493-8523-7.

[NO14] Phong Q. Nguyen et Elisabeth Oswald, éd. *Advances in Cryptology – EUROCRYPT 2014*. T. 8441. Lecture Notes in Computer Science. Copenhagen, Denmark : Springer, Heidelberg, Germany, mai 2014.

[PH78] Stephen Pohlig et Martin Hellman. « An improved algorithm for computing logarithms over GF (p) and its cryptographic significance (corresp.) » In : *IEEE Transactions on information Theory* 24.1 (1978), p. 106-110.

[Pie16] Cécile Pierrot. « Le logarithme discret dans les corps finis. (Discrete logarithm in finite fields) ». Thèse de doct. Pierre et Marie Curie University, Paris, France, 2016. url : https://tel.archives-ouvertes.fr/tel-01633583.

[Pol78] John M Pollard. « Monte Carlo methods for index computation (mod p) ». In : *Mathematics of computation* 32.143 (1978), p. 918-924.

[PS15] David Pointcheval et Olivier Sanders. *Short Randomizable Signatures*. Cryptology ePrint Archive, Report 2015/525. https://eprint.iacr.org/2015/525. 2015.

[PS16] David Pointcheval et Olivier Sanders. « Short Randomizable Signatures ». In : *Topics in Cryptology – CT-RSA 2016*. Sous la dir. de Kazue Sako. T. 9610. Lecture Notes in Computer Science. San Francisco, CA, USA : Springer, Heidelberg, Germany, fév. 2016, p. 111-126. doi : 10.1007/978-3-319-29485-8_7.

[San15] Olivier Sanders. « Conception et Optimisation de Mécanismes Cryptographiques Anonymes. (Design and Improvements of Anonymous Cryptographic Primitives) ». Thèse de doct. École Normale Supérieure, Paris, France, 2015. url : https://tel.archives-ouvertes.fr/tel-01235213.

[SCA06] Michael Scott, Neil Costigan et Wesam Abdulwahab. « Implementing Cryptographic Pairings on Smartcards ». In : *Cryptographic Hardware and Embedded Systems – CHES 2006*. Sous la dir. de Louis Goubin et Mitsuru Matsui. T. 4249. Lecture Notes in Computer Science. Yokohama, Japan : Springer, Heidelberg, Germany, oct. 2006, p. 134-147. doi : 10.1007/11894063_11.

[Sch80] Jacob T Schwartz. « Fast probabilistic algorithms for verification of polynomial identities ». In : *Journal of the ACM (JACM)* 27.4 (1980), p. 701-717.

[Sha71] Daniel Shanks. « Class number, a theory of factorization, and genera ». In : sous la dir. de Donald J. Lewis. T. XX. Proceedings of Symposia in Pure Mathematics. AMS, 1971. isbn : 978-0-8218-1420-8. url : https://doi.org/10.1090/pspum/020.

[Sho97] Victor Shoup. « Lower Bounds for Discrete Logarithms and Related Problems ». In : *Advances in Cryptology – EUROCRYPT'97*. Sous la dir. de Walter Fumy. T. 1233. Lecture Notes in Computer Science. Konstanz, Germany : Springer, Heidelberg, Germany, mai 1997, p. 256-266. doi : 10.1007/3-540-69053-0_18.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*. T. 106. Springer Science & Business Media, 2009.

[Sma18] Nigel P Smart. *Algorithms, Key Size and Protocols Report, ECRYPT - CSA*. 2018. url : www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf.

[SS16] Palash Sarkar et Shashank Singh. « New Complexity Trade-Offs for the (Multiple) Number Field Sieve Algorithm in Non-Prime Fields ». In : *Advances in Cryptology – EUROCRYPT 2016, Part I*. Sous la dir. de Marc Fischlin et Jean-Sébastien Coron. T. 9665. Lecture Notes in Computer Science. Vienna, Austria : Springer, Heidelberg, Germany, mai 2016, p. 429-458. DOI : 10.1007/978-3-662-49890-3_17.

[Sut12] Andrew V Sutherland. « Accelerating the CM method ». In : *LMS Journal of Computation and Mathematics* 15 (2012), p. 172-204. URL : https://math.mit.edu/~drew/classpoly_v1.0.2.tar.

[TCG15] TCG. 2015. URL : https://trustedcomputinggroup.org/authentication.

[Ver01] Eric R. Verheul. « Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems ». In : *Advances in Cryptology – EUROCRYPT 2001*. Sous la dir. de Birgit Pfitzmann. T. 2045. Lecture Notes in Computer Science. Innsbruck, Austria : Springer, Heidelberg, Germany, mai 2001, p. 195-210. DOI : 10.1007/3-540-44987-6_13.

[Ver10] Frederik Vercauteren. « Optimal pairings ». In : *IEEE Trans. Inf. Theory* 56.1 (2010), p. 455-461. DOI : 10.1109/TIT.2009.2034881. URL : https://doi.org/10.1109/TIT.2009.2034881.

[Was08] Lawrence C Washington. *Elliptic curves : number theory and cryptography*. CRC press, 2008.

[Wat69] William C Waterhouse. « Abelian varieties over finite fields ». In : *Annales scientifiques de l'École normale supérieure*. T. 2. 4. 1969, p. 521-560.

[Zip79] Richard Zippel. « Probabilistic algorithms for sparse polynomials ». In : *International symposium on symbolic and algebraic manipulation*. Springer. 1979, p. 216-226.

UNIVERSITÉ DE RENNES 1

Titre : Conception de courbes elliptiques et applications

Mot clés : Courbe elliptique, couplage, signature de groupe

Résumé : Le thème de la sécurité de l'information est prédominant dans nos vies actuelles. En particulier, les utilisateurs de service, plus précisément en ligne, souhaitent que leurs données à caractère personnel soient traitées à des fins légitimes et avec leur consentement. Cela incite donc à concevoir des systèmes se pliant à de telles exigences. La cryptographie dispose de solutions puissantes pour satisfaire ce besoin de protéger la vie privée mais ces derniers nécessitent des outils mathématiques avancés.

Dans cette thèse, nous abordons l'un de ces outils : le couplage sur les courbes elliptiques. Nous divergeons de l'approche générale, celle de prendre une courbe déjà établie, standardisée, quel que soit le protocole cryptographique, et proposons des courbes conçues pour optimiser les performances d'une famille spécifique d'algorithmes cryptographiques. Les courbes proposées dans cette thèse ont des opérations dans le premier groupe du couplage plus performantes, comparées aux courbes de la littérature.

Nous donnons ensuite un schéma de signature de groupe, primitive déployée permettant d'assurer l'anonymat de ses utilisateurs au sein d'un groupe, conçu grâce aux couplages sur courbes elliptiques. Cette signature de groupe est compétitive face à l'état de l'art, tirant au maximum parti du couplage afin d'éviter d'utiliser des constructions lourdes, comme les preuves sans divulgation de connaissance, et de pâtir des limites associées, tant en termes de performances que de sécurité.

Title: Elliptic curve design and applications

Keywords: Elliptic curve, pairing, group signature

Abstract: In our day to day life, information security is a predominant topic. More specifically, (online) users expect to be asked for consent and that service providers handle their personal data with care and integrity. This urges the design of systems enforcing such expectations. The field of cryptography provides such powerful privacy-preserving tools.

In this thesis, we consider one of those tools: pairings over elliptic curves. We strongly diverge from the general approach, *i.e.* taking already standardized curves regardless of the cryptographic protocol, and suggest curves satisfying chosen criteria. The given curves in this thesis have more efficient operations in the first pairing group than the curves from the literature.

We follow by giving a group signature scheme, a primitive enabling the anonymity of its users among the group they belong to, designed using pairing over elliptic curves. This group signature is efficient when compared to the state-of-the-art, thanks to the very nice interaction between two randomizable signature schemes, allowing us to get rid of costly zero-knowledge proofs.