Courbes Algébriques et Cryptologie

Andreas Enge

Courbes Algébriques et Cryptologie

Habilitation à diriger des recherches

Andreas Enge

Courbes Algébriques et Cryptologie

Habilitation à diriger des recherches

Université Paris 7 Denis Diderot

(spécialité mathématiques)

par

Andreas Enge

Soutenue le 7 décembre 2007 devant le jury composé de

Rapporteurs	Jean-Marc Couveignes
	Gerhard Frey
	Jean-François Mestre
Examinateurs	Karim Belabas
	Reinhard Schertz
	Brigitte Vallée

Die Liebe hört niemals auf,
wo doch die Erkenntnis aufhören wird.

À mon père.

Table des matières

Avant-propos

Natur und Kunst, sie scheinen sich zu fliehen
Und haben sich, eh' man es denkt, gefunden.
— Gœthe

Ce mémoire d'habilitation est pour moi l'occasion de faire le point sur les sept années de recherche après ma thèse de doctorat. J'ai effectué ces recherches au Laboratoire d'informatique de l'École polytechnique au sein du projet TANC de l'INRIA, d'abord pendant un séjour postdoctoral, puis en tant que chargé de recherche de l'INRIA Futurs. La dénomination du projet, « Théorie algorithmique des nombres pour la cryptologie », caractérise très bien en quelques mots ma propre démarche : il s'agit de la théorie des nombres, l'un des domaines les plus anciens des mathématiques, appliquée à la cryptologie, cette discipline à l'intersection de l'informatique, des mathématiques et de l'ingénierie qui a connu un grand essor avec l'avènement des réseaux de communication numériques modernes. Et cette application se fait à travers l'algorithmique sous toutes ses facettes. Pour moi, il s'agit d'abord de la conception d'algorithmes optimaux au sens de la théorie de la complexité ; il en résulte comme l'un des fils conducteurs des trois premiers chapitres de ce mémoire la quête d'algorithmes quasi-linéaires, c'est-à-dire linéaires mis à part des facteurs logarithmiques, en la taille de leur sortie. Ces algorithmes sont complétés et validés par des implantations soignées, en faisant attention aux détails et astuces qui ne changent que la constante de la complexité, mais qui feront une grande différence en pratique ; en témoignent de nombreux records de calcul, dont les détails sont également décrits dans ce mémoire. Dans l'idéal, on arrive ainsi à concilier la théorie et la pratique, et dissiper un peu plus le mythe originel de l'inefficacité pratique des algorithmes asymptotiquement rapides. Pour les algorithmes relativement complexes en théorie des nombres, nous sommes dans la situation intéressante que la technologie actuelle nous permet à peine d'atteindre ce point.

Mais cet ordre logique entre théorie et pratique n'est en aucun cas un ordre chronologique ou hiérarchique ; au contraire, la théorie algorithmique des nombres nécessite de passer sans cesse de l'une à l'autre. Nombreux sont ainsi les théorèmes de ce mémoire inspirés par des expériences numériques, et les algorithmes de faible complexité motivés

par un goulot d'étranglement dans l'implantation. Mon insertion dans un laboratoire d'informatique a ainsi contribué à un certain changement dans mes perspectives et mes méthodes, la notion de calcul, si chère aux théoriciens des nombres de Gauß à Weber bien avant l'arrivée de l'ordinateur, étant devenue de plus en plus importante.

L'un des domaines de la cryptologie moderne qui est le plus lié à la théorie des nombres est formé par les cryptosystèmes à clef publique fondés sur les courbes algébriques. Mes contributions de ces dernières années concernent essentiellement la multiplication complexe des courbes elliptiques, qui sera discutée au cours des trois premiers chapitres de ce mémoire. Elle trouve des applications dans les preuves de primalité ainsi que dans la recherche de courbes elliptiques susceptibles de fournir des cryptosystèmes sûrs. Dans le premier chapitre, je détaille mes travaux sur la multiplication complexe proprement dite, culminant dans le meilleur algorithme connu à ce jour, et ce d'un point de vue théorique aussi bien que pratique, pour calculer les corps de classes de corps quadratiques imaginaires et les courbes elliptiques qu'ils paramètrent. Le deuxième chapitre en donne une application à travers la cryptographie fondée sur les couplages dans les courbes algébriques ; en même temps, il marque pour moi une incursion dans le domaine de la sécurité prouvée. Dans le troisième chapitre, je traite les polynômes modulaires et donne encore un algorithme optimal pour les calculer ; ces polynômes sont des ingrédients incontournables dans beaucoup d'algorithmes liés à la multiplication complexe.

Le quatrième chapitre traite un sujet assez différent, à savoir les logarithmes discrets dans les courbes algébriques, notamment de genre élevé. Tandis que les autres chapitres portent plus sur les aspects constructifs de la cryptographie, il s'agit ici de cryptanalyse, c'est-à-dire d'attaques de cryptosystèmes. Ce chapitre est dans la continuité de mes travaux de thèse de doctorat, qui ont eu pour sujet notamment le premier algorithme prouvé de complexité sous-exponentielle en $L(1/2)$ pour calculer des logarithmes discrets dans les courbes hyperelliptiques. La nouvelle contribution décrite dans ce mémoire fournit un algorithme de meilleure complexité en $L(1/3)$ pour une autre classe de courbes.

L'habilitation est censée habiliter à « diriger des recherches » en France ; en Allemagne, elle atteste plutôt la « *facultas docendi* », la qualification d'enseigner. Dans cet esprit, je me suis efforcé de fournir un document qui donne non seulement un résumé de mes propres travaux, mais qui puisse servir à la fois comme point de référence sur l'état de l'art dans les domaines décrits. Ainsi, je consacre une grande part de ce mémoire à la théorie et au contexte dans lesquels s'insèrent mes résultats. J'ai souhaité qu'il puisse être lu avec profit et plaisir sans trop de connaissances préalables, notamment par des étudiants de master ou en début de thèse ; que le lecteur averti me pardonne certaines imprécisions en résultant.

Palaiseau, le 2 septembre 2007.

Andreas Enge

Remerciements

Une habilitation est le point culminant de plusieurs années de recherche, et l'occasion de tourner le regard en arrière et de remercier de quelques mots tous ceux qui y ont contribué directement ou indirectement, et qui m'ont aidé à mener à bien cette aventure.

Les rapporteurs du présent mémoire ont grâcieusement accepté ce travail supplémentaire malgré des agendas bien remplis, et je les en remercie vivement. Pour Gerhard Frey c'est déjà la deuxième fois, après avoir été rapporteur de mon mémoire de thèse. Jean-Marc Couveignes, à travers ses algorithmes p-adiques et nos discussions à La Bussière, m'a inspiré a analyser plus finement les algorithmes fondés sur des approximations flottantes. L'aide de Jean-François Mestre, qui a porté mon habilitation devant les commissions de l'Université Paris 7, et son précieux conseil m'ont été indispensables.

Je remercie également les examinateurs qui sont venus à la soutenance. Reinhard Schertz m'accompagne depuis mes premières années d'études, pendant lesquelles il m'a enthousiasmé pour les courbes elliptiques ; ses cours finement ciselés se sont transformés en livres manuscrits dans mes étagères. Il m'a donné ma première occasion d'enseigner en deuxième année d'études, quand il m'a confié un groupe de travaux dirigés en intégration de Lebesgue et analyse complexe. Je ne lui serai jamais assez reconnaissant pour m'avoir aidé à faire une licence en France ; sans son intervention, j'aurais dû capituler devant les obstacles bureaucratiques. Les discussions avec Karim Belabas sont toujours enrichissantes, et ses connaissances en théorie algorithmique des nombres une incitation continuelle à apprendre plus. Brigitte Vallée était rapporteur de mon dossier lors de ma candidature pour mon premier poste permanent ; je me rappelle toujours de la bonne ambiance dans laquelle elle a su mener l'entretien devant le plus grand jury de ma vie. Je remercie également Antoine Chambert-Loir, qui était prêt à participer au jury, mais qui était malheureusement empêché lors de la soutenance.

Fabien Laguillaumie m'a été d'une aide précieuse lors de la rédaction, que j'ai finalement entamée grâce à nos discussions à Eurocrypt 2007. Je suis dans sa dette pour sa relecture très soigneuse du mémoire, source de discussions sur de nombreux points fins du français.

Je remercie Michèle Wasse pour son aide compétente et de bonne volonté dans les démarches administratives ; sans son concours efficace, je n'aurais pu soutenir en temps.

Les membres du projet TANC ont énormément contribué à rendre ces dernières années de recherche agréables et enrichissantes. Avant tout, François Morain m'a aidé à franchir le pas des mathématiques pures vers l'informatique et les implantations efficaces.

Je le remercie également pour les pauses café dans le vieux style des projets INRIA, remplies de science, d'échanges et de débats. La collaboration avec les autres chercheurs et thésards du projet, notamment Pierrick Gaudry, Nicolas Gürel et Régis Dupont, a également toujours été caractérisée par la confiance et la bonne humeur ; j'en garde de bons souvenirs.

Je suis reconnaissant à Komei Fukuda, coauteur de ma première publication scientifique. Le stage que j'ai effectué sous sa direction à Zürich a confirmé mon goût pour la recherche, et son éthique scientifique reste un modèle pour moi.

L'enseignement a contribué à rendre ces dernières années diversifiées et divertissantes. Mes collègues du département d'informatique, en particulier Philippe Chassignet, sont à remercier pour leur aidé à assimiler les particularités de l'enseignement à l'École polytechnique. Je suis reconnaissant envers Dieter Jungnickel, mon directeur de thèse, pour son amitié et son soutien perpétuel ; je lui dois mon premier poste d'enseignant, et mes premières expériences de cours devant un amphi rempli d'étudiants.

Finalement, mes parents et ma sœur Astrid sont toujours là quand j'ai besoin d'eux. Si mes parents ne m'avaient pas convaincu d'apprendre le français au lieu du grec ancien, ce mémoire n'aurait pu être écrit.

1 Multiplication complexe de courbes elliptiques

La multiplication algébrique est beaucoup plus simple que la numérique ; car pour multiplier une grandeur algébrique par une autre, il ne s'agit que d'écrire ces quantités les unes à côté des autres sans aucun signe.
— Encyclopédie de Diderot et d'Alembert

La théorie de la multiplication complexe des courbes elliptiques a été développée au tournant du XXème siècle ; elle fait le lien entre plusieurs domaines de l'algèbre :

- la théorie des corps de classes des corps quadratiques imaginaires ;
- les fonctions modulaires, dont les valeurs dites *singulières* engendrent ces corps de classes ;
- les anneaux des endomorphismes de courbes elliptiques sur les nombres complexes dites *à multiplication complexe* ; ces anneaux sont des ordres dans des corps quadratiques imaginaires, et leurs groupes de classes sont les groupes de Galois des corps de classes.

Comme par magie, ces différents ingrédients tombent en place pour non seulement former une théorie riche et élégante, mais également pour fournir des algorithmes explicites. Déjà dans son livre [180] de 1908, Weber donne de nombreux exemples obtenus astucieusement à la main. Évidemment, l'ordinateur nous permet aujourd'hui d'aller bien plus loin, et une grande partie de mon travail des dernières années a été consacrée à obtenir les meilleurs algorithmes possibles, complétés d'implantations efficaces.

Le lien entre la multiplication complexe et les courbes elliptiques sur les corps finis a été établi dans la première moitié du XXème siècle, d'abord par Hasse, puis par Deuring. Hasse dans [107] déduit le cardinal d'une courbe elliptique définie sur un corps fini de l'existence d'un endomorphisme séparable qui a les points rationnels comme noyau. (Un cas particulier de ce théorème se trouve d'ailleurs déjà dans le journal de Gauß.) Deuring dans [42] examine la relation précise entre les endomorphismes d'une courbe elliptique

en caractéristique 0 et ceux de sa réduction modulo un idéal premier.

L'application principale de la multiplication complexe aujourd'hui est la recherche de courbes elliptiques sur un corps fini avec un cardinal connu d'avance. Idéalement, on souhaiterait fixer à la fois le corps fini et le cardinal et construire la courbe elliptique correspondante si elle existe. Hélas, même si on peut formuler un algorithme pour résoudre ce problème, il est de complexité exponentielle et ainsi impraticable. Mais en relaxant les contraintes, par exemple en fixant le corps fini et en se contentant de courbes elliptiques avec un cardinal dans un ensemble suffisamment large, les algorithmes de la multiplication complexe permettent de résoudre efficacement des problèmes tels que la recherche d'une courbe elliptique pour créer un cryptosystème à clef publique, les preuves de primalité ou encore la création de courbes elliptiques pour des cryptosystèmes fondés sur l'identité, un sujet qui sera abordé au chapitre 2.

1.1 Théorie et applications cryptographiques

La théorie de la multiplication complexe est exposée dans [180, 42] ; un traitement élémentaire moderne est donné dans [40]. Je me contente ici d'un survol qui résume les propriétés dont nous aurons besoin dans la suite.

1.1.1 Formes et corps quadratiques

En s'intéressant aux corps quadratiques, on ne peut contourner les formes quadratiques, introduites par Lagrange dans [98] et étudiées par Gauß bien avant la formalisation moderne de l'algèbre.

Définition 1.1 Une *forme quadratique primitive définie positive* est un polynôme quadratique $Q = [A, B, C] := AX^2 + BX + C$ tel que A, B, $C \in \mathbb{Z}$, $A > 0$, $\operatorname{pgcd}(A, B, C) = 1$ et de *discriminant* $D = B^2 - 4AC < 0$. De façon équivalente, on considérera la forme homogène $Q^* = Y^2\, Q(X/Y) = AX^2 + BXY + CY^2$.

La question qui a motivé Lagrange dans [98] était de savoir quels entiers pouvaient être représentés par une forme quadratique. Un entier n est dit *représenté par la forme* Q s'il existe des entiers x et y tels que $n = Q^*(x, y)$. La représentation est *propre* si x et y sont premiers entre eux. On notera que $\mathrm{Gl}_2(\mathbb{Z})$ agit par $MQ = M \circ Q = Q^*(aX + b, cX + d) = (cX + d)^2\, Q\left(\frac{aX+b}{cX+d}\right)$ pour $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ sur l'espace des formes, et que les formes dans un même orbite représentent (proprement) les mêmes entiers. Il est donc naturel de considérer les formes quadratiques modulo l'équivalence induite par $\mathrm{Gl}_2(\mathbb{Z})$. Pour des raisons qui deviendront claires dans la suite, nous préférons néanmoins l'équivalence plus fine, dite encore *propre*, donnée par les matrices dans $\mathrm{Sl}_2(\mathbb{Z})$ et introduite par Gauß.

Définition 1.2 Le *demi-plan de Poincaré* $\mathbb{H}$ est l'ensemble des nombres complexes à partie imaginaire strictement positive. Sa complétion par des *pointes* est donnée par $\mathbb{H}^* = \mathbb{H} \cup \mathbb{Q} \cup \{\infty\}$.

À une forme quadratique Q (supposée primitive et définie positive dans la suite) nous pouvons associer son unique racine $\tau = \tau(Q) = \frac{-B+\sqrt{D}}{2A}$ dans le demi-plan de Poincaré. Pour $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{Sl}_2(\mathbb{Z})$, notons la *transformation unimodulaire* associée par $M\tau = M \circ \tau = \frac{a\tau+b}{c\tau+d}$. La relation d'équivalence induite est compatible avec l'équivalence propre des formes car $MQ = (cX+d)^2 Q(MX)$ s'annule en $M^{-1}\tau$.

Proposition 1.3 *Toute forme quadratique est proprement équivalente à une unique forme appelée* réduite *telle que*

$$|B| \leqslant A \leqslant C$$

avec $B > 0$ si l'une des inégalités n'est pas stricte. Il s'ensuit que $A \leqslant \sqrt{\frac{|D|}{3}}$, et que le nombre de classes de formes h_D *pour un discriminant D est fini.*

Comme ce résultat sera utilisé à plusieurs endroits, énonçons l'algorithme qui permet de réduire effectivement une forme donnée. De façon équivalente, nous pouvons argumenter sur les racines $\tau = \frac{-B+\sqrt{D}}{2A}$. Après l'application d'une *translation* $T^b = \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix}$: $\tau \to \tau + b$ nous pouvons supposer que $-\frac{1}{2} \leqslant \Re(\tau) < \frac{1}{2}$; cette transformation réduit B modulo $2A$. Si alors $|\tau| < 1$, l'application de la *réflexion* $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} : \tau \to \frac{-1}{\tau}$ résulte en $|\tau| > 1$; elle échange les rôles de A et C et remplace B par son négatif. Comme $|B|$ décroît à chaque étape, le processus termine. Une analyse plus fine des cas de bord montre qu'on peut obtenir τ dans le *domaine fondamental* pour $\mathrm{Sl}_2(\mathbb{Z})$ donné par

$$\mathcal{F} = \left\{ \tau \in \mathbb{H} : -\frac{1}{2} \leqslant \Re(\tau) < \frac{1}{2}, |\tau| \geqslant 1;\ \text{et } \Re(\tau) \leqslant 0 \text{ si } |\tau| = 1 \right\}.$$

Passons maintenant au langage moderne de la théorie des nombres et introduisons les ordres dans les corps quadratiques imaginaires.

Définition 1.4 Soit $D = f^2\Delta < 0$ un discriminant quadratique avec $f \geqslant 1$ maximal tel que $\Delta \equiv 1$ ou $0 \pmod 4$. Alors Δ est le *discriminant fondamental* et f le *conducteur* associé à D. Notons $K = \mathbb{Q}(\sqrt{\Delta})$, $\omega = \frac{\Delta+\sqrt{\Delta}}{2}$ un élément générateur de l'ordre maximal $\mathcal{O}_\Delta$ de K et $\mathcal{O}_D = [1, f\omega]_\mathbb{Z}$. Le *groupe de classes* Cl_D est le groupe abélien fini des idéaux fractionnaires propres de $\mathcal{O}_D$ modulo idéaux principaux, ou, de façon équivalente, les idéaux fractionnaires de $\mathcal{O}_\Delta$ premier avec f modulo idéaux principaux ; son cardinal est le *nombre de classes* h_D.

La définition du nombre de classes h_D coïncide en fait avec celle du nombre de classes de formes de la proposition 1.3. Associons à un idéal propre $\mathfrak{a} = (\omega_1, \omega_2)$ son *quotient de base* $\tau = \frac{\omega_2}{\omega_1}$ de façon à ce que $\tau \in \mathbb{H}$, et à τ la forme quadratique Q dont il est racine. La base de $\mathfrak{a}$ n'est définie qu'à transformation unimodulaire $(\omega_1, \omega_2) \mapsto (c\omega_2 + d\omega_1, a\omega_2 + b\omega_1)$ pour $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{Sl}_2(\mathbb{Z})$ près ; ainsi, Q n'est définie qu'à équivalence propre près.

Par contre, le passage au quotient de base assure que τ et donc Q ne changent pas si $\mathfrak{a}$ est multiplié par un idéal principal. Ainsi, nous obtenons une bijection entre le groupe de classes d'idéaux et l'ensemble des formes réduites, ce qui induit une loi de groupe sur les classes de formes. Notons qu'il est possible de définir indépendamment cette loi par la *composition* de formes quadratiques.

Notons encore que $Q = [A, B, C]$ représente un entier n si et seulement s'il existe un élément $\alpha = xA - y\frac{-B+\sqrt{D}}{2}$ dans l'idéal associé $\mathfrak{a} = \left(A, \frac{-B+\sqrt{D}}{2}\right)$ de norme A tel que $n = \frac{\mathrm{N}(\alpha)}{A} = \mathrm{N}\left(\frac{\alpha}{A}\mathfrak{a}\right)$; autrement dit, si et seulement si la classe de $\mathfrak{a}$ contient un idéal de norme n.

1.1.2 Fonctions modulaires

Dans la section précédente, nous avons introduit les actions de $\mathrm{Sl}_2(\mathbb{Z})$ sur les formes quadratiques et le demi-plan de Poincaré, et nous avons donné un domaine fondamental pour la dernière (voir le paragraphe suivant la proposition 1.3). Les fonctions modulaires apparaissent naturellement comme les fonctions méromorphes invariantes sous $\mathrm{Sl}_2(\mathbb{Z})$, ou plus précisément sur la surface de Riemann obtenue en compactifiant le quotient du demi-plan de Poincaré par l'action de $\mathrm{Sl}_2(\mathbb{Z})$.

Pour obtenir des invariants de classes à la section 1.2 ainsi que pour les équations modulaires du chapitre 3, nous aurons besoin de fonctions modulaires plus générales, obtenues pour des sous-groupes de $\mathrm{Sl}_2(\mathbb{Z})$.

Définition 1.5 Notons $\Gamma = \mathrm{Sl}_2(\mathbb{Z})$ le *groupe modulaire*. Pour $N \in \mathbb{N}$, soit $\Gamma(N) = \left\{\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \pmod{N}\right\}$, où l'équivalence est comprise entrée par entrée, le *sous-groupe principal de congruences de niveau* N. Tout groupe Γ' compris entre $\Gamma(N)$ et Γ est appelé un *sous-groupe de congruences de niveau* N.

L'exemple de sous-groupe de congruences le plus important est donné par $\Gamma^0(N)$, l'ensemble de toutes les matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ telles que $N|b$.

Définition 1.6 Soit Γ' un sous-groupe de congruences de niveau N. Une fonction $f : \mathbb{H} \to \mathbb{C}$ est appelée *modulaire pour* Γ' si

1. $f(z)$ est méromorphe pour $z \in \mathbb{H}$;
2. f est invariante sous Γ' :

$$f(Mz) = f\left(\frac{az+b}{cz+d}\right) = f(z) \text{ pour } M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma' ;$$

3. $f(z)$ est méromorphe à l'infini, c'est-à-dire qu'elle admet une transformée de Fourier sous la forme d'une série de Laurent en la variable $q^{1/N} = e^{2\pi i z/N}$:

$$f(z) = \sum_{\nu=\nu_0}^{\infty} c_\nu q^{\nu/N} \text{ avec } \nu_0 \in \mathbb{Z} \text{ et } c_\nu \in \mathbb{C} ;$$

4. f est méromorphe dans les autres *pointes*, c'est-à-dire pour tout $M \in \Gamma$, la fonction transformée $f(Mz)$ est méromorphe à l'infini. L'indice de Γ' dans Γ étant fini, il s'agit en fait d'un nombre fini de conditions.

Ainsi, f peut être interprétée comme une fonction $\mathbb{H}^* \to \mathbb{C} \cup \{\infty\}$. Les fonctions modulaires pour Γ' forment un corps noté $\mathbb{C}_{\Gamma'}$.

Le corps $\mathbb{C}_\Gamma$ des fonctions modulaires pour $\mathrm{Sl}_2(\mathbb{Z})$ est en fait un corps de fonctions rationnel, et tout $\mathbb{C}_{\Gamma'}$ en est une extension algébrique ; autrement dit, c'est le corps de fonctions d'une courbe algébrique appelée *modulaire*. Un élément générateur de $\mathbb{C}_\Gamma$ sur $\mathbb{C}$ est donné par la fonction j, qui peut être définie comme suit. Soient

$$G_r(z) = \sum{}' \frac{1}{(mz+n)^r} \tag{1.1}$$

les *séries d'Eisenstein de poids* r, ou $\sum'$ indique que la somme est à prendre sur tous les couples $(m,n) \in \mathbb{Z}^2$ différents de $(0,0)$. Définissons $g_2 = 60G_4$ et $g_3 = 140G_6$; alors,

$$j = 1728\,\frac{g_2^3}{g_2^3 - 27g_3^2}. \tag{1.2}$$

Son développement en série de Fourier est donné par

$$\begin{aligned} j =& q^{-1} + 744 + 196884q + 21493760q^2 + 864299970q^3 + 20245856256q^4 \\ &+ 333202640600q^5 + 4252023300096q^6 + \cdots ; \end{aligned} \tag{1.3}$$

on en déduit que j a un pôle simple à l'infini, et prend donc chaque valeur dans $\mathbb{C} \cup \infty$ exactement une fois, ce qui est la clef pour la démonstration de $\mathbb{C}_\Gamma = \mathbb{C}(j)$.

En passant par la fonction η de Dedekind, nous obtenons une expression pour j qui se prête plus pour les calculs.

Définition 1.7 La *fonction η de Dedekind* est donnée par

$$\eta(z) = q^{1/24} \prod_{n \geqslant 1} (1-q^n) = q^{1/24} \left(1 + \sum_{n=1}^{\infty} (-1)^n \left(q^{n(3n-1)/2} + q^{n(3n+1)/2} \right) \right) ; \tag{1.4}$$

son développement en série découle du *théorème des nombres pentagonaux* d'Euler [72].

Soient les *fonctions de Weber* définies comme dans [180, §§34,54] par

$$\begin{aligned} &f(z) = \zeta_{48}^{-1}\,\frac{\eta((z+1)/2)}{\eta(z)},\ f_1(z) = \frac{\eta(z/2)}{\eta(z)},\ f_2(z) = \sqrt{2}\,\frac{\eta(2z)}{\eta(z)}, \\ &\gamma_2 = \frac{f^{24}-16}{f^8} \text{ et } \gamma_3 = \frac{(f^{24}+8)(f_1^8 - f_2^8)}{f^8} \end{aligned} \tag{1.5}$$

avec $\zeta_{48} = e^{2\pi i/48}$.

Weber démontre dans [180, §54] que

$$j = \gamma_2^3 = \gamma_3^2 + 1728.$$

Nous allons revenir sur l'utilité calculatoire de ces formules à la section 1.5.

L'invariance sous transformations modulaires à la fois du groupe de classes d'un corps quadratique imaginaire et des fonctions modulaires suggère de considérer l'évaluation des fonctions en les classes.

Définition et proposition 1.8 Soit un idéal propre d'un ordre quadratique imaginaire, de quotient de base τ . La valeur d'une fonction modulaire en τ est appelée *valeur singulière*. Si la fonction est modulaire pour Γ, alors la valeur singulière ne dépend que de la classe de l'idéal.

1.1.3 Corps de classes

La théorie des corps de classes traite de la classification des groupes de Galois qui apparaissent pour les extensions d'un corps donné, en utilisant uniquement des données intrinsèques au corps de base. Se limitant aux extensions abéliennes, elle donne des réponses partielles au problème de Galois inverse, à savoir quels groupes peuvent être réalisés comme groupes de Galois. Le résultat le plus connu est le théorème de Kronecker–Weber qui dit que toute extension abélienne des rationnels est contenue dans un corps cyclotomique.

Dans la suite, nous nous intéresserons surtout aux corps de classes de Hilbert et aux corps de classes d'anneaux.

Définition 1.9 Le *corps de classes de Hilbert* d'un corps de nombres K est son extension maximale abélienne et non ramifiée ; son groupe de Galois est isomorphe au groupe de classes de K. Pour un ordre $\mathcal{O}$ de K, le *corps de classes d'anneaux* associé est l'extension abélienne de K dont le groupe de Galois est isomorphe au groupe de classes de $\mathcal{O}$.

L'existence de ce corps de classes a été conjecturée par Hilbert et Weber à la fin du XXème siècle ; elle a été démontrée en 1907 par Furtwängler [80]. Dans le cas particulier des corps quadratiques imaginaires, Weber démontre le résultat suivant dans [180, §§120–124].

Théorème 1.10 (Premier théorème principal de la multiplication complexe) *Soit $\mathcal{O}$ l'ordre de discriminant D dans un corps quadratique imaginaire K. Soit $\mathfrak{k}_1, \ldots, \mathfrak{k}_{h_D}$ un système de représentants du groupe de classes Cl_D, et $\tau_1, \ldots, \tau_{h_D}$ leurs quotients de bases. Alors toute valeur singulière $j(\mathfrak{k}_i) = j(\tau_i)$ engendre le corps de classes d'anneaux associé à $\mathcal{O}$, noté K_D. Son polynôme minimal est donné par le* polynôme de classes

$$H_D(X) = \prod_{i=1}^{h_D} \big(X - j(\mathfrak{k}_i)\big) = \prod_{i=1}^{h_D} \big(X - j(\tau_i)\big),$$

qui est irréductible sur K et à coefficients dans $\mathbb{Z}$. Plus précisément, si $\mathfrak{k}_i$ est d'ordre 2, alors $j(\mathfrak{k}_i)$ est réel, sinon, les valeurs $j(\mathfrak{k}_i)$ et $j(\mathfrak{k}_i^{-1})$ sont conjuguées complexes.

Weber démontre que le groupe de Galois de H_D est isomorphe à Cl_D ; un isomorphisme naturel est donné par le symbole d'Artin, appelé ainsi en l'honneur d'Artin qui a démontré la généralisation à toutes extensions relatives abéliennes en 1927 dans sa loi de réciprocité [8].

Définition et proposition 1.11 Étant donnés une extension abélienne L/K et un idéal premier $\mathfrak{p}$ non ramifié de l'ordre maximal de K, il y a un unique élément $\sigma \in \mathrm{Gal}(L/K)$ tel que

$$\sigma(\alpha) \equiv \alpha^{\mathrm{N}(\mathfrak{p})} \pmod{\mathfrak{P}}$$

pour tout α dans l'ordre maximal de L et tout idéal $\mathfrak{P}$ au-dessus de $\mathfrak{p}$. L'élément σ est appelé *symbole d'Artin* et noté $\left(\frac{L/K}{\mathfrak{p}}\right)$.

Le symbole d'Artin définit un isomorphisme naturel entre le groupe de classes d'un ordre et le groupe de Galois du corps de classes associé. Dans le cas du théorème 1.10, on a d'après [43, §14]

$$\left(\frac{K_D/K}{\mathfrak{l}}\right)\left(j(\mathfrak{k})\right) = j\left(\mathfrak{k}\mathfrak{l}^{-1}\right)$$

pour toutes classes d'idéaux $\mathfrak{k}$ et $\mathfrak{l}$.

1.1.4 Courbes elliptiques sur $\mathbb{C}$

Sur les complexes, une courbe elliptique peut être définie comme $\mathbb{C}$ quotienté par un réseau $\Lambda = [\omega_1, \omega_2]_{\mathbb{Z}}$ de dimension 2. C'est donc naturellement un groupe pour l'addition. Le corps des fonctions sur la courbe est alors donné par les fonctions méromorphes doublement périodiques par rapport à ω_1 et à ω_2, dites encore *fonctions elliptiques*. La fonction $\wp$ de Weierstraß et sa dérivée $\wp'$, définies par rapport à Λ, engendrent l'ensemble de ces fonctions. En appliquant une homothétie complexe et le cas échéant en échangeant les rôles des deux éléments de base du réseau, on peut supposer que $\Lambda = [1, z]$ avec $z \in \mathbb{H}$. Alors, $\wp$ satisfait l'équation différentielle

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

où g_2 et g_3 sont définis comme à la section 1.1.2.

On retrouve du côté droit la forme de Weierstraß courte habituelle d'une courbe elliptique (mise à part le 4, qui s'englobe facilement dans $\wp'$) ; et la loi de groupe triviale dans $\mathbb{C}/\Lambda$ se trouve transportée dans la loi géométrique dite des cordes et des tangentes par les théorèmes d'addition pour $\wp$. On associe au réseau Λ et à cette courbe son invariant j, défini comme dans (1.2), qui classifie les réseaux à homothétie près et donc les courbes elliptiques à isomorphisme près.

Historiquement, on s'est intéressé aux *multiplicateurs* des fonctions elliptiques, qui sont des nombres complexes tels que la multiplication de l'argument par ce nombre résulte encore en une fonction elliptique. Autrement dit, on demande que $\wp(\lambda x)$, $\wp'(\lambda x)$ soient des expressions rationnelles en $\wp(x)$ et $\wp'(x)$, toujours par rapport au même réseau Λ ; ou encore, que λ induit une application rationnelle de la courbe elliptique en

elle-même. En fait, la multiplication par λ se distribuant par rapport à l'addition, il s'agit d'un homomorphisme de groupe et donc d'un endomorphisme de la courbe.

Un multiplicateur doit transformer le réseau $\Lambda = [1, z]$ en lui-même. On peut distinguer deux cas : génériquement, un réseau ne possède de multiplication que par les entiers. Le cas singulier se produit quand $z = \frac{-B+\sqrt{D}}{2A}$ est un quotient de base d'un idéal propre d'un ordre quadratique imaginaire $\mathcal{O}_D$, dans lequel cas Λ (et donc la courbe elliptique associée) admet $\mathcal{O}_D$ comme anneau de multiplicateurs. On dit alors que la courbe a « multiplication complexe par $\mathcal{O}_D$ ». Ainsi, nous obtenons le résultat suivant :

Théorème 1.12 *Sur $\mathbb{C}$, les courbes elliptiques à multiplication complexe par l'ordre $\mathcal{O}_D$ sont, à isomorphisme près, les courbes avec j-invariant parmi les $j(\mathfrak{k})$, où $\mathfrak{k}$ parcourt le groupe de classes de $\mathcal{O}$.*

1.1.5 Courbes elliptiques sur les corps finis

Les théorèmes 1.10 et 1.12 font le lien entre les anneaux d'endomorphismes des courbes elliptiques définies sur $\mathbb{C}$ et les corps de classes : pour obtenir des courbes avec des endomorphismes non triviaux, il faut chercher leurs j-invariants dans un corps de classes d'anneaux d'un corps quadratique imaginaire. Qu'en est-il des courbes elliptiques définies sur un corps fini, qui ont trouvé tant d'intérêt en cryptographie ?

Soit donc

$$E : Y^2 = X^3 + aX + b \text{ avec } a, b \in \mathbb{F}_q$$

une courbe elliptique définie sur un corps $\mathbb{F}_q$ de caractéristique différente de 2 et de 3 (ceci pour simplifier l'exposition ; autrement, il faudrait écrire la courbe sous forme de Weierstraß longue). Le *Frobenius*

$$\varphi : E(\overline{\mathbb{F}}_q) \to E(\overline{\mathbb{F}}_q),\ (x, y) \mapsto (x^q, y^q)$$

est une application rationnelle sur les points de E qui sont définis sur une clôture algébrique $\overline{\mathbb{F}}_q$. Comme la loi de groupe de E est donnée par des formules rationnelles en a et $b \in \mathbb{F}_q$ (et en les coordonnées des points à additionner), il s'agit en fait d'un homomorphisme de groupe et donc d'un endomorphisme, clairement différent d'une multiplication par un entier. Ceci montre que sur un corps fini, toute courbe a *multiplication complexe*, c'est-à-dire un anneau d'endomorphismes différent de $\mathbb{Z}$.

Le terme *multiplication complexe* est justifié car Hasse a montré en 1933 dans [107] que le Frobenius satisfait une équation quadratique

$$\varphi^2 - t\varphi + q = 0 \text{ de discriminant } D = t^2 - 4q \leqslant 0. \tag{1.6}$$

Du fait que les points $\mathbb{F}_q$-rationnels $E(\mathbb{F}_q)$ sont précisément le noyau de $\varphi - \mathrm{id}$, que φ est purement inséparable et donc $\varphi - \mathrm{id}$ séparable, il déduit que le cardinal de $E(\mathbb{F}_q)$ est donné par $\mathrm{N}(\varphi - 1) = q + 1 - t$; cela prouve la fameuse borne de Hasse

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

Hasse montre en plus dans [108, §2.3 et §3.3] qu'il y a deux possibilités pour l'anneau des endomorphismes d'une courbe elliptique sur $\mathbb{F}_q$: il peut être soit un ordre dans un corps quadratique imaginaire comme sur les complexes, dans lequel cas la courbe sera appelée *ordinaire* ; soit un ordre dans un algèbre de quaternions, dans lequel cas la courbe sera appelée *supersingulière*. Deuring précise dans [41, §4] que pour une courbe supersingulière, l'algèbre est ramifiée précisément au-dessus de l'infini et de la caractéristique p de $\mathbb{F}_q$ et que l'ordre est maximal. Dans la suite, nous nous intéresserons surtout aux courbes ordinaires, les courbes supersingulières étant peu nombreuses et complètement classifiées (cf. [42, 179]) ; elles sont toutes définies sur $\mathbb{F}_{p^2}$.

Dans [42], Deuring examine le rapport entre courbes elliptiques en caractéristique 0 et sur un corps fini.

Théorème 1.13 (Théorème de réduction et de relèvement de Deuring) *Soit E une courbe elliptique à multiplication complexe par $\mathcal{O}_D$, définie sur le corps de classes K_D, et $\mathfrak{P}$ un idéal premier de K_D. Alors la réduction de E modulo $\mathfrak{P}$ est une courbe elliptique $\overline{E}$ avec $\mathcal{O}_D \subseteq \mathrm{End}(\overline{E})$*

Soit $p\mathbb{Z} = \mathfrak{P} \cap \mathbb{Z}$. Si p est scindé dans $K = \mathbb{Q}(\sqrt{D})$, soit D' le discriminant obtenu de D en enlevant les facteurs de p du conducteur ; alors $\mathrm{End}(\overline{E}) = \mathcal{O}_{D'}$. Si p est ramifié ou inerte dans K, alors $\overline{E}$ est supersingulière.

Inversement, toute courbe définie sur un corps fini s'obtient de cette manière.

Tel qu'il est formulé, ce théorème laisse planer un doute sur le nombre de courbes avec multiplication complexe par $\mathcal{O}_D$ définies sur un corps fini. En effet, il n'est même pas clair si ce nombre est fini, car *a priori*, les courbes sur $\mathbb{C}$ avec multiplication par $\mathcal{O}_{D'}$, $\mathcal{O}_{p^2D'}$, $\mathcal{O}_{p^4D'}$, ... se réduisent toutes en une courbe avec multiplication par $\mathcal{O}_{D'}$. Hors, les polynômes de classes des sous-ordres se décomposent modulo p comme

$$H_{p^{2k}D'} \equiv (H_{D'})^{h_{p^{2k}D'}/h_{D'}} \pmod{p},$$

de sorte que les courbes à multiplication complexe par $\mathcal{O}_{p^{2k}D'}$ se réduisent modulo p dans les courbes déjà obtenues à travers $\mathcal{O}_{D'}$.

Concernant le corps de définition $\mathbb{F}_{p^m}$ de $\overline{E}$, notons que m est le degré d'inertie de $\mathfrak{p} = \mathfrak{P} \cap K$ dans K_D, et que l'isomorphisme entre Cl_D et $\mathrm{Gal}(K_D/K)$ via le symbole d'Artin implique que m est l'ordre de $\mathfrak{p}$ dans le groupe de classes.

Corollaire 1.14 *Soient $\mathbb{F}_q = \mathbb{F}_{p^m}$ un corps fini de caractéristique p et $\mathcal{O}_D$ l'ordre quadratique imaginaire de discriminant D. Des courbes elliptiques à multiplication complexe par $\mathcal{O}_D$ définies sur $\mathbb{F}_q$ existent si et seulement si $p = \mathfrak{p}\overline{\mathfrak{p}}$ est scindé dans $K = \mathbb{Q}(\sqrt{D})$, p ne divise pas le conducteur de $\mathcal{O}_D$ et l'ordre de $\mathfrak{p}$ dans Cl_D divise m. Dans ce cas, ces courbes sont au nombre de h_D, et s'obtiennent comme réduction des courbes à multiplication complexe par $\mathcal{O}_D$ définies sur K_D modulo un idéal premier de K_D au-dessus de $\mathfrak{p}$.*

1.1.6 Construction de courbes elliptiques à multiplication complexe sur un corps fini

Les sections précédentes suggèrent un algorithme conceptuellement simple pour déterminer les courbes à multiplication complexe donnée sur un corps fini.

Algorithme 1.15
ENTRÉE: *$q = p^m$ et D compatibles avec les hypothèses du corollaire 1.14*
SORTIE: *les h_D courbes à multiplication complexe par $\mathcal{O}_D$ définies sur $\mathbb{F}_q$*

1. *calculer le corps de classes d'anneaux K_D ;*
2. *exprimer les $j(\mathfrak{k}_1)$, ..., $j(\mathfrak{k}_{h_D})$ comme éléments de K_D, où les $\mathfrak{k}_i$ parcourent un système de représentants de Cl_D ;*
3. *réduire les $j(\mathfrak{k}_i)$ modulo un idéal $\mathfrak{P}$ de K_D au-dessus de p pour obtenir $\bar{\jmath}_1$, ..., $\bar{\jmath}_{h_D} \in \mathbb{F}_q$;*
4. *retourner des équations de courbes sur $\mathbb{F}_q$ avec les $\bar{\jmath}_i$ comme j-invariants.*

Il convient de préciser les différentes étapes. Pour D différent de -3 et de -4, l'étape 4 se résout en notant qu'une courbe de j-invariant $\bar{\jmath}$ est donnée par
- $E : Y^2 = X^3 + 3kX + 2k$ avec $k = \frac{\bar{\jmath}}{1728-\bar{\jmath}}$ pour $p \neq 2, 3$;
- $E : Y^2 + XY = X^3 + \bar{\jmath}^{-1}$ pour $p = 2$;
- $E : Y^2 = X^3 + \bar{\jmath}^{3^{m-1}} X^2 + 1$ pour $p = 3$.

L'approche la plus naturelle pour les trois premiers pas consiste à construire K_D comme le corps donné par le polynôme de classes H_D de sorte que les $j(\mathfrak{k}_i)$ sont tout simplement les racines de H_D. La réduction modulo $\mathfrak{P}$ se fait alors en réduisant le polynôme $H_D \in \mathbb{Z}[X]$ modulo p en un polynôme $\overline{H}_D$ défini sur $\mathbb{F}_p$, et qui se décompose sur $\mathbb{F}_p$ en h_D/m' facteurs irréductibles de degré m' où $m'|m$ est l'ordre de $\mathfrak{p}$ dans le groupe de classes. Les h_D racines de $\overline{H}_D$ sur $\mathbb{F}_q$ sont précisément les $\bar{\jmath}_i$.

Classiquement, H_D s'obtient à partir d'approximations complexes de ses racines. Si la précision de calcul est suffisamment grande, on peut arrondir les coefficients ainsi calculés vers les entiers. Clairement, il faut calculer avec au moins autant de chiffres qu'il y a dans le plus grand coefficient ; autrement dit, la précision du calcul doit être au moins la hauteur logarithmique du polynôme. Cette hauteur est examinée en plus de détails à la section 1.3. Je décris dans la section 1.5 un algorithme asymptotiquement optimal fondé sur des approximations complexes, ainsi que des alternatives p-adiques.

Le polynôme de classes H_D est sympathique car il fait le lien direct entre corps de classes et courbes elliptiques. Mais il s'avère difficile à calculer à cause de la taille de ses coefficients. En pratique, on préfère engendrer K_D par un polynôme plus petit, ce qui a pour contrepartie de rendre les étapes 2 et 3 de l'algorithme plus compliquées. Nous sommes ainsi amenés à examiner d'un côté à la section 1.2 les invariants de classes en tant que générateurs de K_D, de l'autre côté au chapitre 3 les équations qui font le lien entre deux fonctions modulaires et qui permettent de nous ramener à j.

1.1.7 Applications cryptographiques

L'application principale de la multiplication complexe en cryptographie vient du fait qu'elle permet de construire des courbes elliptiques dont le cardinal est connu d'avance. Soient $q = p^m$ et D comme dans l'algorithme 1.15. Le théorème de Hasse nous dit que le Frobenius est un élément de norme q dans $\mathcal{O}_D$; autrement dit, l'équation

$$4q = t^2 - u^2 D \tag{1.7}$$

a une solution en entiers t et u. Alors le cardinal des courbes sur $\mathbb{F}_q$ à multiplication complexe par $\mathcal{O}_D$ est donné par

$$|E(\mathbb{F}_q)| = q + 1 - t. \tag{1.8}$$

Une petite complication est introduite par le nombre de solutions à l'équation de la norme (1.7), qui est soit nul, soit, à conjugaison complexe près, égal au nombre d'unités dans $\mathcal{O}_D$ (quatre pour $D = -4$, six pour $D = -3$, deux dans tous les autres cas). Pour $|D| > 4$, par exemple, cela se traduit par l'existence de deux courbes sur $\mathbb{F}_q$ avec le même j-invariant, l'une avec $q + 1 - t$, l'autre avec $q + 1 + t$ points. Les courbes sont données par deux équations

$$E : Y^2 = X^3 + aX + b \text{ et } E' : Y^2 = X^3 + a\gamma^2 X + b\gamma^3$$

dont la deuxième est obtenue en *tordant* la première par un non-résidu quadratique $\gamma \in \mathbb{F}_q$ (en caractéristique 2, il faut utiliser un élément de trace 1 pour agir sur l'équation d'Artin–Schreier). Sur $\mathbb{F}_{q^2}$, ces deux courbes sont isomorphes ; sur $\mathbb{F}_q$, elles sont algébriquement indistinguables, et si une courbe à cardinal (1.8) est recherchée, il faut en général la considérer avec sa tordue.

La multiplication complexe peut donc servir à construire des courbes pour un cryptosystème elliptique, la contrainte principale étant que le problème du logarithme discret doit être difficile dans la courbe. À notre connaissance actuelle, c'est le cas si le corps de définition est suffisamment grand et que le cardinal de la courbe a un grand facteur premier (voir le chapitre 4). Le meilleur rapport entre qualité (sécurité) et prix (taille des clefs et complexité de l'implantation) est obtenu pour une courbe de cardinal premier. Cette application de la multiplication complexe n'est plus d'actualité ; en effet, l'algorithme de Schoof [166] pour compter le nombre de points sur une courbe elliptique quelconque et ses améliorations successives [53, 141, 39, 36, 128] ainsi que les algorithmes p-adiques en petite caractéristique [159, 137, 74, 161] sont suffisamment efficaces pour trouver aisément des courbes aléatoires au cardinal premier. L'algorithme 1.15 ne permet de traiter que des discriminants relativement petits (plus de détails sont donnés au chapitre 1.5) et crée donc des courbes avec beaucoup d'endomorphismes. Même si aucune attaque sur ces courbes n'a été trouvée à ce jour, cette particularité les rend suspectes au cryptographe, qui préférera une courbe aléatoire.

Par contre, dans le contexte de la cryptographie fondée sur les couplages, les courbes utilisables sont tellement rares qu'on ne peut les obtenir aléatoirement. On est contraint

d'utiliser des courbes spéciales, soit supersingulières, soit calculées à l'aide de la multiplication complexe. Plus de détails sont donnés au chapitre 2.

Une dernière application à évoquer ici est fournie par les preuves de primalité, utiles par exemple pour vérifier que le cardinal d'une courbe elliptique est effectivement premier ou qu'une clef RSA est effectivement le produit de deux nombres premiers. Les preuves de primalité elliptiques ont été introduites par Goldwasser et Kilian dans [96]. C'est une simple application du théorème de Hasse : pour un N à prouver premier, on suppose donnée une courbe elliptique sur $\mathbb{Z}/N\mathbb{Z}$ avec un point d'ordre premier plus grand que $(\sqrt[4]{N}+1)^2$. Si N avait un facteur premier $p \leqslant \sqrt{N}$, par le théorème de Hasse le point aurait sur la courbe réduite modulo p un ordre n'excédant pas la borne, une contradiction. (Pour rendre cette argumentation rigoureuse, il faut contourner plus soigneusement le fait que $\mathbb{Z}/N\mathbb{Z}$ n'est potentiellement pas un corps.) Il s'en déduit une approche récursive : pour prouver que N est premier, il faut exhiber une courbe modulo N avec un point d'ordre N_1 premier ; on fait de même pour N_1, et ainsi de suite. Goldwasser et Kilian ont proposé d'utiliser des courbes aléatoires et l'algorithme de Schoof pour déterminer leur cardinal. Atkin et Morain ont observé dans [9] qu'il vaut mieux se servir de la multiplication complexe. L'algorithme qui en résulte, avec toutes ses améliorations, n'est pas prouvé polynomial ; heuristiquement, il a une complexité de $O(\log^{4+\varepsilon} N)$. Notons également qu'une fois la suite de courbes obtenue, elle peut servir comme certificat qui se vérifie plus rapidement en temps $O(\log^{3+\varepsilon} N)$. Ainsi, les preuves de primalité par courbes elliptiques restent d'actualité même après AKS [4] et ses améliorations. Entre autres en utilisant les résultats des sections 1.2 et 1.4, Morain a prouvé premiers des nombres de 20 000 chiffres décimaux [144]. Pour plus de détails sur les preuves de primalité, voir le survol [143] de Morain au séminaire Bourbaki.

1.2 Fonctions de classes

Le polynôme de classes H_D défini dans le théorème 1.10 a des coefficients qui croissent très vite avec le discriminant, et se prête donc assez peu aux calculs. À titre d'exemple,

$$\begin{aligned} H_{-195}(X) = & X^4 + 11284411506057216000\, X^3 \\ & - 25349140792043819237376000\, X^2 \\ & + 10477310031960033617510400000 0\, X \\ & - 23349028549243275367258521600000 0. \end{aligned} \tag{1.9}$$

La folklore attribue ce comportement à la croissance rapide des coefficients de la série (1.3) en q pour j ; nous allons voir à la section 1.3 que c'est plutôt l'ordre du pôle de j à l'infini qui pose problème.

On préférerait engendrer les corps de classes par des éléments qui ont des polynômes minimaux plus petits, quitte à s'éloigner de j et donc des courbes elliptiques ; il se trouve que des valeurs singulières d'autres fonctions modulaires peuvent avantageusement remplacer celles de j.

Définition 1.16 Une *fonction de classes* pour le discriminant D est une fonction modulaire f telle qu'il existe un idéal propre de $\mathcal{O}_D$ de quotient de base τ avec $f(\tau) \in K_D$.

Le polynôme minimal de cette valeur, encore appelé *polynôme de classes*, est noté par $H_D[f]$.

Cette définition est inspirée par Weber, qui, dans [180, §115], appelle *invariant de classes* la valeur singulière $j(\tau)$. Dans le §125, il applique cette terminologie à d'autres fonctions modulaires et semble demander en plus que les fonctions s'expriment rationnellement en j.

1.2.1 Résultats classiques

Pour une fonction de classes f qui n'est plus modulaire pour Γ, mais seulement pour un sous-groupe, la valeur singulière d'un idéal est mal définie, car elle dépend de la base choisie. Il convient d'utiliser plutôt le formalisme des formes quadratiques, et de résoudre deux questions : comment faut-il normaliser une forme quadratique pour que sa racine donne une valeur dans le corps de classes ? Si on veut calculer son polynôme minimal, quelles sont ses conjuguées ?

Weber a donné des réponses à la première question pour les fonctions de (1.5) dans [180, §§125–131] en fixant les coefficients de la forme quadratique modulo des puissances de 2 et 3 (ce qui s'explique du fait que f^{24}, f_1^{24} et f_2^{24} sont modulaires pour les trois groupes conjugués de $\Gamma^0(2)$). Des lacunes dans ses démonstrations ont été comblées dans [17, 138, 162]. De plus, Weber calcule astucieusement des valeurs exactes de ses fonctions ; par exemple, il donne dans le §150 le polynôme minimal de $z = \frac{f(\sqrt{-41})}{\sqrt{2}} + \sqrt{2}f(\sqrt{-41})$ comme $z^4 - 5z^3 + 3z^2 + 3z + 2$. Plus de résultats dans cette direction se trouvent dans la thèse [106] de Hart, dont j'étais rapporteur.

La dérivation des conjuguées peut se faire aujourd'hui en utilisant la loi de réciprocité de Shimura [173]. Elle montre que des valeurs singulières de certaines fonctions modulaires vivent dans un corps de classes de rayons, et fait le lien entre le symbole d'Artin et l'action de matrices sur le développement en série de la fonction.

On peut alors procéder de deux manières. Premièrement, on peut attribuer à toute forme quadratique Q_i une fonction différente f_{Q_i} de sorte que $f_{Q_i}(\tau_i)$ ne dépend que de la classe de Q_i ; en faisant varier les Q_i sur un système de représentants du groupe de classes, on obtient un système complet de conjuguées. C'est l'approche choisie dans [184] pour les fonctions de Weber (sans utiliser la loi de réciprocité de Shimura, mais uniquement le comportement des fonctions sous transformations unimodulaires) et dans [94, 95], où les fonctions w_p de la section 1.2.2 sont traitées dans les cas spéciaux $p = 3$ et $p = 5$.

Schertz dans [164] procède de la manière inverse. Il fixe la fonction de classes et obtient un système de conjuguées en normalisant les formes quadratiques. Essentiellement, il traite des fonctions modulaires pour $\Gamma^0(N)$ et exige que les formes quadratiques satisfassent certaines congruences modulo N.

Définition et proposition 1.17 Étant donné un entier positif N, un *N-système* pour le discriminant D est donné par un système de représentants $Q_i = [A_i, B_i, C_i]$ du groupe de classes Cl_D tel que tous les A_i sont premiers avec N et tous les B_i sont égaux modulo $2N$.

Un tel système existe pour tous N et D et se calcule effectivement en suivant [164, proposition 3].

Cette approche est préférable en pratique : il suffit d'implanter l'évaluation d'une seule fonction modulaire, et l'adaptation des formes quadratiques se fait aisément en parallèle avec l'énumération du groupe de classes.

L'exemple $D = -195$ de (1.9) est choisi tel que les fonctions de Weber ne donnent pas d'invariants de classes ; c'est dû à $\left(\frac{D}{2}\right) = -1$. Il est possible de passer au corps de classes de $\mathcal{O}_{4D}$ de nombre de classes $3h_D$ par le cube d'une fonction de Weber (ce qui est nécessaire car $3|D$) :

$$\begin{aligned} H_{-780}[f^3] =& X^{12} - 252\,X^{11} + 2744\,X^{10} - 8368\,X^9 + 688\,X^8 + 33536\,X^7 \\ & - 28672\,X^6 - 116736\,X^5 + 328448\,X^4 - 381952\,X^3 + 231424\,X^2 \\ & - 53248\,X + 4096 \end{aligned} \tag{1.10}$$

Les coefficients sont certes plus petits que dans le polynôme pour j, mais la perte du facteur 3 pour le degré montre que la recherche de fonctions de classes alternatives reste d'intérêt.

1.2.2 Les quotients simples de η

Cette section détaille des travaux communs avec François Morain, partiellement publiés dans [64].

Les fonctions de Weber sont des quotients de deux fonctions η, dont une transformée de niveau 2. Elles se généralisent naturellement à un niveau premier ℓ quelconque en prenant une 24ème racine de la fonction $\varphi_{\left(\begin{smallmatrix}1 & 0\\ 0 & \ell\end{smallmatrix}\right)}$ de Klein [121, Abschnitt II, §16] :

Définition 1.18 Soit ℓ premier. Le *simple quotient de η de niveau ℓ* est donné par

$$w_\ell = \frac{\eta(z/\ell)}{\eta(z)}.$$

Notons $s = 12/\operatorname{pgcd}(12, \ell - 1)$ le défaut de divisibilité de $\ell - 1$ par 12.

Dans [79], Fricke constate que quand ℓ est scindé ou ramifié dans $K = \mathbb{Q}(\sqrt{D})$ pour D fondamental, w_ℓ^{2s} est une fonction de classes pour D. Parfois, il est possible d'utiliser des puissances plus basses. Gee et Stevenhagen utilisent w_3^2 dans [95] pour traiter un discriminant particulier, et Gee obtient un résultat général pour des résolvantes formées avec les conjuguées de w_5 dans [94].

Dans [64], nous donnons le résultat suivant sans preuve.

Théorème 1.19 *Soient ℓ un premier impair et D un discriminant tel que $\left(\frac{D}{\ell}\right) \neq -1$. Soient la puissance w_ℓ^e et l'entier N, un multiple de ℓ, choisis comme dans le tableau suivant.*

$\ell \bmod 12$	D	invariant	N	B
$\ell = 3$	—	w_3^{12}	3	—
	$2 \nmid D$	w_3^6	6	$B \equiv 1 \pmod 4$
1	—	w_ℓ^2	ℓ	—
5	—	w_ℓ^6	ℓ	—
	$3 \nmid D$	w_ℓ^2	3ℓ	$3 \mid B$
7	—	w_ℓ^4	ℓ	—
	$2 \nmid D$	w_ℓ^2	2ℓ	$B \equiv 1 \pmod 4$
11	—	w_ℓ^{12}	ℓ	—
	$2 \nmid D$	w_ℓ^6	2ℓ	$B \equiv 1 \pmod 4$
	$3 \nmid D$	w_ℓ^4	3ℓ	$3 \mid B$
	$\gcd(D, 6) = 1$	w_ℓ^2	6ℓ	$B \equiv 9 \pmod{12}$

Si $Q = [A, B, C]$ est une forme quadratique de discriminant D telle que $\gcd(A, N) = 1$, $B^2 \equiv D \pmod{4\ell}$ et B satisfait les congruences modulo 3 et 4 données dans le tableau, alors w_ℓ^e est une fonction de classes pour D. Le polynôme $H_D[w_\ell^e]$ est à coefficient dans l'ordre maximal de $K = \mathbb{Q}(\sqrt{D})$ et se calcule par un N-système à partir de Q.

Si de plus $\ell | D$ et que le tableau ne donne pas de contrainte pour $B \bmod 4$, alors $H_D[w_\ell^e] \in \mathbb{Z}[X]$.

Une démonstration du théorème peut être obtenue comme dans [164] et fera l'objet d'une publication ultérieure. Notons qu'en fait le polynôme de classes dépend de la valeur initiale choisie pour B ; quand $\left(\frac{D}{\ell}\right) = 1$, les deux choix de la racine mènent à deux polynômes de classes conjugués complexes.

Des expériences numériques nous ont amenés à conjecturer le résultat suivant pour $\ell = 3$, qui ne se démontre pas en utilisant uniquement [164].

Conjecture 1.20 *Les résultats du théorème 1.19 restent vrais dans les cas suivants :*

$\ell = 3$	$3 \mid D, D/3 \equiv 8 \pmod{12}$	w_3^4	$N = 9$	$9 \mid B$
	$3 \mid D, D/3 \equiv 11 \pmod{12}$	w_3^2	$N = 18$	$3 \mid B$

En poursuivant l'exemple du discriminant -195, voici quelques polynômes de classes

obtenus avec les quotients simples de η ; posons $\omega = \frac{1+\sqrt{-195}}{2}$.

$$\begin{aligned} H_{-195}[w_7^2] =& X^4 + (-475 + 34\omega)\, X^3 + (-7920 - 165\omega)\, X^2 + (-16997 - 2505\omega)\, X \\ &+ (2255 + 97\omega) \\ H_{-195}[w_{13}^2] =& X^4 + 39\, X^3 + 260\, X^2 + 507\, X + 169, \\ &\text{qui est bien réel comme prédit par la théorie.} \end{aligned}$$

Finalement, un exemple pour illustrer la conjecture :

$$H_{-39}[w_3^2] = X^4 + 729\, X^3 + 18225\, X^2 + 531441\, X + 531441.$$

1.2.3 Les quotients doubles de η

Je présente ici mes contributions de [68], obtenues avec Reinhard Schertz.

La généralisation des fonctions de Weber de la section précédente se généralise naturellement en itérant le processus du passage au quotient. En composant la procédure deux fois, on obtient les fonctions suivantes.

Définition 1.21 Soient p_1 et p_2 deux premiers non nécessairement distincts. Le *double quotient de η de niveau p_1p_2* est donné par

$$\frac{w_{p_1}(z)}{w_{p_1}\left(\frac{z}{p_2}\right)} = \frac{w_{p_2}(z)}{w_{p_2}\left(\frac{z}{p_1}\right)} = \frac{\eta\left(\frac{z}{p_1}\right)\eta\left(\frac{z}{p_2}\right)}{\eta(z)\,\eta\left(\frac{z}{p_1p_2}\right)}.$$

Désignons par $s = 24/\operatorname{pgcd}(24, (p_1-1)(p_2-1))$ le défaut de divisibilité de $(p_1-1)(p_2-1)$ par 24.

Remarquons que l'identité $ff_1f_2 = \sqrt{2}$, prouvée déjà dans [115], fournit un autre lien avec les fonctions de Weber :

$$f\left(\frac{z}{2}\right) = w_{2,2}(z).$$

Les fonctions $w^s_{p_1,p_2}$ sont modulaires pour $\Gamma^0(p_1p_2)$. Dans le cadre de la multiplication complexe, elles ont été introduites dans [163] pour obtenir des bases entières dans des corps de classes de rayons. En utilisant encore l'approche de [164] et nos résultats sur le polynôme modulaire liant $w^s_{p_1p_2}$ à j publiés dans [69] (et qui seront détaillés à la section 3.3), nous démontrons les théorèmes suivants dans [68].

Théorème 1.22 *Soient p_1 et p_2 deux premiers non nécessairement distincts, $N = p_1p_2$ et $D = f^2\Delta$ un discriminant de conducteur f. Supposons que p_1 et p_2 satisfont l'une des conditions suivantes :*

- *$p_1 \neq p_2$ et $\left(\frac{D}{p_1}\right), \left(\frac{D}{p_2}\right) \neq -1$;*
- *$p_1 = p_2 = p$, et $\left(\frac{D}{p}\right) = 1$ ou $p|f$.*

Il existe alors une forme quadratique $Q = [A, B, C]$ de discriminant D telle que $N|C$ et $\gcd(A, N) = 1$. *La fonction $w^s_{p_1,p_2}$ est une fonction de classes pour D, et le polynôme de classes associé se calcule par un N-système à partir de Q.*

De plus, les valeurs singulières sont des entiers algébriques, et hormis le cas $p_1 = p_2 = p$ et $p|f$, elles sont même des unités.

Dans les polynômes de classes pour j, le plus grand coefficient est, la plupart du temps, donné par la norme (voir la section 1.3). Notre espoir en utilisant des unités était de maîtriser la norme et d'arriver ainsi à des polynômes plus petits. Mais les quotients doubles de η possèdent un autre avantage sur les quotients simples.

Ils sont également invariants sous l'involution de Fricke–Atkin–Lehner sur $\mathbb{C}_{\Gamma^0(N)}$ donnée par $z \mapsto \frac{-N}{z}$. En ajoutant cette hypothèse au théorème 4 de [164] et en supposant que non seulement $N|C$, mais que $N = C$, nous démontrons que le polynôme de classes est en fait à coefficients dans $\mathbb{Z}$, et donnons l'action de la conjugaison complexe sur les conjuguées de la valeur singulière. Ainsi, nous déterminons explicitement le groupe de Galois du polynôme de classes sur $\mathbb{Q}$ en tant que produit semi-direct du groupe de classes de $\mathcal{O}_D$ et du groupe d'ordre deux engendré par la conjugaison complexe. Précisément, soit w la fonction de classes, et notons $\mathfrak{n}$ l'idéal associé à la forme de départ telle que $C = N$; on a donc $\mathfrak{n} = \mathfrak{p}_1\mathfrak{p}_2$ où $\mathfrak{p}_i$ est l'idéal ramifié au-dessus de p_i. Alors, les deux idéaux $\mathfrak{a}$ et $\mathfrak{n}\mathfrak{a}^{-1}$ associés à deux formes du N-système satisfont

$$\overline{w(\mathfrak{a})} = w(\mathfrak{n}\mathfrak{a}^{-1}).$$

Dans la pratique, cette identification explicite de deux conjuguées complexes permet d'économiser la moitié des calculs.

Spécialisant ce résultat au cas des quotients doubles de η, nous obtenons que le polynôme de classes est à coefficients dans $\mathbb{Z}$ si, en plus des hypothèses du théorème 1.22, l'une des conditions suivantes est satisfaite :

- $p_1 \neq p_2$ et $p_1, p_2 \nmid f$;
- $p_1 = p_2 = p \neq 2$;
- $p_1 = p_2 = 2$ et $\left(\frac{D}{2}\right) = 1$; ou
- $p_1 = p_2 = 2$, $2|f$ et $D \not\equiv 4 \pmod{32}$.

Les quotients simples de η, quant à eux, ne sont pas invariants sous l'involution de Fricke–Atkin–Lehner ; on pourrait obtenir un polynôme sur $\mathbb{Z}$ en utilisant, par exemple, la trace par rapport à cette involution, mais le polynôme de classes en résultant auraient des coefficients nettement plus grands.

Comme dans le cas des quotients simples de η, il est possible d'utiliser des puissances moindres de w_{p_1,p_2} comme fonctions de classes, moyennant des conditions arithmétiques sur D modulo $N = p_1p_2$. C'est d'un intérêt pratique car comme exposé à la section 1.3, les fonctions avec $s > 1$ résultent en des polynômes s fois plus grands. Néanmoins, nous n'avons pas recueilli les différentes conditions ; notons que les couples (p_1, p_2) avec $s = 1$ ont une densité de Dirichlet positive, ce qui permet d'en trouver pour n'importe quel discriminant.

Ainsi, nous avons exhibé la première famille infinie de fonctions de classes dont les

polynômes de classes sont à coefficients dans $\mathbb{Z}$, et de sorte que pour tout discriminant, on peut trouver une fonction adéquate dans la famille.

Par exemple, même en nous limitant au cas de $s = 1$, nous obtenons déjà de nombreux polynômes de classes pour $D = -195$:

$$\begin{aligned} H_{-195}[w_{3,13}] &= X^4 + 6\,X^3 + 11\,X^2 + 6\,X + 1 \\ H_{-195}[w_{5,7}] &= X^4 - 3\,X^3 + 10\,X^2 + 3\,X + 1 \\ H_{-195}[w_{5,13}] &= X^4 - 6\,X^3 + 7\,X^2 + 6\,X + 1 \\ H_{-195}[w_{7,13}] &= X^4 - 9\,X^3 + 2\,X^2 + 9\,X + 1 \end{aligned} \tag{1.11}$$

et ainsi de suite. Effectivement, les polynômes ont tous 1 comme coefficient constant ; notons également que tous ces polynômes sont à coefficients bien plus petits que les polynômes pour les autres fonctions de classes vues jusqu'ici.

Néanmoins, les quotients doubles de η (ainsi que les quotients simples pour $\ell = 11$ et $\ell \geqslant 17$) posent un problème : ils fournissent certes des polynômes de classes petits, mais comment en déduire les courbes elliptiques à multiplication complexe ? Dans le cas des fonctions de Weber et des quotients simples pour $\ell \in \{3, 5, 7, 13\}$, c'est simple : j s'écrit comme une expression rationnelle avec coefficients sur $\mathbb{Q}$ en la fonction de classes w, ce qui permet de calculer le j-invariant réduit $\overline{j}$ dans le corps fini a partir d'une racine $\overline{w}$ du polynôme de classes dans le même corps.

Nous proposons dans [68] comme solution à ce problème de se servir de l'équation modulaire, un polynôme bivarié dont le couple (w, j) est une racine, et qui pour les fonctions qui nous intéressent est définie sur $\mathbb{Z}$ (pour plus de détails, voir le chapitre 3). Pour obtenir $\overline{j}$, on réduit donc ce polynôme modulo la caractéristique du corps fini, substitue $\overline{w}$ pour w et calcule toutes les racines du polynôme univarié en résultant. Malheureusement, quand il y a plusieurs racines, toutes ne correspondent pas forcément à une bonne courbe ; dans ce cas, il faut construire toutes les courbes et vérifier leur anneau d'endomorphismes. Cela peut en général se faire en choisissant un point au hasard et en éliminant la courbe si ce point n'a pas la torsion attendue. Théoriquement, il est possible que plusieurs courbes survivent à ce test ; dans ce cas, on pourrait calculer l'anneau des endomorphismes par l'algorithme de Kohel [123, 75].

Notons que pour les quotients doubles de η qui sont invariants sous l'involution de Fricke–Atkin–Lehner, les racines du polynôme modulaire viennent en couples ; à une valeur $w(z) = w\left(\frac{-N}{z}\right)$ sont associées à la fois les racines $j(z)$ et $j\left(\frac{-N}{z}\right) = j\left(\frac{z}{N}\right)$ par l'invariance de j sous la réflexion $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. Ainsi, les deux courbes elliptiques associées sont reliées par une isogénie de degré N, et au moins quand ni p_1 ni p_2 ne divisent le conducteur, elles ont la même multiplication complexe. Cela permet de conclure directement quand l'équation modulaire a un degré 2 en j, ce qui est le cas pour $(p_1, p_2) = (3, 13)$ ou $(3, 5)$; dans les autres cas, cette observation permet de se passer de l'algorithme de Kohel quand deux courbes survivent au test de la torsion.

1.3 Hauteur des polynômes de classes

Les résultats de cette section sont tirés de [64], travail commun avec François Morain, notamment en ce qui concerne les expériences numériques ; et de [58] pour la plupart des théorèmes.

L'existence d'une infinité de fonctions de classes exhibée à la section 1.2 engendre l'embarras du choix : quelle fonction utiliser pour un discriminant donné ? Quelques éléments de réponse ont été donnés à la fin de la section 1.2.3 : on préférera un invariant avec un polynôme de classes défini sur $\mathbb{Q}$ et non sur le corps quadratique K ; on préférera un invariant qui donne peu de candidats pour le j-invariant de la courbe, c'est-à-dire un invariant dont le degré en j de l'équation modulaire est aussi petit que possible.

Mais le critère crucial est donné par la hauteur logarithmique du polynôme, c'est-à-dire par la longueur binaire de son plus grand coefficient. C'est cette hauteur qui détermine la précision flottante minimum à laquelle il faut exécuter les calculs. En supposant que les erreurs d'arrondi n'ont pas d'effet important sur l'exactitude des calculs (une supposition non prouvable, mais soutenue par les expériences numériques), on a même égalité entre la hauteur du polynôme et la précision des calculs.

Rappelons que la *hauteur logarithmique* d'un polynôme $f = \sum_k f_k X^k$ défini sur un corps de nombres L est donnée par

$$\mathcal{H}(f) = \frac{1}{[L : \mathbb{Q}]} \sum_v \log \max_k |f_k|_v,$$

où la somme est prise sur les valeurs absolues $|\cdot|_v$ de L, normalisées pour tenir compte de la ramification et de l'inertie. Cette définition rend la hauteur invariante sous extensions de corps. Quand le polynôme est unitaire et que ses coefficients sont des entiers algébriques, le maximum pour les valuations non-archimédiennes est de 1, pris en le coefficient dominant 1. Ainsi, seules les valuations archimédiennes comptent dans ce cas. Plus particulièrement, quand les coefficients sont dans $\mathbb{Z}$ ou l'anneau des entiers $\mathcal{O}_\Delta$ d'un corps quadratique imaginaire, les seuls cas qui nous intéressent dans le contexte des polynômes de classes, la définition se simplifie en

$$\mathcal{H}(f) = \max_k \log |f_k| \,;$$

dans le cas de $\mathbb{Z}$, il s'agit donc bien du nombre de chiffres du plus grand coefficient, qui détermine la précision des calculs.

Quand le polynôme est défini sur $\mathcal{O}_\Delta$, la précision dépend plutôt du maximum des $\log |f_{k,0}|$ et $\log |f_{k,1}|$, où les f_k ont été décomposés comme $f_k = f_{k,0} + f_{k,1}\omega$ sur une $\mathbb{Z}$-base $[1, \omega]$ de $\mathcal{O}_\Delta$. Comme $\log |\omega|$ est petit devant les $\log |f_{k,i}|$, plutôt de l'ordre de $\sqrt{|D|}$ comme expliqué dans la suite, cette notion de taille coïncide quasiment avec la hauteur logarithmique ; expérimentalement, nous n'avons observé de différences entre les deux que dans le troisième chiffre significatif.

1.3.1 Résultats expérimentaux

La contribution essentielle de [64] est une étude expérimentale et heuristique des hauteurs de polynômes de classes, à commencer par les polynômes classiques $H_D[j]$. Expérimentalement, il s'avère que les valeurs singulières de j sont grandes, de sorte que la hauteur se manifeste essentiellement dans la norme. Nous avons calculé les $H_D[j]$ pour tous les discriminants de nombre de classes entre 2 et 64. Parmi les 17 702 polynômes, le plus grand coefficient est la norme dans 17 120 cas ; dans 380 cas, c'est l'avant-dernier coefficient, dans 202, c'est le coefficient de X^2 et une seule fois, le coefficient de X^3.

Partant de cette observation, nous considérons le logarithme de la norme comme bonne approximation de la hauteur du polynôme de classes. Puis nous approchons la fonction j par le premier terme dans son développement de Laurent à l'infini, q^{-1}. Dans une forme réduite quadratique $[A, B, C]$, la valeur absolue de j serait donc proche de $e^{2\pi\Im(\tau)} = e^{\pi\sqrt{|D|}/A}$, ce qui donne une approximation pour la hauteur de

$$\mathcal{H}(H_D[j]) \approx \pi\sqrt{|D|} \sum_{[A,B,C]\in \mathrm{Cl}_D} \frac{1}{A}, \tag{1.12}$$

la somme étant prise sur les formes réduites représentant le groupe de classes.

Dans mon implantation (décrite en plus de détails à la section 1.7), j'utilise avec succès cette approximation, augmentée de 1% pour les grands discriminants.

Pour d'autres fonctions de classes, nous avons tracé la hauteur du polynôme par rapport à l'approximation (1.12) pour j, voir la figure 1.1.

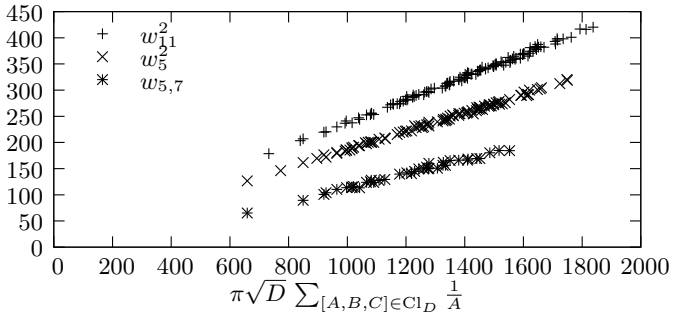


FIG. 1.1 – Hauteur pour des fonctions de classes alternatives

Il s'avère que les hauteurs croissent linéairement avec l'approximation, et que la pente dépend uniquement de la fonction de classes. Par régression linéaire, nous avons déterminé la pente et constaté qu'une bonne approximation en est donnée par

- $\frac{e(\ell-1)}{12(\ell+1)}$ pour w_ℓ^{2e} ;
- $\frac{(p_1-1)(p_2-1)}{12\psi(p_1p_2)}$ pour w_{p_1,p_2} avec $\psi(p_1p_2) = (p_1+1)(p_2+1)$ pour $p_1 \neq p_2$ et $\psi(p^2) = p(p+1)$.

Mais cette quantité n'est rien d'autre que le quotient des degrés en j et en w de l'équation modulaire reliant j et w, voir les théorèmes 3.3 et 3.5 et (3.3).

Cette observation peut se formaliser comme suit. Notons par $\mathcal{M}$ la *mesure de Mahler* d'un polynôme, qui est à un facteur constant près la hauteur logarithmique d'une de ses racines en tant que nombre algébrique. En appliquant la proposition B.3.5(b) de [113] nous démontrons dans [64] le théorème suivant :

Théorème 1.23 *Pour une fonction de classes w soit $\Phi(w, j)$ le polynôme modulaire entre w et j. Si $|D|$ tend vers l'infini, alors*

$$\mathcal{M}(H_D[w]) \sim \frac{\deg_j \Phi}{\deg_w \Phi} \mathcal{M}(H_D[j]).$$

Nos observations peuvent donc été réinterprétées comme une confirmation expérimentale que pour les polynômes de classes, la mesure de Mahler se comporte essentiellement comme la hauteur logarithmique.

Le théorème 1.23 nous permet de classer les fonctions modulaires par rapport au facteur qu'elles font gagner asymptotiquement sur j dans la hauteur du polynôme. En pratique, on choisira alors la première fonction adaptée au discriminant donné, en accordant éventuellement une préférence aux polynômes à coefficients entiers.

Le tableau 1.1 donne le début du classement par ordre décroissant de mérite ; sont mentionnées les fonctions de niveau ℓ respectivement p_1p_2 ne dépassant pas 200 avec le facteur qu'elles font gagner si celui-ci est au moins 13. La fonction w_2 peut être remplacée par n'importe quelle fonction de Weber.

	w_2	$>$	$w_{2,73}$	$>$	$w_{2,97}$	$>$	w_2^2	$>$	$w_{3,13}$	$>$	$w_{3,37}$	$>$	$w_{3,61}$						
	72		37		147/4		36		28		76/3		124/5						
$>$	w_2^3	$=$	w_3^2	$=$	$w_{5,7}$	$>$	$w_{2,13}^2$	$=$	$w_{5,13}$	$>$	$w_{5,19}$	$>$	$w_{5,31}$						
	24						21				20		96/5						
$>$	$w_{2,37}^2$	$=$	$w_{5,37}$	$>$	$w_{7,13}$	$>$	$w_{2,61}^2$	$>$	w_2^4	$=$	w_5^2	$=$	$w_{7,17}$						
	19				56/3		93/5		18										
$>$	$w_{11,13}$	$>$	w_7^2	$=$	$w_{3,7}^2$	$>$	$w_{13,13}$	$>$	w_{11}^2	$>$	w_{13}^2								
	84/5		16				91/6		72/5		14								
$>$	w_{17}^2	$=$	$w_{2,17}^3$	$>$	w_{19}^2	$=$	$w_{3,19}^2$	$>$	w_{23}^2										
	27/2				40/3				144/11										

Tab. 1.1 – Fonctions de classes et gains de hauteur

1.3.2 Théorèmes sur la hauteur

Bien que les résultats expérimentaux et heuristiques de la section précédente pourraient être considérés suffisants d'un point de vue d'un praticien, il reste néanmoins à donner des preuves rigoureuses expliquant le comportement observé. La hauteur est en effet cruciale non seulement pour les algorithmes à base d'approximations flottantes, mais également pour les algorithmes p-adiques dont il sera question à la section 1.5.1. En principe, ces algorithmes renvoient de façon prouvée les polynômes de classes recherchés, tandis qu'il ne semble pas possible de rigoureusement exclure que des erreurs d'arrondi n'engendrent un faux résultat pour les calculs flottants (mais encore faudrait-il que cette erreur approche un entier à très grande précision...). Pour garder cette propriété de calcul certifié, il faut néanmoins borner la hauteur des polynômes afin de déduire l'exactitude du polynôme sur les entiers de l'exactitude pour une précision p-adique choisie.

Ainsi, il est souhaitable de disposer d'une borne supérieure rigoureuse pour la hauteur ; en même temps, pour être pratiquement exploitable, cette borne doit être proche de la vérité, disons à un facteur logarithmique près. Dans [58], je donne la première telle borne pour les polynômes classiques $H_D[j]$. Il est sans doute possible d'appliquer le même raisonnement à d'autres invariants de classes ; le théorème 1.23 donne déjà le bon résultat, à la subtilité près que la mesure de Mahler n'est pas tout à fait la hauteur logarithmique.

Voici une esquisse de la démonstration donnée dans [58]. Dans un premier temps, on approche j par le premier terme de son développement en série. Comme dans le raisonnement heuristique de la section précédente, cela donne l'approximation $\pi\sqrt{|D|}\sum_{[A,B,C]}\frac{1}{A}$, la somme étant prise sur les formes réduites. Une borne explicite sur les coefficients de la série pour j tirée de [25], ainsi que l'observation que l'argument de j peut être supposé dans le domaine fondamental évoqué à la proposition 1.3, permettent de borner l'erreur $|j(z) - q^{-1}|$ par une constante explicite.

Il reste alors à borner la somme des $\frac{1}{A}$. Schoof démontre dans [168, Lemma 2.2] que cette somme est dans $O(\log^2|D|)$. En bornant le nombre de formes avec un A donné (autrement dit, le nombre de B solutions de l'équation $B^2 \equiv D \pmod{4A}$) et en utilisant quelques estimations standard de la théorie analytique des nombres, on peut rendre toutes les constantes explicites. La borne qui en résulte dépend du nombre de classes h_D, ce qui n'est pas gênant en pratique : en tant que degré du polynôme de classes, il est forcément connu. En appliquant un raisonnement semblable au précédent, on peut également obtenir une borne supérieure pour h_D et ainsi rendre la borne pour la hauteur indépendante de h_D. J'obtiens finalement les bornes explicites suivantes.

Théorème 1.24 *Soit D un discriminant quadratique imaginaire. Alors son nombre de classes h_D est borné supérieurement par*

$$2N(\log N + \gamma) + 1 \in \sqrt{\frac{|D|}{3}}\log|D| + O\left(\sqrt{|D|}\right)$$

où $N = \sqrt{\frac{|D|}{3}}$ et $\gamma = 0{,}577\ldots$ est la constante d'Euler.

La hauteur logarithmique de $H_D[j]$ est bornée supérieurement par

$$\begin{aligned} & c_5 h_D + c_1 N \left(\log^2 N + 4\gamma \log N + \frac{\log N + \gamma + 1}{N} + c_6 \right) \\ \leqslant\ & c_1 N \log^2 N + c_2 N \log N + c_3 N + c_1 \log N + c_4 \\ \in\ & \frac{\pi}{4} \sqrt{|D|} \log^2 |D| + O\left(\sqrt{|D|} \log |D| \right) \end{aligned}$$

où $c_1 = \sqrt{3}\pi = 5{,}441\ldots$, $c_2 = 17{,}824\ldots$, $c_3 = 15{,}603\ldots$, $c_4 = 11{,}212\ldots$, $c_5 = 2{,}630\ldots$ *et* $c_6 = 2{,}309\ldots$.

1.4 Décomposition galoisienne du corps de classes

Cette section présente les résultats de [65], obtenus en collaboration avec François Morain.

En regardant de plus près l'algorithme 1.15 pour obtenir des courbes elliptiques à multiplication complexe donnée sur un corps fini, on réalise qu'il se décompose en deux phases, qui sont régies par des paramètres bien distincts : dans un premier lieu, il faut calculer un polynôme de classes ; cette phase dépend du nombre de classes et de la précision flottante requise (tous les deux fonctions du déterminant comme détaillé à la section précédente). Deuxièmement, il faut trouver une racine de ce polynôme et construire effectivement la courbe ; cette phase dépend également du nombre de classes, qui est le degré du polynôme, mais un nouveau paramètre s'introduit, le cardinal du corps fini.

La première phase a une complexité en $O(h_D^{2+\varepsilon})$ pour tout $\varepsilon > 0$ (voir l'analyse de la section 1.5). La factorisation du polynôme, étape dominante de la deuxième phase, prend un temps $O((h_D \log^2 q)^{1+\varepsilon})$ si des algorithmes probabilistes et l'arithmétique rapide sont utilisés. Ainsi, aucune des deux phases ne domine l'autre, et il faut tenir compte du rapport des deux paramètres dans une analyse du temps de calcul.

Quand on détermine des courbes elliptiques à des fins cryptographiques, la taille du corps est généralement fixée entre 160 et 200 bits (voir les considérations de sécurité à la section 4.1), tandis qu'il y a des recommandations pour un nombre de classes minimum. Suivant Frey [77], l'agence fédérale allemande pour la sécurité informatique (BSI) préconise depuis 1999 des nombres de classes d'au moins 200 [156]. Dans ces conditions, la recherche de la racine est en général négligeable par rapport au calcul du polynôme de classes.

La situation est différente pour les preuves de primalité. Ici, la taille du corps peut atteindre 20 000 chiffres décimaux pour les calculs record [144], et la factorisation du polynôme de classes est loin d'être négligeable.

Indépendamment de l'application, notons également qu'un polynôme de classes pour un seul et même discriminant peut être réutilisé afin d'obtenir des courbes sur un grand nombre de corps finis. Ainsi, le calcul des polynômes de classes peut être considéré comme précalcul, et il convient d'optimiser la phase de factorisation. C'était le cas pour les premières versions d'ECPP [142] distribuées notamment à travers le système de calcul

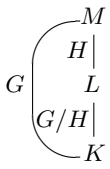
symbolique `magma`, et il y a également des collections de polynômes de classes sur le web [182]. (Ces considérations restent pourtant, au moins pour les preuves de primalité, d'ordre pratique et ne changent rien à la complexité asymptotique. En effet, pour certifier des nombres premiers de plus en plus grands, il faut que le discriminant quadratique tende vers l'infini.)

Il peut donc être rentable de passer plus de temps lors de la construction des polynômes de classes, pour en gagner surproportionnellement lors de la recherche de racines. Une approche naturelle consiste à décomposer le corps de classes, obtenu par un élément primitif, en une tour d'extensions relatives.

1.4.1 Algorithme de base

Dans l'optique de la résolution par radicaux, un algorithme pour la décomposition d'une extension Galoisienne résoluble en une tour d'extensions cycliques de degré premier est donné dans [104]. L'hypothèse de travail est que les conjuguées d'un élément primitif ainsi que l'action explicite du groupe de Galois sur elles sont connues ; c'est le cas, par exemple, en multiplication complexe via l'isomorphisme entre le groupe de Galois et le groupe de classes du corps quadratique sous-jacent.

Soit M/K une extension de corps engendrée par un élément x de polynôme minimal f, et G son groupe de Galois. Supposons H un sous-groupe distingué de G, et L le sous-corps de M fixé par H, comme dans le dessin suivant.



La première étape de l'algorithme consiste à calculer un élément engendrant L/K et son polynôme minimal. Pour cela, soit $\overline{G} \subseteq G$ un système de représentants de G/H, et définissons

$$u_g(X) = \prod_{h \in H} (X - x^{hg}) = \sum_{k=0}^{|H|} (-1)^{|H|-k} u_{g,k} X^k \text{ pour tout } g \in \overline{G},$$

où les $u_{g,k}$ sont les fonctions symétriques élémentaires en les x^{hg}. Elles sont invariantes sous H et vivent donc dans L ; en effet, $f = \prod_{g \in \overline{G}} u_g$ est la factorisation de f dans $L[X]$. En relevant les coefficients des u_g colonne par colonne, on peut former les polynômes

$$v_k(Y) = \prod_{g \in \overline{G}} (Y - u_{g,k}).$$

Étant invariants sous G, ces polynômes sont à coefficients dans K. Au moins l'un d'entre eux est irréductible et engendre L/K ; notons-le par v et une de ses racines par y. (Dans l'embarras du choix, on préférera comme souvent le polynôme $v_{|H|-1}$ associé à la trace.)

Les conjuguées de x et l'action du groupe de Galois étant connues, les v_k se calculent explicitement, par exemple par approximations flottantes.

Dans une deuxième phase, il faut exprimer les coefficients des u_g en tant qu'éléments de L, ce qui donne à la fois le polynôme minimal u_1 de x sur L et la factorisation totale de f. Dans [104], la base polynomiale $(y^i)_{i=1}^{|G/H|}$ de L/K est utilisée ; la connaissance de l'action du groupe de Galois permet d'interpréter l'écriture des éléments de L comme un problème d'interpolation d'un polynôme dont les valeurs dans les conjuguées de y sont données.

Dans notre cas, $K = \mathbb{Q}$ et tous les nombres algébriques qui apparaissent sont des entiers. Cela permet d'obtenir v exactement en arrondissant ses coefficients vers $\mathbb{Z}$. Si le défaut d'intégralité d de la base polynomiale formée à partir de y est connu, l'interpolation des éléments de L donne des rationnels avec dénominateur d, qui peuvent également être récupérés exactement.

Pour un G résoluble, on peut supposer H d'ordre premier et itérer la décomposition sur L/K pour finalement arriver à une tour d'extensions de degré premier.

Le but de nos travaux dans [65] était de rendre cet algorithme de base aussi efficace que possible. Dans un premier lieu, nous remplaçons la base polynomiale par la représentation de Hecke, plus compatible avec l'intégralité des nombres. Cette représentation a été utilisée par Hecke dans [109, §20] pour passer d'un élément engendrant une extension de corps à un autre.

Définition 1.25 Soit L/K engendré par un élément y de polynôme minimal v, et soit $z \in L$. Notons les conjuguées de y et de z par y_i et z_i respectivement. La *représentation de Hecke* de z est donnée par le polynôme

$$g_z(Y) = \sum_{i=1}^{[L:K]} z_i \frac{v(Y)}{Y - y_i}.$$

Notons que $g_z \in K[Y]$, et que

$$z = \frac{g_z}{f'}(y).$$

De plus, cette écriture préserve l'intégralité : si z et y sont entiers, alors les coefficients de g_z le sont.

Notons que le calcul de g_z à partir des conjuguées en nombres flottants ressemble à l'interpolation de Lagrange, et est en fait plus simple : pour interpoler, il faudrait d'abord calculer les z_i, qui sont gratuits dans notre contexte. Cette demie-interpolation est une opération standard du calcul formel et est appelée « combinaison linéaire à partir de modules linéaires » dans [85, Algorithm 10.9]. En organisant les calculs sous forme d'un arbre, elle se fait en temps

$$O(\mathrm{M_X}(d) \log d),$$

avec $d = [L : K]$ et $\mathrm{M_X}(d)$ le temps pour multiplier deux polynômes de degré d à coefficients flottants. Supposons que les coefficients arrondis de g_z ont $O(n)$ bits, de

sorte qu'il suffit d'exécuter les calculs flottants avec cette même précision. En utilisant la FFT partout, nous obtenons

$$\mathrm{M_X}(d) = O(d \log d \, \mathrm{M}(n)) = O(d \log d \, n \log n \log\log n),$$

où $\mathrm{M}(n)$ est la complexité binaire pour multiplier deux flottants à précision n bits. La complexité totale de l'écriture d'un élément de L devient

$$O(dn \log^2 d \log n \log\log n).$$

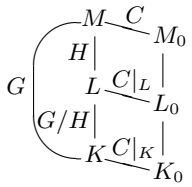
C'est le premier exemple d'un algorithme quasiment linéaire en la taille de sa sortie $O(dn)$.

1.4.2 Le cas non-galoisien

La deuxième contribution de [65] est de généraliser l'algorithme de la section précédente au cas non-galoisien tel qu'il se présente pour les corps de classes. Ici, $K = \mathbb{Q}(\sqrt{D})$, et M est le corps de classes d'anneaux pour l'ordre de discriminant D, effectivement engendré par une valeur singulière $x = f(\tau)$ d'une fonction de classes, de sorte que le polynôme minimal de x est donné par $H_D[f]$, voir la définition 1.16. M/K est galoisien de groupe de Galois Cl_D. Pour la plupart des fonctions de classes, $H_D[f]$ est à coefficient dans $\mathbb{Z}$, et x engendre alors le sous-corps réel M_0 de M sur le sous-corps réel $K_0 = \mathbb{Q}$ de K. Ce corps M_0 n'est pas galoisien sur $\mathbb{Q}$, mais ressemble beaucoup à une extension galoisienne : en ajoutant $\sqrt{D}$, on arrive à M, qui est galoisien sur $\mathbb{Q}$. (La situation est analogue au traitement des extensions de Kummer, simplifié en ajoutant d'abord les bonnes racines de l'unité.)

Comme déjà évoqué à la section 1.2.3, le groupe de Galois $\hat{G}$ de $M/\mathbb{Q}$ est donné par le produit semi-direct de $G = \mathrm{Cl}_D$ et du groupe C d'ordre 2 engendré par la conjugaison complexe κ, de sorte que $\kappa \mathfrak{a} \kappa = \mathfrak{a}^{-1}$ pour $\mathfrak{a} \in G$. Évidemment, on peut décomposer l'extension $M/\mathbb{Q}$ de degré $2h_D$ en une tour d'extensions de degré premier comme décrit à la section précédente ; néanmoins, il serait préférable de travailler directement sur l'extension $M_0/\mathbb{Q}$ de degré h_D, bien qu'elle ne soit pas galoisienne. D'un point de vue pratique, cela permettra de faire tous les calculs sur les réels au lieu des complexes.

La constellation examinée dans [65] ne se limite pas aux corps de classes d'ordres quadratiques. Elle est représentée par le dessin suivant :



Nous supposons M/K_0 galoisien de groupe $\hat{G}$, M/M_0 galoisien de groupe C, et que C possède un complément normal G dans $\hat{G}$, de sorte que $\hat{G}$ est en fait le produit semi-direct $\hat{G} = G \rtimes C$. Soit L le sous-corps fixé par un sous-groupe distingué $H \lhd G$, et L_0 le sous-corps fixé par $\langle H, C \rangle$. Nous supposons que C normalise H, c'est-à-dire que

$chc^{-1} \in H$ pour $c \in C$ et $h \in H$. Alors, $\langle H, C\rangle = H \rtimes C$, et L/L_0 est galoisien de groupe $C|_L$, de même que K/K_0 est galoisien de groupe $C|_K$.

Nous montrons alors qu'en définissant les u_g et v_k comme à la section précédente, leurs coefficients se retrouvent dans L_0 et K_0 respectivement, de sorte que l'algorithme s'applique directement à M_0/K_0. *A priori*, la représentation de Hecke des éléments de L_0 s'obtient par des calculs avec les racines de f dans M, tandis que le résultat final est un polynôme défini sur K_0. En réunissant d'abord les orbites sous C, nous généralisons les algorithmes rapides sur les polynômes pour travailler directement dans M_0 à la place de M. Dans le cas des polynômes de classes définis sur $\mathbb{Z}$, cela revient à regrouper d'abord une valeur singulière et sa conjuguée complexe ; à partir de cela, les calculs se font avec des nombres réels au lieu de complexes, ce qui fait gagner un facteur 3 au temps de calcul.

1.5 Algorithme quasi-linéaire pour le corps de classes

Dans cette section je présente les résultats de [58].

1.5.1 Motivation

En 2002, l'approche classique de la multiplication complexe, passant par des approximations complexes flottantes, a été concurrencée par deux algorithmes p-adiques pour le calcul du polynôme de classes dus à Couveignes et Henocq [38]. Partant d'une courbe elliptique ordinaire avec le bon anneau d'endomorphismes sur un petit corps fini $\mathbb{F}_p$, le premier algorithme détermine son *relèvement canonique* non plus dans $\mathbb{C}$, mais dans le corps p-adique $\mathbb{C}_p$. Ce relèvement est caractérisé par le fait qu'il a multiplication complexe par le même ordre quadratique ; son existence découle du théorème de Deuring 1.13. Les calculs se font modulo des puissances de plus en plus grandes de l'idéal premier au-dessus de p, c'est-à-dire en fin de compte dans $\mathbb{Z}/p^k\mathbb{Z}$.

La méthode proposée est de relever le j-invariant de la courbe arbitrairement dans $\mathbb{Q}_p$ et d'appliquer une isogénie correspondant à l'action d'un idéal principal. Cette action est calculable algébriquement via des polynômes modulaires (cf. le chapitre 3) si l'isogénie est *friable*, c'est-à-dire se décompose en un produit d'isogénies de petit degré. Si le j-invariant était déjà correct, il resterait fixe sous l'action d'un idéal principal. Ici, il n'est donné que par une approximation p-adique, et on peut appliquer une méthode de Newton pour mettre à jour le j-invariant en doublant sa précision. Une fois la précision souhaitée atteinte, les conjuguées s'obtiennent en calculant les isogénies correspondant au groupe de classes, le polynôme de classes est dérivé sur les p-adiques et ses coefficients sont interprétés comme des éléments de $\mathbb{Z}$. Si une borne explicite est connue sur la hauteur du polynôme (ce qui est le cas à travers le théorème 1.24 au moins pour le polynôme de classes de base $H_D[j]$), l'algorithme p-adique est garanti de fournir le bon résultat. Son temps de calcul est donné sous l'hypothèse de Riemann généralisée (utilisée pour estimer la probabilité de friabilité d'un idéal principal dans le corps quadratique imaginaire) par $O(|D|^{1+\varepsilon})$.

Cette complexité peut être considérée comme quasi-linéaire en la taille de la sortie, qui est de l'ordre de $O(|D| \log^3 |D|)$ d'après le théorème 1.24. Notons néanmoins qu'ici, le terme $|D|^\varepsilon$ n'est pas polynômial en $\log |D|$, mais seulement sous-exponentiel ; il ne s'agit donc pas de quasi-linéarité *stricto sensu*.

L'algorithme p-adique a été implanté par Bröker dans sa thèse [26], dont j'étais examinateur. Bröker donne également une généralisation pour pouvoir traiter des fonctions de classes alternatives, condition nécessaire pour arriver à une implantation compétitive. Le plus grand exemple donné dans [26, Section 7.5] concerne le discriminant $D = -92\,806\,391$ de nombre de classes $h_D = 15\,610$, pour lequel un polynôme de classes avec l'une des fonctions de Weber est calculé. Le temps de calcul sur un PC de 32 bits cadencé à 2,8 GHz est donné comme une quinzaine de minutes. À titre de comparaison, mon implantation utilisant des nombres flottants prend 570 s sur un Athlon-64 cadencé à 2,2 GHz ou 2 900 s sur un Pentium-M cadencé à 1,8 GHz, ce qui est du même ordre de grandeur.

Le deuxième algorithme opère sur une courbe supersingulière et calcule par des méthodes similaires un relèvement de Deuring vers une courbe à multiplication complexe par $\mathcal{O}_D$. Son avantage réside dans le fait qu'il est plus facile de fixer un p (tout p inerte dans l'ordre maximal associé à $\mathcal{O}_D$ fera l'affaire) et de trouver une courbe supersingulière sur $\mathbb{F}_p$ (en tant que réduction d'une courbe à multiplication complexe par un autre, petit discriminant D' tel que p est également inerte dans $\mathbb{Q}(\sqrt{D'})$), au lieu de résoudre l'équation de la norme et de parcourir les courbes ordinaires pour en trouver une avec le bon anneau d'endomorphismes. Pour le moment, il n'a pas été généralisé à des fonctions de classes autres que j ; c'est pourquoi le record obtenu par Lercier et Riboulet-Deyris dans [127] n'est que pour $D = -(10^9 + 4099)$ de nombre de classes $h_D = 21\,313$, dont le polynôme de classes prend 4 gigaoctets.

Cette nouvelle concurrence à laquelle l'approche classique a l'air de bien résister en pratique, pose la question de la complexité théorique de l'algorithme utilisant des approximations flottantes. En effet, cette complexité n'avait jamais été correctement analysée dans la littérature, certains auteurs allant aussi loin que de prétendre qu'elle était exponentielle en $|D|$.

1.5.2 Évaluation rapide de fonctions modulaires

Le goulot d'étranglement pour le calcul d'un polynôme de classes par approximations flottantes s'avère être l'évaluation de fonctions modulaires pour obtenir les conjuguées. Supposons donc qu'une fonction de classes f ait été fixée. Soit $n = n_D$ la précision en bits nécessaire pour calculer $H_D[f]$; en pratique, on choisira n comme une borne supérieure sur la hauteur du polynôme, telle que donnée au théorème 1.24. Examinons la complexité d'évaluer f quand le discriminant et ainsi n tendent vers l'infini.

L'approche la plus naturelle pour évaluer une fonction modulaire est de se servir de son développement en série de Laurent en la variable $q^{1/N}$ à l'infini, qui existe d'après la définition 1.6. Quand il s'agit de j, on peut supposer l'argument z dans le domaine fondamental de la proposition 1.3, de sorte que $\Im z \geqslant \frac{\sqrt{3}}{2}$ et $|q| \leqslant e^{-\pi\sqrt{3}}$ est borné par

une constante. Les coefficients de la série (1.3) $j = q^{-1} + \sum_{\nu \geqslant 0} c_\nu q^\nu$ croissant moins que linéairement, plus précisément $0 \leqslant c_\nu \leqslant \frac{e^{4\pi\sqrt{\nu}}}{\sqrt{2}\,\nu^{3/4}}$ d'après [25], on déduit aisément que $O(n)$ termes suffisent pour obtenir une précision du résultat de n chiffres. Ainsi, la complexité d'une évaluation est donnée par

$$O(n\,\mathrm{M}(n)),$$

où $\mathrm{M}(n)$ désigne la complexité d'une multiplication à précision n. Ceci comprend également l'exponentielle complexe, de complexité $O(\log n\ \mathrm{M}(n))$ d'après [23].

Pour d'autres fonctions de classes, leur domaine fondamental contient nécessairement une autre pointe que l'infini, qui est un nombre rationnel et donc à partie imaginaire nulle. Dans ce cas, il faudrait se servir des autres développements en série quand on se rapproche trop d'une pointe, qui existent d'après le quatrième point de la définition 1.6, pour arriver au même résultat.

Dans la pratique, on préférera une autre approche, toute aussi simple. Notons que j ainsi que toutes les autres fonctions de classes introduites à la section 1.2 s'écrivent à partir de la fonction η de Dedekind. Ainsi, il suffit de savoir évaluer cette fonction en n'importe quel argument. η étant une forme modulaire de poids $1/2$, elle n'est pas invariante sous toute matrice dans Γ ; par contre, sa transformation sous matrices unimodulaires est bien connue (cf. [43, §4]), de sorte qu'on peut transformer son argument dans le domaine fondamental standard de la proposition 1.3. Cela résout déjà le problème des pointes. Notons ensuite que la représentation en série (1.4) est creuse (tandis que le produit infini est dense) : pour une précision de n chiffres, il faut calculer les termes jusqu'à un exposant de $O(n)$, et il y en a $O(\sqrt{n})$. Le fait que les exposants sont les valeurs de deux polynômes permet d'utiliser l'approche des différences itérées pour obtenir chaque nouvel exposant par un nombre constant d'additions, ou autrement dit, chaque nouveau terme de la série par un nombre constant de multiplications. Précisément, la récursion classique suivante pour q^ν, $q^{2\nu-1}$, $q^{\nu(3\nu-1)/2}$ et $q^{\nu(3\nu+1)/2}$ renvoie deux nouveaux termes au prix de quatre multiplications :

$$\begin{aligned} q^{\nu+1} &= q^\nu \cdot q \\ q^{2(\nu+1)-1} &= q^{2\nu-1} \cdot q^2 \\ q^{\nu(3(\nu+1)-1)/2} &= q^{\nu(3\nu+1)/2} \cdot q^{2(\nu+1)-1} \\ q^{\nu(3(\nu+1)+1)/2} &= q^{\nu(3(\nu+1)-1)/2} \cdot q^{\nu+1} \end{aligned}$$

La complexité d'évaluer une fonction modulaire composée d'un nombre constant de fonctions η devient ainsi

$$O(\sqrt{n}\ \mathrm{M}(n)),$$

ce qui n'est pas encore linéaire, mais déjà mieux que quadratique, et presque toujours le plus rapide en pratique.

Une possibilité d'obtenir une complexité quasi-linéaire est développée dans [58] ; elle s'applique aussi bien au calcul direct d'une fonction modulaire à partir de son développement en série qu'au calcul indirect en passant par η. La clef en est l'observation que dans

tous les cas, l'évaluation se réduit en une évaluation d'un polynôme après avoir tronqué la série ; et cette évaluation doit se faire non en un seul point, mais en plusieurs. Le calcul formel a développé des algorithmes rapides pour cette tâche de *multiévaluation*, qui partent de l'observation que $f(z)$ pour un polynôme $f(X)$ n'est rien d'autre que $f(X) \bmod X - z$, et qui organisent encore une fois les calculs sous la forme d'un arbre, comme déjà évoqué à la section 1.4.1. Ainsi, la complexité de l'algorithme de [85, §10.1] pour évaluer à précision n un polynôme de degré $O(d)$ en $O(h_D)$ arguments devient

$$O\left((d \log d + h_D \log^2 h_D + \log n)\ \mathrm{M}(n)\right),$$

si l'arithmétique rapide des polynômes par la FFT et la division avec reste par des itérations de Newton sont employées. Le terme $\log n\, M(n)$ correspond au précalcul d'une racine de l'unité d'ordre suffisamment grand. Dans notre cas, le nombre d'arguments est bien au plus h_D et en général proche de $\frac{h_D}{2}$ quand l'action de la conjugaison complexe sur le groupe de classes est exploitée comme décrit aux sections 1.2.3 et 1.4.2. Le degré d est de l'ordre de $O(n)$. Au final, il faudra remplacer n et h_D par les bornes du théorème 1.24, qui sont plus grandes d'un facteur logarithmique pour n que pour h_D. Ainsi, $d \log d$ et $h_D \log^2 h_D$ sont échangeables, et comme $\log h_D$ est de l'ordre de $\log n$, la complexité totale amortie par évaluation devient

$$O(\log^2 n\ \mathrm{M}(n)).$$

Cette complexité est quasi-linéaire en la taille de la sortie.

Encore une autre approche quasi-linéaire est donnée par Dupont. Elle se fonde sur la moyenne arithmético-géométrique et des itérations de Newton. L'algorithme de base de [49, Theorem 4] calcule la fonction modulaire k', dont le carré λ satisfait

$$j = \frac{256(1 - \lambda + \lambda^2)^3}{(\lambda(1-\lambda))^2}.$$

Si l'argument de k' a une partie imaginaire bornée supérieurement par une constante et que la précision n tend vers l'infini, la complexité devient

$$O(\log n\ \mathrm{M}(n)),$$

ce qui gagne un facteur $\log n$ par rapport à la multiévaluation. Notons que c'est précisément la complexité de la moyenne arithmético-géométrique.

Dans notre contexte, la précision augmente en même temps que le discriminant, qui influe sur la partie imaginaire maximum. Après discussion avec Dupont, celui-ci a raffiné son argumentation pour aboutir à [49, Theorem 5], qui obtient la même complexité uniformément dans l'argument. La modification consiste essentiellement en un passage aux algorithmes simples décrits ci-dessus dès que la partie imaginaire devient trop grande. Des valeurs de fonctions f autres que k' et j peuvent se calculer comme racine de l'équation modulaire reliant f à k' ou j, cf. le chapitre 3. Si une telle équation n'est pas (encore) disponible (notons que les techniques rapides d'évaluation de fonctions

modulaires serviront précisément pour calculer des équations modulaires par l'algorithme de la section 3.4), il est encore une fois possible de passer par η. L'algorithme de [49, section 7] calcule d'abord k' et λ, puis ϑ_{00}^2, une forme modulaire de poids 1, en tant qu'inverse de la moyenne arithmético-géométrique de 1 et de k', et finalement η par des itérations de Newton en tant que douzième racine de $\lambda(1-\lambda)\vartheta_{00}^2/16$. La complexité reste de $O(\log n\ \mathrm{M}(n))$, mais au prix d'une petite constante.

1.5.3 Calcul du groupe de classes

Une fois l'évaluation de fonctions modulaires instanciée par des algorithmes quasi-linéaires, c'est l'énumération des formes réduites représentant le groupe de classes qui risque de devenir la phase limitante au moins du point de vue de la théorie de la complexité. En pratique, par contre, elle s'avère être complètement négligeable même pour l'algorithme naïf que j'ai implanté dans le logiciel `cm`, décrit plus en détail à la section 1.7.3 ; celui-ci énumère tous les $A \leqslant \sqrt{\frac{|D|}{3}}$ et tous les $B \leqslant A$ et vérifie s'il y a un C entier tel que $D = B^2 - 4AC$. La complexité de cette approche est de $O(|D|)$ opérations arithmétiques sur des entiers de $\log|D|$ bits, ce qui s'approche dangereusement de la complexité totale du calcul de polynômes de classes comme énoncée au théorème 1.27.

Notons que le groupe de classes d'un corps quadratique imaginaire se calcule en temps sous-exponentiel en $\log|D|$ par un algorithme qui collecte des relations entre idéaux premiers de petite norme et effectue de l'algèbre linéaire pour obtenir la forme normale de Smith de la matrice entière en résultant, voir [103, 116, 56] et la section 4.2. L'algorithme renvoie la décomposition du groupe de classes dans un produit de groupes cycliques avec leur ordre et générateur. Un système de représentants du groupe peut en être déduit par au plus $h_D - 1$ compositions et réductions d'idéaux quadratiques. Chaque telle opération prends un temps de $O(\log|D|\ \mathrm{M}(\log|D|))$. Ainsi le groupe de classes est énuméré en $O\left(h_D \log|D|\, \mathrm{M}(\log|D|)\right)$. Utilisant la borne sur h_D du théorème 1.24, nous obtenons une complexité de

$$O\left(\sqrt{|D|}\log^3|D|\log\log|D|\log\log\log|D|\right),$$

ce qui est encore une fois quasi-linéaire en la taille de la sortie.

Une variante, sans doute préférable en pratique, consiste à faire engendrer le groupe de classes directement par les petits idéaux premiers. D'après un résultat de Bach [10, p. 376] prouvé sous l'hypothèse de Riemann généralisée, le groupe de classes est engendré par les idéaux premiers de degré d'inertie 1 et de norme bornée par $6\log^2|D|$. Notons l'ensemble de ces idéaux par $\{\mathfrak{p}_1, \mathfrak{p}_2, \ldots\}$. L'algorithme procède en énumérant les puissances de $\mathfrak{p}_1$ jusqu'à son ordre e_1 tel que $\mathfrak{p}_1^{e_1} = 1$ dans le groupe de classes. Puis, il calcule les puissances de $\mathfrak{p}_2$ jusqu'à ce que $\mathfrak{p}_2^{e_2}$ soit contenu dans le sous-groupe engendré par $\mathfrak{p}_1$ qui vient d'être construit ; il faut alors ajouter tous les $\mathfrak{p}_1^{a_1}\mathfrak{p}_2^{a_2}$ avec $0 \leqslant a_1 < e_1$ et $0 < a_2 < e_2$. On continue avec les puissances de $\mathfrak{p}_3$ jusqu'à tomber dans le sous-groupe engendré par $\mathfrak{p}_1$ et $\mathfrak{p}_2$, et ainsi de suite. Si les éléments déjà construits sont stockés dans une table de hachage, la recherche se fait en temps constant, et la complexité est dominée par $O(h_D)$ opérations dans le groupe de classes comme pour l'algorithme précédent.

Par les heuristiques de Cohen et Lenstra [32], on s'attend à ce que le groupe de classes soit presque toujours cyclique mis à part le sous-groupe de 2-torsion, dont les composants correspondent aux diviseurs du discriminant ; cette heuristique est très bien vérifiée expérimentalement [117]. Si le nombre de classes est connu, cela permet en pratique d'arrêter les calculs rapidement dès que suffisamment d'éléments ont été collectés. Après factorisation du discriminant en temps sous-exponentiel, le nombre de classes peut s'obtenir pour le discriminant fondamental Δ par la formule de Louboutin [130, Theorem 4] avec $O(\sqrt{|\Delta|\log|\Delta|})$ opérations arithmétiques, et h_D se calcule alors aisément par la formule analytique du nombre de classes de Kronecker.

Dans [58], je décris un algorithme légèrement plus rapide asymptotiquement, mais plus complexe à mettre en place. Il boucle sur $0 < A \leqslant \sqrt{\frac{|D|}{3}}$ et détermine toutes les solutions aux congruences $B = D \bmod 2$ et $\frac{B^2-D}{4} = 0 \bmod A$. Pour cela, il faut factoriser A en temps sous-exponentiel, ou mieux encore, énumérer directement les A jusqu'à la borne sous leur forme factorisée. Les racines carrées de D modulo les petits nombres premiers jusqu'à la borne pour A sont alors calculées par l'algorithme de Cipolla [30] et relevées modulo les puissances des premiers. Une recombinaison par restes chinois renvoie tous les candidats possibles. Ainsi, j'obtiens le résultat suivant :

Théorème 1.26 *Le groupe de classes d'un ordre quadratique imaginaire de discriminant D s'énumère par un algorithme probabiliste en temps*

$$O\left(\sqrt{|D|}\log^2|D|(\log\log|D|)^2\log\log\log|D|\right).$$

La taille de la sortie de l'algorithme est de l'ordre de $O\left(\sqrt{|D|}\log^2|D|\right)$ d'après le théorème 1.24 ; cette borne est donc linéaire mis à part un facteur non seulement logarithmique, mais doublement logarithmique.

1.5.4 Complexité du calcul du polynôme de classes

L'obstacle de l'énumération du groupe de classes étant écarté par les résultats de la section précédente, et la reconstruction du polynôme de classes à partir de ses racines se fondant sur un algorithme rapide qui agence les calculs sous forme d'un arbre, l'étape dominante pour le calcul d'un polynôme de classes devient le calcul de ses racines. En m'appuyant sur l'évaluation rapide de fonctions modulaires due à Dupont et décrite à la section 1.5.2, j'obtiens le résultat suivant dans [58] :

Théorème 1.27 *Soit f une fonction de classes pour une famille de discriminants D de nombres de classes h_D, et supposons une précision flottante de $n = n_D$ bits. Alors une approximation flottante du polynôme de classes $H_D[f]$ se calcule en temps*

$$O(h_D\, n\log^2 n\log\log n).$$

Avec les bornes du théorème 1.24 pour h_D et n, la complexité devient

$$O\left(|D|\log^5|D|\log\log|D|\right).$$

Avouons que ce « théorème » revêt néanmoins un caractère heuristique dans le sens que la correction du polynôme obtenu n'est pas démontrée. Dans le contexte de la multiplication complexe, ce n'est pas forcément gênant : on peut vérifier que la courbe elliptique sur un corps fini obtenue grâce à une racine de $H_D[f]$ admet bien $\mathcal{O}_D$ comme anneau d'endomorphismes.

1.6 Calcul direct de sous-corps du corps de classes

Les résultats de ce chapitre ont fait l'objet du brevet [67], déposé avec Michael Pohst et Reinhard Schertz.

Étant donné l'algorithme quasi-linéaire de la section 1.5, appliqué aux fonctions de classes de la section 1.2, on pourrait penser que l'optimum est atteint. Une analyse plus en profondeur des polynômes de classes pour w_{p_1,p_2}, par contre, nous a menés à des améliorations pour certains discriminants. Prenons, par exemple, les quatre polynômes pour $D = -195$ de (1.11). Deux parmi eux sont irréductibles sur $\mathbb{Q}$, ils engendrent donc bien le sous-corps réel du corps de classes de Hilbert sur $\mathbb{Q}$. Les deux autres, par contre, s'avèrent être des carrés :

$$H_{-195}[w_{3,13}] = (X^2 + 3X + 1)^2$$
$$H_{-195}[w_{5,13}] = (X^2 - 3X - 1)^2$$

Ainsi, leurs racines vivent bien dans le corps de classes conformément à la définition 1.16, mais elles engendrent un sous-corps d'indice 2.

On remarque que dans ces cas, les deux premiers composant le niveau sont également des diviseurs du discriminant. Ce n'est pas un hasard, comme le démontrent les résultats suivants.

Lemme 1.28 *Soient p_1 et p_2 deux premiers impairs distincts tels que $24|(p_1-1)(p_2-1)$, et soit $D = f^2\Delta$ un discriminant pour le discriminant fondamental Δ tel que p_1 et p_2 divisent Δ sans diviser f. Soient $Q = [A, B, C]$ une forme quadratique de discriminant D telle que $\operatorname{pgcd}(A, N) = 1$ et $C = N = p_1p_2$, et soit $\tau = \frac{-B+\sqrt{D}}{2A}$. D'après le théorème 1.22, la valeur singulière $w_{p_1,p_2}(\tau)$ associée à Q est un élément du corps de classes K_D. Notons par $\mathfrak{p}_i$ l'idéal ramifié au-dessus de p_i dans $\mathcal{O}_\Delta$, et par $\left(\frac{K_D/K}{\mathfrak{p}_i}\right)$ son symbole d'Artin suivant la définition 1.11. Alors,*

$$\left(\frac{K_D/K}{\mathfrak{p}_i}\right)(w_{p_1,p_2}(\tau)) = \frac{1}{w_{p_1,p_2}(\tau)}.$$

La démonstration se fait encore en utilisant la loi de réciprocité de Shimura et le comportement de η sous transformations unimodulaires.

Une conséquence simple du lemme est que w_{p_1,p_2} est invariant sous l'automorphisme de K_D associé à $\mathfrak{p}_1\mathfrak{p}_2$, de sorte que la valeur singulière se trouve dans le sous-corps de K_D fixé par $\langle\mathfrak{p}_1\mathfrak{p}_2\rangle$, ou de manière équivalente dans le sous-corps de K_D de groupe de Galois $\mathrm{Cl}_D/\langle\mathfrak{p}_1\mathfrak{p}_2\rangle$ sur K.

On vérifie aisément que pour $D = -kN$ impair, la forme réduite équivalente à Q se trouve parmi $\left[N, N, \frac{N+k}{4}\right]$, $\left[k, k, \frac{N+k}{4}\right]$ ou $\left[\frac{N+k}{4}, \left|\frac{N-k}{2}\right|, \frac{N+k}{4}\right]$; quand $D = -4kN$ est pair, elle est soit $[N, 0, k]$, soit $[k, 0, N]$. On en déduit que Q (ou de façon équivalente $\mathfrak{p}_1\mathfrak{p}_2$) est principal si et seulement si $k = 1$. Cela démontre le résultat suivant :

Théorème 1.29 *Si en plus des hypothèses du lemme 1.28, D est différent de $-N$ et de $-4N$, alors la valeur singulière $w_{p_1,p_2}(\tau)$ se trouve dans un sous-corps d'indice 2 de K_D.*

Comme dans le théorème 1.22, les conjuguées de la valeur singulière s'obtiennent à partir d'un N-système. Le lemme 1.28 permet en plus de prédire quelles sont les deux formes donnant la même conjuguée : soit $\mathfrak{a}$ un idéal correspondant à une forme du N-système, et $\mathfrak{n}$ l'idéal correspondant à Q. Alors les valeurs singulières pour $\mathfrak{a}$ et $\mathfrak{an}$ sont les mêmes. D'après la discussion suivant le théorème 1.22, les conjuguées complexes en sont données par $\mathfrak{a}^{-1}\mathfrak{n}$ et $\mathfrak{a}^{-1}$. Cette observation permet de réduire le nombre d'évaluations de w_{p_1,p_2} d'approximativement $\frac{h_D}{2}$ à $\frac{h_D}{4}$. La hauteur du polynôme sera également divisée par 2, de sorte qu'on peut s'attendre à gagner un facteur au moins 4 dans le temps de calcul.

Notons qu'on peut obtenir ce résultat directement à partir du théorème 1.22 sans réutiliser la loi de réciprocité de Shimura. Partons pour cela du N-système servant à déterminer les conjuguées de la valeur singulière et satisfaisant $N|C$ (ce qui est demandé dans les théorèmes pour une forme, mais alors c'est vrai pour toutes les formes grâce aux propriétés d'un N-système). Considérons l'application qui envoie une forme $Q_1 = [A, B, C]$ sur la forme $Q_2 = \left[\frac{C}{N}, -B, AN\right]$. Sous les hypothèses du théorème 1.29, on a $B \equiv 0 \bmod N$ à cause de $N|D$, et donc $-B \equiv B \bmod 2N$; et $\operatorname{pgcd}\left(\frac{C}{N}, N\right) = 1$ à cause de $\operatorname{pgcd}(f, N) = 1$. La forme Q_2 satisfait donc les conditions de la définition 1.17, et on peut supposer qu'elle apparaît également dans le N-système. De plus, Q_1 et Q_2 ne sont pas équivalentes ; en fait, l'une est la multiplication de l'autre par Q, qui n'est pas principal. Notons par τ_i les racines respectives de Q_i ; l'application $\tau_1 \mapsto \tau_2 = \frac{-N}{\tau_1}$ n'est rien d'autre que l'involution de Fricke–Atkin–Lehner déjà évoquée à la section 1.2.3. Comme w_{p_1,p_2} est invariante sous cette involution, Q_1 et Q_2 donnent les mêmes valeurs singulières, et plus généralement, les conjuguées viennent en couples.

Le théorème 1.29 peut donc se généraliser à d'autres fonctions de classes, pourvu qu'elles soient invariantes sous l'involution de Fricke–Atkin–Lehner associée à $\Gamma^0(N)$.

Une autre possibilité de généralisation, déjà décrite dans [67], est de passer à un niveau N avec trois ou plusieurs facteurs. Soit pour un troisième nombre premier impair p_3 le quotient triple de η donné par

$$w_{p_1,p_2,p_3} = \frac{w_{p_1,p_2}(z)}{w_{p_1,p_2}\left(\frac{z}{p_3}\right)} = \frac{\eta\left(\frac{z}{p_1}\right)\eta\left(\frac{z}{p_2}\right)\eta\left(\frac{z}{p_3}\right)\eta\left(\frac{z}{p_1p_2p_3}\right)}{\eta(z)\,\eta\left(\frac{z}{p_1p_2}\right)\eta\left(\frac{z}{p_1p_3}\right)\eta\left(\frac{z}{p_2p_3}\right)}.$$

C'est une fonction modulaire pour $\Gamma^0(N)$ avec $N = p_1p_2p_3$. Supposons les trois premiers distincts et divisant le discriminant fondamental sans diviser le conducteur, et une forme quadratique de dernier coefficient égal à N. Alors on peut montrer que la valeur singulière

de w_{p_1,p_2,p_3} dans la forme est invariante sous les symboles d'Artin associés à $\mathfrak{p}_1\mathfrak{p}_2$ et à $\mathfrak{p}_1\mathfrak{p}_3$ (et à leur produit $\mathfrak{p}_2\mathfrak{p}_3$), où les $\mathfrak{p}_i$ sont les idéaux ramifiés au-dessus des p_i. Si le sous-groupe $\langle \mathfrak{p}_1\mathfrak{p}_2, \mathfrak{p}_1\mathfrak{p}_3 \rangle$ du groupe de classes est d'ordre 4, alors la valeur singulière vit dans un sous-corps d'indice 4 du corps de classes ; c'est le cas si et seulement si ni $\mathfrak{p}_1\mathfrak{p}_2$, ni $\mathfrak{p}_1\mathfrak{p}_3$ ni $\mathfrak{p}_2\mathfrak{p}_3$ ne sont principaux. Pour un idéal de départ $\mathfrak{a}$, les quadruples de formes donnant la même conjuguée correspondent aux idéaux $\mathfrak{a}$, $\mathfrak{a}\mathfrak{p}_1\mathfrak{p}_2$, $\mathfrak{a}\mathfrak{p}_1\mathfrak{p}_3$ et $\mathfrak{a}\mathfrak{p}_2\mathfrak{p}_3$; la conjuguée complexe s'obtient à partir de $\mathfrak{a}^{-1}\mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3$, $\mathfrak{a}^{-1}\mathfrak{p}_1$, $\mathfrak{a}^{-1}\mathfrak{p}_2$ et $\mathfrak{a}^{-1}\mathfrak{p}_3$.

La généralisation de ce résultat à plus de trois nombres premiers est évidente.

À ma connaissance, c'est la première fois que des sous-corps de corps de classes ont été obtenus directement par des valeurs singulières de fonctions modulaires.

Notons encore un lien avec la décomposition de Galois du chapitre 1.4, induit encore une fois par l'invariance de w_{p_1,p_2} sous l'involution de Fricke–Atkin–Lehner. Dans le cas du théorème 1.29 elle s'écrit sur les classes d'idéaux comme $\mathfrak{a} \mapsto \mathfrak{a}\mathfrak{n}$ avec $\mathfrak{n}$ l'idéal de norme N d'ordre 2. Soit $\Phi(X,Y)$ le polynôme modulaire s'annulant en (w_{p_1,p_2}, j) (voir le chapitre 3). Comme $w_{p_1,p_2}(\mathfrak{a}) = w_{p_1,p_2}(\mathfrak{a}\mathfrak{n})$, la spécialisation $\overline{\Phi} = \Phi(w_{p_1,p_2}(\mathfrak{a}), Y)$ a au moins les deux racines $j(\mathfrak{a})$ et $j(\mathfrak{a}\mathfrak{n})$; en faisant varier $\mathfrak{a}$ parmi les $\frac{h_D}{2}$ idéaux du demi-N-système, on récupère ainsi les h_D valeurs singulières de j. Dans le cas particulier que le degré de Φ en Y est 2, qui se produit d'après le théorème 3.5 pour $(p_1,p_2) = (3,13)$ ou $(3,5)$, le corps de classes est alors donné par la tour de corps engendrée par le demi-polynôme de classes $\sqrt{H_D[w_{p_1,p_2}](X)}$ de degré $\frac{h_D}{2}$ et le polynôme modulaire $\Phi(X,Y)$ de degré 2. C'est un cas particulier de la décomposition galoisienne du corps de classes vue au chapitre 1.4, dans laquelle le dernier polynôme ne dépend pas du discriminant.

1.7 Réalisations logicielles

1.7.1 mpc

La bibliothèque `mpc` *[71] a été développée en collaboration avec Paul Zimmermann.*

La première brique de base pour implanter le calcul de polynômes de classes par approximations flottantes est l'arithmétique des nombres complexes à grande précision. C'est le but de la bibliothèque `mpc`, implantée en C au-dessus de `mpfr` [105] qui se sert de `gmp` [99] pour l'arithmétique des grands entiers.

`mpfr` permet des calculs avec des réels flottants de précision arbitraire, définie au bit près pour chaque variable, tout en suivant la sémantique des flottants à double précision spécifiée dans les normes IEEE. Essentiellement, elle garantit l'arrondi correct des opérations élémentaires. Les entrées des fonctions et opérations sont considérées comme des nombres exacts, c'est-à-dire des rationnels avec une puissance de 2 comme dénominateur. La sortie correspond au résultat exact, arrondi au choix de l'utilisateur à un nombre représentable avec la précision demandée, soit au nombre le plus proche, soit vers zéro, soit vers l'infini, soit vers moins infini.

`mpc` est une implantation de l'arithmétique complexe suivant les mêmes principes que `mpfr`. Un nombre complexe flottant est représenté par sa partie réelle et sa partie complexe, toutes les deux de types `mpfr`, et pouvant être de précision différente. La

sortie d'une fonction ou opération correspond au résultat exact arrondi vers un nombre représentable, les parties réelle et imaginaire pouvant être arrondies séparément selon les modes spécifiés pour `mpfr`.

Pour rendre la multiplication plus performante, on pourrait imaginer passer à une représentation sous coordonnées polaires. Malheureusement, celle-ci ferait perdre plus pour l'addition qu'elle ne fait gagner pour la multiplication, et n'a donc pas été retenue. Un autre point de critique est que l'arrondi séparé des parties réelle et imaginaire n'est pas forcément très intéressant d'un point de vue pratique. Pour beaucoup d'applications (dont la multiplication complexe) il serait suffisant de garantir l'erreur relative des opérations. Par exemple, à précision n bits, on pourrait demander que l'erreur d'arrondi, divisée par la valeur absolue du résultat, ne dépasse pas 2^{-n}. Surtout dans les cas où le résultat exact est réel, cela éviterait de perdre du temps à calculer une partie imaginaire à grande précision. Le grand inconvénient de cette approche est que sa sémantique est ambiguë : le résultat d'une opération n'est plus bien défini quand les parties réelle et imaginaire ont des ordres de grandeurs distincts. Par exemple, le réel 1 admettrait un nombre infini d'approximations à n bits par $1 + 2^{-m}i$ pour $m \geqslant n$.

La bibliothèque `mpc` est distribuée sous la licence libre LGPL. Elle est à la base des nombres complexes dans le système de calcul formel `magma`, et sous considération d'être utilisée (comme `mpfr`) dans `gcc` pour évaluer des expressions flottantes à la compilation. Actuellement, `mpc` fait l'objet d'une opération de développement logiciel de l'INRIA.

1.7.2 mpfrcx

Toujours du domaine plutôt du calcul formel que de la théorie algorithmique des nombres, une arithmétique performante de polynômes à coefficients flottants est également au cœur des algorithmes de la multiplication complexe. L'une des applications en est le calcul du polynôme de classes à partir de ses racines par multiplication successive des facteurs linéaires. De façon moins évidente, elle apparaît dans les divers algorithmes rapides exposés tout au long du chapitre 1, dans la multiévaluation à la section 1.5.2 comme dans le calcul d'une représentation de Hecke à la section 1.4.1.

J'ai écrit une bibliothèque en C, `mpfrcx`, pour l'arithmétique des polynômes à coefficients dans `mpfr` et `mpc`. Une partie du code s'applique *mutatis mutandis* aux polynômes réels et complexes ; elle est créée par application automatique de `sed`, une façon simple de faire des génériques en C. Il s'agit notamment de la multiplication à la Karatsuba ou Toom–Cook, ou des divisions avec reste rapide par itérations de Newton modulo des puissances croissantes de la variable.

Pour l'application à la multiplication complexe, les calculs avec les polynômes ont essentiellement été réduits aux coefficients réels grâce au regroupement des couples de racines conjuguées complexes décrit aux sections 1.2.3 et 1.6, et à la décomposition galoisienne de 1.4.2. Quand la transformée de Fourier rapide (FFT) entre en jeu, par contre, un passage aux complexes est nécessaire pour pouvoir disposer des racines de l'unité. `mpfrcx` contient plusieurs variantes de la FFT et de son inverse selon le livre de Nussbaumer [149], soit avec la sortie dans le bon ordre (ce qui nécessite l'application d'une permutation supplémentaire après la récursion), soit dans l'ordre de bit inversé

pour économiser cette permutation. Dans ce dernier cas, l'entrée de la transformée qui devrait se trouver à l'indice $\sum_{i=0}^{n} c_i 2^i$ du tableau est stockée à l'endroit $\sum_{i=0}^{n} c_i 2^{n-i}$, ce qui demande d'adapter la transformation inverse. Pour l'instant, c'est la transformation à la Rader–Brenner qui s'est avérée être la plus efficace ; au lieu de multiplier par des racines de l'unité, elle multiplie par leur trace réelle, une valeur du cosinus. La FFT est activée dès que les degrés des polynômes à multiplier sont au moins 512.

Pour la multiplication de polynômes réels, la FFT des deux polynômes de départ est calculée par une seule FFT complexe ; son inverse est obtenue par une FFT complexe de l'ordre la moitié du degré du résultat.

D'autres algorithmes implantés concernent la multiévaluation de la section 1.5.2.

Je projette de publier ces bibliothèques également sous licence libre une fois atteinte la stabilité nécessaire.

1.7.3 cm

Les astuces et expériences d'implantation de cette section sont tirées de [64], écrit avec François Morain, et de [58].

La section 5 de [64] porte sur l'accélération de l'évaluation des fonctions modulaires en passant par la série creuse pour η. On observe premièrement que si τ est la racine d'une forme quadratique de discriminant D apparaissant dans un N-système selon la définition 1.17, alors c'est encore vrai pour les arguments transformés $\frac{\tau}{p_1}$, $\frac{\tau}{p_2}$ et $\frac{\tau}{p_1 p_2}$ dans lesquels il faut évaluer η pour les quotients doubles de la section 1.2.3. Quand le degré de transformation ℓ est impair, ce raisonnement s'applique également aux quotients simples de la section 1.2.2 (tandis que pour les fonctions de Weber de degré de transformation 2, on atterrit souvent dans un ordre de conducteur modifié d'un facteur 2). Ainsi, au lieu de devoir évaluer η dans $4h_D$ ou $2h_D$ arguments, on peut se contenter des h_D valeurs dans les formes réduites, et utiliser le comportement de η sous transformations unimodulaires pour les valeurs dans d'autres formes. L'observation que la forme inverse résulte en la conjuguée complexe de la valeur de η permet en plus de se limiter à presque $\frac{h_D}{2}$ évaluations.

Pour la série creuse, il a été argumenté à la section 1.5.2 que 4 multiplications donnent 2 nouveaux termes par l'approche des différences itérées. En essayant de systématiquement écrire chaque terme comme le produit de deux ou trois termes précédents de la série, j'ai observé qu'en moyenne, 5 multiplications donnent 4 nouveaux termes, un gain de près d'un facteur 2 par rapport à l'algorithme générique. Je n'ai pas réussi à expliquer ce comportement par des propriétés arithmétiques de la série pour η, et je le suspecte d'être une simple conséquence de la densité des termes non nuls.

Finalement, on remarque que l'exponentiation complexe $z \mapsto q = e^{2\pi i z}$ prend un temps non négligeable par rapport à l'évaluation de la série en q. Nous utilisons le fait que les arguments rencontrés ont une forme spéciale pour accélérer les calculs. Notamment, pour $z = \frac{-B+\sqrt{D}}{2A}$, on a

$$q = \varrho_A \, \zeta_{A,B} \text{ avec } \varrho_A = e^{-\pi\sqrt{|D|}/A} \text{ et } \zeta_{A,B} = e^{-\pi i B/A}.$$

L'observation que si $A'|A$, alors $\varrho_{A'} = \varrho_A^{A/A'}$, permet de remplacer des exponentiations réelles par des multiplications, plus rapides. Des cosinus réels sont économisés en récrivant $\zeta_{A,B} = \zeta_{A/\operatorname{pgcd}(A,B),B/\operatorname{pgcd}(A,B)}$.

Ainsi optimisée, l'évaluation de la série creuse pour η est étonnamment compétitive par rapport aux algorithmes quasi-linéaires asymptotiquement. En témoigne le tableau 1.2, tiré de [58], qui donne le temps de calcul (en secondes sur des AMD Opteron 250 cadencés à 2,4 GHz) pour des polynômes de classes $H_D[w_{3,13}]$ obtenu avec mon implantation `cm` en utilisant les différentes techniques d'évaluation de η évoquées à la section 1.5.2.

	h_D	5 000	10 000	20 000	40 000	100 000
	$\lvert D\rvert$	6 961 631	23 512 271	98 016 239	357 116 231	2 093 236 031
(1)	précision n (bits)	9 540	20 317	45 179	96 701	264 727
(2)	hauteur (en base 2)	8 431	18 114	40 764	87 842	242 410
(3)	$M(n)$	7,3	23	75	230	1 080
(4)	groupe de classes	0,1	0,1	0,4	1,3	6,8
(5)	conjuguées depuis η	3,4	21	140	890	10 000
(6)	poly. des conjuguées	13	93	730	5 200	120 000
	série creuse					
(7)	η	12	98	900	7 700	140 000
(8)	dont les q	3,0	22	170	1 300	20 000
(9)	temps total	28	210	1 800	14 000	270 000
	multiévaluation					
(10)	η	93	640	5 700	42 000	arrêté
(11)	temps total	110	750	6 500	48 000	
	(10) / (7)	7,8	6,5	6,3	5,5	—
	AGM					
(12)	η	32	200	1 400	9 900	130 000
(13)	temps total	48	320	2 300	16 000	260 000
	(12) / (7)	2,7	2,0	1,6	1,3	0,93

TAB. 1.2 – Temps de calcul pour les polynômes de classes

Le bloc des lignes (4) à (6) donne le temps de calcul pour les pas qui ne dépendent pas de l'algorithme utilisé pour l'évaluation de η. La ligne (4) montre qu'effectivement en pratique, le calcul du groupe de classes est négligeable même avec l'algorithme le plus naïf qui soit. Les lignes (5) et (6) correspondent au temps passé pour calculer les quotients doubles à partir des valeurs tabulées pour η et au temps de la multiplication de tous les facteurs linéaires pour reconstituer le polynôme de classes à partir de ses racines ; ce dernier se fait sentir bien que la FFT soit déployée.

On voit que la multiévaluation, bien qu'asymptotiquement plus rapide, n'arrive pas à lutter contre la série creuse pour des instances maniables ; l'évaluation par AGM, dont le code m'a été donné par Dupont, ne gagne qu'à la dernière colonne. Cette colonne, avec un nombre de classes de 100 000, constitue un record qui dépasse de loin ce qui a été fait par l'approche p-adique dans [26]. Pour faire tenir les calculs dans la mémoire de

la machine, la FFT a dû être désactivée pour les multiplications de polynômes à partir d'un degré 16 384, ce qui ralentit considérablement la reconstruction du polynôme depuis ses racines. Néanmoins, le temps de calcul total de trois jours reste très raisonnable. On observe ici le comportement typique d'un algorithme quasi-linéaire en la taille de sa sortie ; en fin de compte, la limitation provient de la mémoire plutôt que du temps de calcul.

Le logiciel `cm` qui implante ce calcul de courbes elliptiques à multiplication complexe n'a pas encore été publié. Néanmoins, il a déjà servi à fournir les courbes utilisées dans [152] pour comparer la performance de différents cryptosystèmes fondés sur les couplages, cf. le chapitre 2. Je projette de mettre à disposition publique au moins un exécutable binaire.

1.8 Perspectives

Les aboutissements de plusieurs années de recherche décrits tout au long de ce chapitre permettent de bien maîtriser les algorithmes de la multiplication complexe pour les courbes elliptiques. Notamment grâce à la famille infinie de fonctions de classes de la section 1.2.3, nous pouvons trouver pour n'importe quel discriminant une fonction avec un polynôme de classes raisonnablement petit, gagnant un facteur d'au moins 12 par rapport au polynôme classique $H_D[j]$. Grâce à l'algorithme de complexité quasi-linéaire de la section 1.5, ce polynôme se calcule rapidement, le facteur limitant étant devenu la taille de la sortie. Et la décomposition galoisienne de la section 1.4, également quasi-linéaire, permet de maîtriser la factorisation du polynôme sur le corps fini quand celle-ci devient le goulot d'étranglement. Autrement dit, les possibilités de progrès seront limitées : ce qui sera calculable demain, l'est essentiellement déjà aujourd'hui.

Néanmoins, il y a des perspectives d'amélioration dans les détails. La recherche des meilleurs fonctions de classes n'est pas finie, notamment pour les quotients simples de η de niveau ℓ composé, et concernant les plus petites puissances d'une fonction utilisables pour un discriminant donné.

Concernant la vitesse de l'évaluation, on note que les séries ϑ (par exemple $\vartheta_0(z) = 1 + 2\sum_{n\geqslant 0} q^{n^2/2}$) peuvent être encore plus creuses que la série pour η, et que ϑ_0 est également plus proche de l'AGM. Il serait donc intéressant de bâtir une théorie de fonctions de classes sur ϑ, un sujet bien adapté pour un étudiant.

Le record de [58] pour un nombre de classes $h_D = 100\,000$, évoqué à la section 1.7.3, semble avoir établi l'idée que les polynômes de classes se calculent aisément pour tous les nombres de classes jusqu'à 10^5 et, en s'appuyant sur le théorème de Siegel $\log|D| \sim 2\log h_D$, pour tous les discriminants fondamentaux $|D|$ jusqu'à 10^{10}, cf. la discussion dans [76, section 2].

Or, les nombres de classes rencontrés pour $|D| < 10^{10}$ dépassent bien 10^5 ; par exemple, $h_{-9\,999\,815\,591} = 222\,948$. Inversement, des discriminants bien plus grands peuvent avoir un nombre de classes autour de 10^5, résultant en une hauteur nettement plus élevée que pour le record, dont le discriminant n'était finalement que d'environ $2 \cdot 10^9$. Par exemple, $h_{-(10^{12}+20\,427)} = 69\,737$. J'aimerais bien à court terme adapter le

logiciel `cm` pour qu'il puisse effectivement traiter tous ces discriminants.

Un inconvénient plus métaphysique que pratique du calcul de polynômes de classes par approximations flottantes est que le résultat n'est pas démontré correct, à cause des erreurs d'arrondi potentielles. (En pratique, on est généralement plus intéressé par une courbe elliptique à multiplication complexe choisie que par le polynôme de classes lui-même ; mais l'anneau d'endomorphismes se vérifie indépendamment du polynôme par l'algorithme de Kohel [123].) Une vérification probabiliste des polynômes de classes est aisée : si la réduction modulo des nombres premiers fournit des courbes elliptiques avec le bon anneau d'endomorphismes, c'est une bonne indication que le polynôme est juste. Il serait intéressant d'obtenir des preuves de correction irréfutables, de préférence sous forme de certificats tels qu'ils sont connus pour les problèmes dans NP. Le temps d'obtention de tels certificats et surtout le temps de vérification devraient être moindres que le calcul du polynôme, un défi rendu d'autant plus difficile par l'algorithme quasi-linéaire de la section 1.5.

Concernant la construction directe de sous-corps de corps de classes par des valeurs singulières de fonctions modulaires, le théorème 1.29 n'est qu'un premier résultat. L'observation faite ici que le point crucial est l'invariance sous l'involution de Fricke–Atkin–Lehner pourrait ouvrir la porte vers d'autres résultats du même genre.

Finalement, la multiplication complexe en genre supérieur demande encore beaucoup de travail. Le record actuel en genre 2 est obtenu pour un nombre de classes de 50 dans [87]. Pour aller plus loin, il faudrait sans doute développer des fonctions de classes susceptibles de remplacer les invariants d'Igusa.

2 Cryptographie fondée sur l'identité

> Identité, *s. f. (Métaphysiq.) l'identité d'une chose est ce qui fait dire qu'elle est la même & non une autre.*
> — Encyclopédie de Diderot et d'Alembert

L'une des idées reçues sur la cryptographie à clef publique est qu'elle résout les problèmes liés aux échanges de clefs. Dans les cryptosystèmes *à clef secrète* ou *symétriques*, la même clef est utilisée pour deux opérations cryptographiques inverses, telles que le chiffrement et le déchiffrement d'un message, ou l'ajout d'un code d'authentification à un message et la vérification de sa validité. Ainsi, chaque couple de participants au système doit au préalable se mettre d'accord sur une clef secrète, ce qui nécessite la mise en place d'un canal sécurisé dédié. L'effort de gestion des clefs croît ainsi quadratiquement avec le nombre de participants à l'infrastructure cryptographique.

Dans un cryptosystème *à clef publique* ou *asymétrique*, par contre, les clefs ne dépendent plus des deux parties voulant communiquer, mais sont liées à seulement une personne ou *acteur*. En revanche, tout acteur dispose d'un *couple de clefs*. La *clef publique* est rendue publique comme l'indique son nom ; elle sert pour les opérations exécutées par tous ceux qui veulent communiquer avec son propriétaire, par exemple en lui envoyant des messages chiffrés ou en voulant vérifier les signatures sur ses messages. La *clef privée* doit le rester, car c'est sur elle que repose toute la sécurité du système ; elle sert à son détenteur légitime pour déchiffrer les messages qui lui sont adressés ainsi qu'à signer les messages qu'il émet. À première vue, comme le nombre de clefs est linéaire en le nombre de participants au système, l'effort pour les gérer devrait l'être aussi.

Mais c'est se tromper sur la sécurité demandée au système. De nos jours, on demande à un bon cryptosystème de venir avec sa *preuve de sécurité* (au sens de l'informatique théorique ; nous y revenons à la section 2.2.2). Ainsi, le fait d'arriver à vérifier une signature à l'aide d'une clef publique démontre (ou au moins donne de la confiance en la supposition) que celui qui a signé le message possédait la clef privée correspondante. Mais c'est sans intérêt pratique ; ce qu'on voudrait, c'est de montrer qu'un certain acteur

a fait la signature et ainsi ratifié le contenu. Il manque donc un lien entre l'acteur et la clef privée ou bien la clef publique (le lien entre les clefs privée et publique étant créé mathématiquement), ou autrement dit, une preuve de *l'authenticité* de la clef. On a alors l'impression de se retrouver à la case départ : chaque détenteur d'une clef publique doit l'envoyer à tout autre participant au système par le moyen d'un canal sûr. Néanmoins, les prérequis au canal ont changé ; tandis qu'il devrait être *secret* et *authentifié* dans le cas de la cryptographie symétrique, seule l'authenticité est nécessaire pour la cryptographie asymétrique.

La solution à ce problème généralement acceptée dans la cryptographie à clef publique est d'introduire un tiers de confiance, appelé *autorité de certification*, qui est chargé de vérifier les identités des acteurs et de certifier leurs clefs publiques, c'est-à-dire de signer un document appelé *certificat* contenant à la fois l'identité et la clef publique. Le formalisme retenu est celui des certificats X.509, standardisés par l'IETF dans le RFC 2459 [114], suffisamment complexe pour avoir besoin d'un « guide de style X.509 » [102]. L'existence d'une autorité de certification ne résout pas le problème, mais le fait ressurgir au niveau supérieur : si un certificat est essentiellement une signature, comment assurer l'authenticité de la clef publique de l'autorité de certification ? Ainsi ont été introduites des hiérarchies d'autorités de certification, appelées *infrastructures de clefs publiques (PKI)*, sous forme arborescente (ou plus généralement sous forme d'un graphe orienté). Après avoir vérifié une clef manuellement, on peut avoir confiance en toutes les clefs dans le sous-graphe avec la clef vérifiée comme racine. On voit que le système commence à devenir lourd, et pire, il ne fonctionne pas : par exemple, l'utilisateur commun n'a aucun moyen de vérifier un certificat inconnu que son navigateur web lui présente, et l'accepte en général sans hésiter.

Mentionnons également le problème de la *révocation de clefs*. Dans la vraie vie, des clefs privées peuvent être compromises, tout comme des cartes bancaires ; il faut à ce moment répandre la nouvelle et retirer la clef publique correspondante de la circulation. Cela se fait à l'aide de *listes de révocation de certificats (CRL)*, spécifiées également dans le RFC [114]. Idéalement, il faudrait consulter ces listes avant chaque utilisation d'une clef publique, ce qui rendrait toute la cryptographie impraticable.

La cryptographie fondée sur l'identité propose une solution alternative et originale à ce problème ; essentiellement en authentifiant la clef privée au lieu de la clef publique. À ce jour le seul moyen d'obtenir de tels systèmes à la fois sûrs et efficaces est de passer par les courbes algébriques, et plus particulièrement des couplages définis avec de telles courbes.

Les domaines de la cryptographie fondée sur l'identité et les couplages sont devenus tellement vastes qu'il ne m'est pas possible d'en faire un survol. Dans les sections suivantes, je me contenterai de donner les quelques informations de base qui me permettront de parler de mes propres contributions. La première partie de la thèse [129] de Libert, dont j'étais membre du comité d'encadrement, donne d'ailleurs une excellente introduction à ces sujets, avec l'accent sur la sécurité prouvée.

2.1 Historique

2.1.1 Concepts de la cryptographie fondée sur l'identité

L'idée de la cryptographie fondée sur l'identité a été présentée par Shamir dans [171] en remplaçant une clef publique plus ou moins aléatoire tout simplement par l'identité de l'acteur. Ainsi, il n'est pas nécessaire de la faire certifier. Par contre, le propriétaire de la clef publique ne peut plus obtenir sa clef privée tout seul : s'il arrivait à la calculer avec rien d'autre que son identité comme point de départ, n'importe qui pourrait également la calculer, et le système ne présenterait aucune sécurité. À la place, sa clef privée lui est fournie par une tierce instance, appelée le *générateur de clefs privées (PKG)*. Le PKG dispose d'une clef maîtresse, et cette information privilégiée le met en mesure d'obtenir les clefs privées de tout le monde. Le rôle d'un PKG est assez proche d'une autorité de certification classique. Mais au lieu de certifier des clefs publiques, il distribue des clefs privées ; ainsi, au lieu d'avoir besoin d'un canal authentique avec l'autorité de certification, il faut désormais un canal secret et authentifié avec le PKG. Le fait que le PKG connaît forcément les clefs de tous les participants est un problème de sécurité qui peut être vu comme un atout dans l'environnement d'une entreprise et qui peut également plaire aux agences gouvernementales. Pour y remédier, il est possible de distribuer le calcul des clefs privées sur plusieurs PKG, dont un certain nombre devrait collaborer pour apprendre le secret. Une autre solution a été proposée par Al-Riyami et Paterson dans [5]. Dans leur modèle de *cryptographie sans certificats*, chaque participant se fait aider par le PKG pour dériver sa clef privée, qu'il sera le seul à connaître.

Notons comme application de la cryptographie fondée sur l'identité une solution simple à la révocation de clefs : il suffit d'ajouter à l'identité une période de validité et de demander aux acteurs de contacter périodiquement le PKG pour obtenir la clef privée en cours de validité. Évidemment, cette solution a son prix, qui peut être ajusté en fonction de la sécurité souhaitée en choisissant des périodes plus ou moins longues.

2.1.2 Systèmes sans couplages

Le système fondé sur l'identité donné par Shamir dans [171] est un schéma de signature numérique à base du problème RSA. Un autre exemple est donné par la signature dérivée du schéma d'identification de Guillou et Quisquater [101]. Mais c'est plutôt le chiffrement fondé sur l'identité qui présente le plus d'intérêt. En effet, toute signature classique avec PKI peut être trivialement transformée en une signature fondée sur l'identité : il suffit d'augmenter la signature de tous les certificats formant un chemin de certification jusqu'à l'identité du signataire.

Bien que non formulé dans ce cadre, le protocole non-interactif de distribution de clefs donné par Maurer et Yacobi dans [135] s'étend trivialement à un système de chiffrement fondé sur l'identité. Le secret maître du PKG y est la factorisation d'un entier RSA n. Pour obtenir des clefs secrètes, il faut calculer des logarithmes discrets modulo n, ce qui peut se faire en temps essentiellement $O(\sqrt{q})$ par la méthode ϱ de Pollard, où q est le plus grand facteur de $p-1$ pour p un premier divisant n (voir le chapitre 4) ;

alternativement, des algorithmes sous-exponentiels de complexité $L_p(1/2)$ peuvent être utilisés. Un attaquant doit factoriser n, ce qui peut se faire en temps $O(q)$ par la méthode $p-1$. Ainsi, la différence entre l'effort d'un attaquant et du PKG est assez faible ; pour arriver à un système sûr, il faut choisir des paramètres de telle taille que le système est considéré trop inefficace pour être déployé.

Le premier système de chiffrement fondé sur l'identité, explicitement publié en tant que tel, est dû à Cocks en 2001 [31]. Il se sert implicitement de signatures à la Rabin et est raisonnablement efficace pour le PKG. Par contre, il ne chiffre qu'un bit à la fois, pour lequel il exige la transmission de deux entiers de la taille d'un module RSA, donc d'au moins 1024 bits. Bien que peu efficace de ce point de vue, il pourrait servir pour envoyer des clefs symétriques dans un système hybride.

2.1.3 Couplages dans les courbes elliptiques

Dans la suite, nous nous intéressons à des courbes elliptiques E définies sur un corps fini $\mathbb{F}_q$ de caractéristique p. Supposons que n est un entier premier avec p. Une telle courbe admet essentiellement deux *couplages*

$$e : E[n] \times E[n] \to \mu_n \subseteq \mathbb{F}_{q^k}^\times.$$

Ici, $E[n]$ sont les points de n-torsion sur la courbe, qui peuvent être définis sur une extension $\mathbb{F}_{q^k}$ de $\mathbb{F}_q$ et qui forment un groupe de type $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$, et μ_n sont les racines n-èmes de l'unité, qui vivent dans le même corps $\mathbb{F}_{q^k}$. Dans la suite, k est supposé au moins 2. Un couplage est une telle application qui est *bilinéaire* entre $\mathbb{Z}/n\mathbb{Z}$-modules, ou autrement dit homomorphe dans les deux arguments en tant qu'application entre groupes, et *non dégénérée* : pour tout $P \in E[n]$, il doit exister un $Q \in E[n]$ tel que $e(P,Q) \neq 1$, et *vice versa*.

Le *couplage de Weil* est directement défini sur les points de n-torsion ; il est compatible avec les endomorphismes de la courbe et antisymétrique (de sorte que $e(P,P) = 1$ pour tout $P \in E[n]$, ce qui peut être indésirable).

Le *couplage de Tate–Lichtenbaum* peut être défini comme application bilinéaire

$$e' : E[n] \times E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^\times/(\mathbb{F}_{q^k}^\times)^n.$$

Quand $\operatorname{pgcd}\left(\frac{|E(\mathbb{F}_{q^k})|}{n^2}, n\right) = 1$ ou de façon équivalente $E(\mathbb{F}_{q^k}) \cap E[n^2] = E[n]$, on peut identifier $E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$ avec $E[n]$ via l'homomorphisme surjectif

$$E(\mathbb{F}_{q^k}) \to E[n], \quad P \mapsto \frac{|E(\mathbb{F}_{q^k})|}{n^2}\, P$$

de noyau $nE(\mathbb{F}_{q^k})$. Élevant de même les valeurs du couplage e' à la puissance $\frac{q^k-1}{n}$ quand $\operatorname{pgcd}\left(\frac{q^k-1}{n}, n\right) = 1$, on arrive au couplage dit *réduit*

$$e : E[n] \times E[n] \to \mu_n.$$

Étant asymétrique par définition, ce couplage ne souffre pas du problème que $e(P,P)=1$ pour tous les points de n-torsion P. Néanmoins, comme les deux couplages se calculent algébriquement depuis les coefficients de la courbe et des points, on a toujours $e(P,P)=1$ si P est défini sur $\mathbb{F}_q$ au lieu de $\mathbb{F}_{q^k}$.

Les deux couplages s'obtiennent avec $O(\log n)$ opérations dans $\mathbb{F}_{q^k}$, essentiellement en multipliant un point par n et en gardant la trace des droites utilisées dans la loi de groupe de la courbe elliptique. Ils se généralisent à d'autres courbes et variétés algébriques. Pour plus de détails sur les couplages, voir [33, chapitre 6] ou [83].

2.1.4 Cryptologie et couplages

Les premières applications des couplages en cryptologie étaient cryptanalytiques. Un couplage transporte en fait le problème du logarithme discret de la courbe dans le groupe multiplicatif du corps fini $\mathbb{F}_{q^k}$, et ce de façon effective. Si k, le *degré de plongement* ou *degré MOV*, est suffisamment petit, le logarithme discret se calcule par un algorithme sous-exponentiel dans le corps fini, ce qui est à opposer à la complexité *a priori* exponentielle sur la courbe (voir aussi le chapitre 4). C'est systématiquement le cas pour les courbes supersingulières comme observé dans [136] en utilisant le couplage de Weil et dans [78] en utilisant le couplage de Tate–Lichtenbaum.

La première utilisation constructive a été proposée par Joux dans [118]. Comme la multiplication par des scalaires dans un groupe, qui est une application linéaire, rend possible l'échange de clefs à la Diffie–Hellman parmi deux participants, l'argument supplémentaire dans une application bilinéaire permet de réaliser une échange de clefs entre trois acteurs. Chacun choisit un entier a_i et publie a_iP et a_iQ, où P et Q sont deux points tels que $e(P,Q)\neq 1$ (le cas $P=Q$ étant possible pour le couplage de Tate–Lichtenbaum). L'acteur i utilise son propre secret a_i et deux points publiés par ses partenaires, par exemple a_jP pour $j\neq i$ et a_kQ pour $k\neq i,j$ afin de calculer la clef partagée

$$e(P,Q)^{a_1a_2a_3}=e(a_jP,a_kQ)^{a_i}.$$

Le troisième paramètre introduit par les couplages a ensuite été utilisé par Sakai, Ohgishi et Kasahara en tant que clef maîtresse d'un PKG, pour arriver à un système d'échange de clefs sans interaction fondé sur l'identité [158], qui sera discuté en plus de détails à la section 2.2. En remplaçant l'identité de l'expéditeur par de l'aléa et en utilisant la clef commune ainsi dérivée dans un système de chiffrement symétrique, on arrive naturellement à un système de chiffrement fondé sur l'identité. (Il s'agit du même procédé que lors du passage de l'échange de clef à la Diffie–Hellman au système de chiffrement d'ElGamal ; c'est un mécanisme d'*encapsulation de clef* ou *KEM*, suivi d'une *encapsulation des données* ou *DEM*.) Ce système de chiffrement a été publié par Boneh et Franklin dans [19].

Ces travaux fondateurs ont ouvert une boîte de Pandore, et les cryptosystèmes fondés sur l'identité et s'appuyant sur les couplages sont devenus à la mode. Chaque primitive cryptographique aux propriétés exotiques a été retranscrite en sa version fondée sur l'identité ; souvent, la preuve de sécurité associée se fait par réduction à un nouveau

problème algorithmique de difficulté inconnue inventé pour l'occasion. Barreto écrit sur sa page web [11], dédiée aux cryptosystèmes fondés sur les couplages et mise à jour entre 2002 et 2005 : « The volume of research papers on pairing-based cryptography has increased exponentially over the past few years ». Dans la suite, je me contenterai de détailler mes propres contributions faites au début du millénaire.

2.2 Échange de clefs sans interaction

Cette section se fonde sur [52], dont les résultats ont été obtenus lors de l'encadrement du stage de master de Régis Dupont, et publiés en 2003 dans la version colloque [51].

Dans l'article [52], nous reprenons le protocole d'échange de clefs sans interaction de [158], dans une formulation légèrement généralisée, nous définissons un modèle de sécurité et donnons une preuve que le système est sûr dans ce modèle.

2.2.1 Le protocole ...

Soit

$$e : G_1 \times G_2 \to G_T$$

un couplage de deux groupes d'ordre ℓ premier, $G_1 = \langle P \rangle$ et $G_2 = \langle Q \rangle$, dans un troisième, G_T ; en pratique, il sera instancié comme à la section 2.1.3 par des couplages de points de ℓ-torsion d'une courbe elliptique dans un corps fini. La condition sur la primalité de ℓ n'est pas essentielle pour le protocole, mais elle assure que les problèmes de logarithme discret ont une chance d'être difficiles, voir le chapitre 4.

Nous supposons données deux fonctions de hachage prenant leurs valeurs dans les groupes, $H_1 : \{0,1\}^* \to G_1 \backslash \{0\}$ et $H_2 : \{0,1\}^* \to G_2 \backslash \{0\}$; de telles fonctions s'obtiennent aisément à partir de fonctions de hachage cryptographiques standard.

Le système étant fondé sur l'identité des participants, il y a un PKG qui choisit aléatoirement une clef maîtresse $s \in \{1, \ldots, \ell - 1\}$. La clef privée d'un acteur A d'identité ID_A (qui peut être son adresse mél, son numéro IP, ou toute chaîne de bits qui l'identifie uniquement) est donnée par le couple $(S_{A,1}, S_{A,2}) = (sH_1(\mathrm{ID}_A), sH_2(\mathrm{ID}_A))$; il la demande préalablement au PKG.

Maintenant, la clef secrète partagée entre deux acteurs A et B se calcule sans interaction entre les deux comme

$$e(H_1(\mathrm{ID}_A), H_2(\mathrm{ID}_B))^s = e(S_{A,1}, H_2(\mathrm{ID}_B)) = e(H_1(\mathrm{ID}_A), S_{B,2}),$$

chacun des acteurs utilisant sa propre clef privée et l'identité du partenaire pour arriver au même résultat. Un deuxième secret partagé,

$$e(H_1(\mathrm{ID}_B), H_2(\mathrm{ID}_B))^s,$$

s'obtient de la même façon. En posant $G_1 = G_2$ et en faisant les simplifications évidentes, on arrive au protocole de [158].

2.2.2 ... et sa preuve de sécurité

Sous une *preuve de sécurité* d'un système cryptographique on entend la démonstration de l'équivalence en temps polynomial entre le cassage du cryptosystème (dans un sens qu'il faut préciser) et la solution d'un problème relevant de la théorie algorithmique des nombres qui est supposé difficile. Notons bien qu'une preuve de sécurité ne démontre point la sécurité du système ; il n'y a aucun cryptosystème actuellement proposé pour lequel la difficulté du problème sous-jacent soit démontrée, ne serait-ce que sous des hypothèses raisonnables comme P≠NP.

Dans le cas du protocole d'échange de clefs sans interaction, le problème sous-jacent (c'est-à-dire pour lequel la réduction de sécurité peut se faire sans trop de peine) s'avère être le *problème bilinéaire de Diffie–Hellman (calculatoire)*, le *(C)BDH*, défini dans [19] pour le cas $G_1 = G_2$. Nous le définissons comme suit dans [52] :

> Étant donnés les éléments P, aP, cP de G_1 et Q, bQ, cQ de G_2,
> calculer $e(P,Q)^{abc}$.

On définit aisément la version décisionnel *DBDH* de ce problème :

> Étant donnés les éléments P, aP, cP de G_1 et Q, bQ, cQ de G_2
> et un élément $\zeta \in G_T$,
> décider si $e(P,Q)^{abc} = \zeta$.

Clairement, CBDH devient facile si on sait calculer des logarithmes discrets soit dans G_1, soit dans G_2, ce qui donne a ou b. Il suffirait aussi d'obtenir un logarithme discret dans G_T après avoir calculé deux valeurs du couplage, car $a = \log_{e(P,Q)} e(aP,Q)$. Finalement, résoudre le *problème de Diffie–Hellman calculatoire* ou *CDH* dans G_1 donnerait abP à partir de aP et bP, de sorte que la valeur cherchée du couplage se calculerait comme $e(abP,cQ)$, et de même pour le CDH dans G_2 et, par un argument similaire au précédent, dans G_T. Sur la réciproque, quasiment rien n'est connu ; cf. [81], où les auteurs font un lien entre les problèmes d'inverser un couplage, de résoudre CDH dans les groupes G_1, G_2 et G_T et des variantes de BDH.

Clairement aussi, une solution à CBDH donne une solution à DBDH, et rien n'est connu sur la réciproque.

Il convient ensuite de spécifier le modèle de sécurité, ou autrement dit, les capacités de l'adversaire (qu'on supposera aussi étendues que possible) et le but de son attaque (qu'on essaiera de sous-estimer). Dans [52], nous accordons à l'adversaire la capacité d'obtenir les clefs privées pour toutes les identités qu'il choisira (au cours de la preuve, c'est modélisé par des appels à des *oracles*). Son but est de deviner une clef partagée entre deux identités de son choix, mais dont il n'a pas au préalable demandé les clefs secrètes.

Comme d'habitude, il reste le problème de modéliser les fonctions de hachage. Nous nous sommes mis dans le modèle de l'*oracle aléatoire* ; c'est-à-dire, les fonctions de hachage sont supposées être des fonctions aléatoires entre les domaines spécifiés, et au cours de la preuve, elles sont modélisées encore une fois par des appels de l'attaquant à un oracle, qui lui répond par des valeurs aléatoires selon une distribution uniforme.

Le résultat principal de [52] est le suivant :

Théorème 2.1 *Supposons l'existence d'un attaquant probabiliste du cryptosystème qui en temps t et avec au plus q requêtes à l'oracle de clefs privées a une probabilité $\varepsilon > 0$ de renvoyer la bonne clef partagée entre deux identités de son choix. Alors, il y a un algorithme probabiliste qui résout le problème CBDH avec probabilité au moins*

$$\frac{\varepsilon}{2\,e^2\,(1+q)^2}$$

en temps

$$t' = O(t(t_1 + t_2 + \log q)) + t_3,$$

où t_1 est le temps de la multiplication par un scalaire dans G_1 ou G_2 ou d'une exponentiation dans G_T, t_2 est le temps pour créer un bit aléatoire, $\log q$ est le temps de recherche dans une liste triée avec au plus q entrées et t_3 est le temps utilisé pour inverser un élément de $\mathbb{F}_\ell^\times$.

Pour prouver le théorème, on construit un algorithme $\mathcal{B}$ pour résoudre CBDH qui crée une instance du cryptosystème étroitement liée à l'instance de CBDH et qui se sert de la réponse de l'attaquant $\mathcal{A}$. L'algorithme $\mathcal{B}$ contrôle en même temps les oracles pour les fonctions de hachage en renvoyant des valeurs aléatoires, mais dont il connaît le logarithme discret par rapport à l'un des éléments P ou aP ; cela peut être réalisé en renvoyant un multiple aléatoire soit de P, soit de aP. Pour pouvoir répondre à des requêtes de clefs privées, $\mathcal{B}$ doit connaître le logarithme à la base P ; pour transformer la réponse de $\mathcal{A}$ en une solution de CBDH, il doit connaître le logarithme à la base aP. Ainsi, parmi les au plus q requêtes, $\mathcal{B}$ choisit au préalable une à laquelle il garde le logarithme par rapport à aP. S'il a mal deviné, il doit déclarer forfait (et la probabilité de cet événement est assez grande, comme témoignée par la mauvaise probabilité de succès de $\mathcal{B}$ par rapport à q) ; sinon, il arrive à résoudre CBDH.

Ce modèle de l'oracle aléatoire, dans lequel toutes les preuves de sécurité se faisaient il y a encore quelques années, est assez contesté ; le contrôle accordé à $\mathcal{B}$ sur les fonctions de hachage laisse une impression de tricherie, même si formellement, la preuve est correcte. Évidemment, dès qu'on instancie les fonctions de hachage, toutes les preuves s'écroulent, voir aussi [28].

Concernant le protocole d'échange de clefs sans interaction, je ne pense pas qu'une preuve sans oracle aléatoire (on parlerait alors du *modèle standard*) soit facilement disponible, vu qu'il n'en est pas connu pour le système de chiffrement dérivé [19]. Notons que cela demanderait de modéliser soigneusement les propriétés des fonctions de hachage, et la propriété la plus forte demandée traditionnellement aux fonctions de hachage, la *résistance aux collisions*, ne suffit pas : supposons que H est une fonction de hachage $H : \{0,1\}^* \to \{1, \ldots, \ell - 1\}$ résistant aux collisions ; clairement,

$$H_1 : \{0,1\}^* \to G_1 = \langle P \rangle, \quad m \mapsto H(m) \cdot P$$

résiste encore aux collisions. Mais la façon de calculer H_1 expose les logarithmes discrets de ses valeurs, ce qui permettrait à tout participant au système de calculer toutes les clefs partagées. En effet, la clef partagée entre B et C s'obtiendrait par A comme

$$e(H_1(\mathrm{ID}_B), H_2(\mathrm{ID}_C))^s = e(P, H_2(\mathrm{ID}_C))^{sH(\mathrm{ID}_B)} = e(S_{A,1}, H_2(\mathrm{ID}_C)^{H(\mathrm{ID}_B)/H(\mathrm{ID}_A)}.$$

Le modèle de sécurité développé dans [52] peut être renforcé en affaiblissant le but de l'attaquant : si on demande qu'il soit incapable de distinguer la bonne clef partagée entre deux acteurs d'un élément aléatoire du groupe G_T, on peut prouver l'équivalent du théorème 2.1 en utilisant le problème décisionnel DBDH au lieu de CBDH.

2.3 Courbes elliptiques à petit degré de plongement

Je présente dans cette section les résultats de [50].

2.3.1 Le degré de plongement

Quand on essaie de mettre en œuvre un cryptosystème utilisant des couplages, il se pose le problème de trouver une courbe adéquate. Comme vu à la section 2.2.2, les seules attaques connues passent par un calcul de logarithme discret soit dans la courbe, soit dans le corps fini dans lequel le couplage prend ses valeurs. Pour que le logarithme discret dans la courbe elliptique définie sur $\mathbb{F}_q$ soit difficile, on doit imposer un sous-groupe d'ordre premier ℓ suffisamment grand (en général, 160 à 200 bits sont considérés sûrs, voir le chapitre 4). Le couplage prendra alors ses arguments en $E[\ell] \times E[\ell]$. Concernant le corps fini, il faut imposer une taille q^k suffisamment grande, où k est le plus petit entier tel que $E[\ell]$ soit défini sur $\mathbb{F}_{q^k}$ ou, de façon équivalente quand $E[\ell] \not\subseteq E(\mathbb{F}_q)$, tel que $\ell | q^k - 1$ (en général, q^k ayant entre 1024 et 2048 bits est considéré sûr). En même temps, on aimerait avoir ℓ et k aussi petit que possible pour des raisons d'efficacité. L'entier k est appelé le *degré de plongement* ou le *degré MOV* (en l'honneur de [136]) par rapport à la ℓ-torsion de la courbe E.

Contrairement à des cryptosystèmes elliptiques ordinaires, la solution de tirer une courbe au hasard jusqu'à tomber sur une courbe utilisable (cf. la section 3.5) ne peut être retenue, la densité de bonnes courbes étant trop faible : notons que k est l'ordre de q modulo ℓ, qui en général va être proche de $\ell - 1$ et donc au moins 2^{160}.

Une possibilité est de prendre des courbes supersingulières, qui ont des valeurs de k ne dépassant pas 6. Mais ces courbes étant très particulières, ont peut se demander si le problème du logarithme discret ne risque pas d'être plus facile pour ces courbes qu'en moyenne. En même temps, $k = 6$ peut être considéré comme trop petit, surtout car le rapport q^k/ℓ doit tendre vers l'infini pour contrer la loi de Moore.

Le seul moyen d'obtenir des courbes elliptiques ordinaires avec un petit degré de plongement est de passer par les constructions de la multiplication complexe, voir le chapitre 1.

En 2001, Miyaji, Nakabayashi et Takano ont été les premiers à analyser le degré de plongement pour des courbes ordinaires dans [140]. Ils donnent une caractérisation complète des courbes d'ordre premier sur un corps premier ayant un degré de plongement de 3, 4 ou 6.

Autour de 2002, il y a eu trois solutions indépendantes trouvant des courbes pour n'importe quel degré de plongement fixé ; [12, 50] et une approche attribuée à Cox et Pinch, mais non publiée par ses auteurs. Je me contente d'exposer nos résultats de [50].

Un survol récent sur toutes les techniques connues à ce jour est donné par Freeman, Scott et Teske dans [76].

2.3.2 Courbes elliptiques sur le bord de l'intervalle de Hasse

Supposons k fixé ; l'enjeu est de trouver des paramètres pour des courbes elliptiques avec multiplication complexe par un discriminant suffisamment petit, ayant un facteur premier ℓ du cardinal suffisamment grand et un degré de plongement égal à k.

Notre point de départ dans [50] est de considérer l'équation (1.7)

$$u^2 D = t^2 - 4q$$

et de forcer $u^2 D$ à être petit ; cela entraîne $|t| = \lfloor 2\sqrt{q} \rfloor$, et la courbe se trouve sur le bord de l'intervalle de Hasse. En effet, $2\sqrt{q} - |t| \geqslant 1$ résulterait tout de suite en $|u^2 D| \geqslant 4\sqrt{q} - 1$. Comme en plus $t^2 - 4q$ n'a aucune raison d'être divisible par un grand carré, on aurait alors un discriminant proche de $\sqrt{q} \geqslant 2^{80}$, inutilisable en vue des résultats du chapitre 1. Une approche alternative, poursuivie dans [12] et par Cocks et Pinch, serait de forcer un grand facteur carré dans $t^2 - 4q$. Soit

$$q = n^2 + a \text{ avec } n = \lfloor \sqrt{q} \rfloor \text{ et } 0 \leqslant a \leqslant n$$

ou

$$q = n^2 + n + a \text{ avec } n = \lfloor \sqrt{q} \rfloor \text{ et } 1 \leqslant a \leqslant n\,;$$

tout entier q s'écrit de façon unique dans exactement l'une de ces deux formes. Nous obtenons $|t| = \lfloor 2\sqrt{q} \rfloor = 2n$ dans le premier et $|t| = 2n + 1$ dans le deuxième cas.

Dans la suite, supposons que $q = n^2 + a$ et $t = 2n$ et donc $u^2 D = -4a$, les trois autres cas se traitant de manière analogique. Soit ℓ un grand facteur premier (et pour l'instant inconnu) du cardinal $q + 1 - t$. Le cardinal de la courbe est donné d'après (1.8) par $q + 1 - t = (n-1)^2 + a$, et on a $q \equiv t - 1 = 2n - 1 \bmod \ell$. Le degré de plongement k est égal à l'ordre de q modulo ℓ, ce qui se traduit par

$$\Phi_k(2n-1) \equiv 0 \bmod \ell \text{ et } k | \ell - 1$$

où Φ_k est le k-ème polynôme cyclotomique ; la deuxième condition assure que l'ordre de q n'est pas un diviseur propre de k.

Ainsi, nous obtenons le système d'équations

$$\begin{aligned} \Phi_k(2n-1) &\equiv 0 \bmod \ell \\ (n-1)^2 + a &\equiv 0 \bmod \ell, \end{aligned}$$

la condition $k | \ell - 1$ étant la plupart du temps satisfaite automatiquement, et se vérifiant à la fin.

Éliminant la variable n par un résultant, nous arrivons à un polynôme R_k, dont a est une racine modulo ℓ et dont on vérifie les propriétés suivantes :

Lemme 2.2 *R_k est un polynôme irréductible dans $\mathbb{Z}[X]$, de degré $\varphi(k)$ et de coefficient dominant $4^{\varphi(k)}$.*

Son coefficient constant est 1, à moins que k ne soit une puissance d'un premier p, dans lequel cas il est p^2.

Son contenu est 1, à moins que k ne soit une puissance de 2, dans lequel cas il est 4.

En particulier, on s'attend à ce que R_k représente une infinité de nombres qui sont soit premiers, soit quatre fois un premier. Maintenant, on fixe des valeurs de a et calcule les autres paramètres jusqu'à tomber sur une bonne combinaison.

Algorithme 2.3

Entrée: *$k \in \mathbb{N}$; un paramètre de sécurité $L \in \mathbb{N}$;*
un ensemble de discriminants $\mathcal{D}$ pour lesquels l'algorithme 1.15 peut être exécuté

Sortie: *des premiers p et $\ell \geqslant L$ et une courbe elliptique définie sur $\mathbb{F}_p$, de cardinal divisible par ℓ et de degré de plongement k*

pour $D \in \mathcal{D}$
  pour $u = u_{\min}, \ldots, u_{\max}$ *tel que* $4 | u^2 D$
    $a \leftarrow \frac{u^2 |D|}{4}$
    si $R_k(a)$ *a un facteur premier* $\ell \geqslant L$ *tel que* $k | \ell - 1$
      $n \leftarrow$ *une racine de* $\operatorname{pgcd}(\Phi_k(2Y-1), (Y-1)^2 + a) \bmod \ell$
      si $a \leqslant n$
        $p \leftarrow n^2 + a$
        si p *est premier*
          calculer la courbe E pour p et D par l'algorithme 1.15
          et renvoyer p, ℓ et E

Pour $u_{\min}$, il convient de choisir une valeur telle que $R_k(a) \geqslant L$; la valeur de $u_{\max}$ doit être suffisamment petite pour pouvoir factoriser $R_k(a)$ et avoir une chance raisonnable d'obtenir un facteur de l'ordre de L. L'entier n n'étant déterminé que modulo ℓ, on peut en tester plusieurs représentants.

Heuristiquement, comme on demande aux deux nombres p et ℓ de l'ordre de L d'être premier, le nombre de combinaisons à tester est de l'ordre de $O(\log^2 L)$.

L'algorithme possède l'avantage de pouvoir fonctionner avec des discriminants suffisamment larges pour se prémunir contre une attaque potentielle quand le nombre de classes est trop petit, voir p. 23. Comme les deux autres approches trouvées en même temps, il souffre du grand inconvénient qu'en général, $\log p \approx 2 \log \ell$. Ainsi, pour obtenir un niveau de sécurité de b bits, il faut travailler avec un sous-groupe d'une courbe elliptique d'un cardinal de $2b$ bits, ce qui rend l'arithmétique moins efficace. Pire encore, certaines applications telles que les signatures courtes de [20] ne peuvent pas être réalisées avec de telles courbes. Un résultat récent de Luca et Shparlinski donne une explication heuristique pourquoi il devrait être difficile d'éviter ce facteur de 2 [131]. Néanmoins, il y a des techniques pour obtenir ℓ plus proche de p ; la première a été décrite par Brezing et Weng dans [24]. Pour l'état de l'art, voir [76].

3 Équations modulaires

> Module, *(Architecture.) mesure prise à volonté pour régler les proportions des colonnes, & la symmétrie ou la distribution de l'édifice.*
> — Encyclopédie de Diderot et d'Alembert

Les équations modulaires sont étroitement liées aux polynômes de classes rencontrés au chapitre 1 sur la multiplication complexe. Si ces derniers paramètrent des courbes elliptiques ayant le même anneau d'endomorphismes, les polynômes modulaires, quant à eux, mettent en relation des courbes elliptiques différentes et leurs anneaux d'endomorphismes respectifs. Ils ont déjà servi à la section 1.2 pour obtenir des courbes elliptiques depuis des valeurs singulières de fonctions de classes. À la section 1.3, ils ont aidé à expliquer la hauteur des polynômes de classes pour des fonctions alternatives. Dans ce chapitre, nous introduisons plus formellement les équations modulaires, nous expliquons l'origine des théorèmes utilisés au chapitre 1, et nous donnons le meilleur algorithme connu à ce jour pour calculer toutes sortes de polynômes modulaires ; cet algorithme est une fois de plus quasi-linéaire en la taille de sa sortie.

3.1 Définitions et applications cryptographiques

3.1.1 Polynômes modulaires et isogénies entre courbes elliptiques

Soient Γ' un sous-groupe de congruences et f une fonction modulaire pour Γ' selon la définition 1.6. Considérons les classes à gauche $\Gamma'\backslash\Gamma$, posons $n = [\Gamma : \Gamma']$ et formons le polynôme

$$\Phi(X) = \prod_{M \in \Gamma'\backslash\Gamma} (X - f \circ M) = X^n + \sum_{\nu=1}^{n} c_\nu X^{n-\nu}. \tag{3.1}$$

Les coefficients de ce polynôme sont invariants sous toute transformation de Γ, et donc des fonctions modulaires de $\mathbb{C}_\Gamma = \mathbb{C}(j)$; et Φ est en fait le polynôme caractéristique (et génériquement le polynôme minimal) de f par rapport à l'extension algébrique $\mathbb{C}_{\Gamma'}/\mathbb{C}_\Gamma$.

Si f est holomorphe dans le demi-plan de Poincaré $\mathbb{H}$ (voir la définition 1.2), alors toutes ses conjuguées le sont, et les c_ν sont des polynômes en j. Par le principe de développement en q de Hasse, on a même $c_\nu \in \mathbb{Z}[j]$ quand les coefficients du développement en série de f sont des entiers rationnels, et que ceux de toutes les conjuguées sont des entiers algébriques, ce qui sera le cas pour toutes les fonctions qui nous intéressent.

Définition 3.1 Si f est comme ci-dessus, son polynôme caractéristique, vu comme polynôme bivarié

$$\Phi(X, Y) = \Phi_f(X, Y) \in \mathbb{Z}[X, Y]$$

tel que $\Phi(f, j) = 0$ est appelé le *polynôme modulaire* de f.

Par extension, si g est une deuxième fonction modulaire (éventuellement pour un autre sous-groupe de congruences) satisfaisant les mêmes propriétés d'intégralité et de rationnalité que f, un polynôme

$$\Psi(X, Y) = \Psi_{f,g}(X, Y)$$

tel que $\Psi(f, g) = 0$ est appelé polynôme modulaire entre f et g.

Deux fonctions f, modulaire pour Γ', et g, modulaire pour Γ'', sont également modulaires pour $\Gamma' \cap \Gamma''$, qui est encore un sous-groupe de congruences (pour le plus petit commun multiple des deux niveaux). Sans nuire à la généralité des arguments, on peut donc supposer que $\Gamma' = \Gamma''$. L'existence d'un polynôme $\Psi_{f,g}(X, Y)$ se déduit de l'existence des deux polynômes $\Phi_f(X, Z)$ et $\Phi_g(Y, Z)$: il suffit d'en prendre le résultant par rapport à Z. D'un point de vue pratique, il est préférable de prendre $\Psi_{f,g}$ comme le polynôme caractéristique de g par rapport à l'extension de corps $\mathbb{C}(f, g)/\mathbb{C}(f)$ ou *vice versa*; nous allons y revenir à la section 3.4.3.

Génériquement, f et g engendrent $\mathbb{C}_{\Gamma'}$, qui est un corps de fonctions de degré de transcendance 1 sur $\mathbb{C}$. Il peut ainsi être interprété comme le corps de fonctions de la courbe $\Gamma' \backslash \mathbb{H}^*$, appelée la *courbe modulaire* pour Γ'.

Certains sous-groupes s'avèrent d'un intérêt particulier. Il s'agit des $\Gamma^0(N)$ pour un entier N (voir la définition 1.5), et plus particulièrement pour $N = \ell$ premier, et de tous ses conjugués $R^{-1}\Gamma R \cap \Gamma$, ou R est une matrice primitive de déterminant N dans $\mathrm{Gl}_2(\mathbb{Z})$. La conjugaison peut ici être interprétée de deux façons; quand f est modulaire pour Γ^0, ses conjuguées $f \circ M$ sont modulaires pour les différents sous-groupes conjugués de $\Gamma^0(N)$. Le polynôme caractéristique de f est un modèle pour la courbe modulaire pour $\Gamma^0(N)$, notée par $X_0(N)$.

Le polynôme modulaire entre deux fonctions algébriquement indépendantes pour $\Gamma^0(N)$, qui en plus sont invariantes sous l'involution de Fricke–Atkin–Lehner $z \mapsto \frac{-N}{z}$ déjà rencontrée aux sections 1.2.3 et 1.6, est un modèle pour la courbe $X_0(N)$ quotientée par cette involution, notée par $X_0^+(N)$.

Définition et proposition 3.2 Une *isogénie* entre deux courbes elliptiques E et E' définies sur un corps K est une application rationnelle $\varphi : E \to E'$ qui en plus est un

homomorphisme de groupes. Elle définit un homomorphisme injectif φ^* des corps de fonctions par

$$\varphi^* : K(E') \to K(E), \quad f \mapsto f \circ \varphi.$$

Le *degré* de φ est le degré de l'extension de corps $K(E)/\varphi^*(K(E'))$. Si φ (c'est-à-dire $K(E)/\varphi^*(K(E'))$) est séparable, alors

$$\deg \varphi = |\ker \varphi|.$$

Une isogénie est ainsi la généralisation de la notion d'endomorphisme à des applications entre deux courbes différentes.

Les isogénies de degré ℓ premier sont d'un intérêt particulier ; elles ont un sous-groupe d'ordre ℓ comme noyau. En revenant à la théorie des courbes elliptiques sur $\mathbb{C}$ exposée à la section 1.1.4, on voit que cela correspond au passage du réseau $\Lambda = [1, z]$ à un sous-réseau Λ' d'indice ℓ. Les $\ell^2 - 1$ points d'ordre ℓ de la courbe de départ donnent $\ell + 1$ sous-groupes d'ordre ℓ, correspondant aux sous-réseaux $\left[1, \frac{z+\nu}{\ell}\right]$ pour $\nu = 0, \ldots, \ell - 1$ et $\left[\frac{1}{\ell}, z\right]$. Le polynôme qui a les j-invariants des quotients de base de ces sous-réseaux comme racines est donné par

$$\prod_{\nu=0}^{\ell-1} \left(X - j\left(\frac{z+\nu}{\ell}\right)\right) (X - j(\ell z)),$$

qui n'est rien d'autre que le polynôme modulaire Φ_f pour la fonction $f(z) = j(z/\ell)$, modulaire pour $\Gamma^0(\ell)$. Ainsi, $X_0(\ell)$, et plus généralement $X_0(N)$ pour N non nécessairement premier, paramètrent les couples de courbes elliptiques avec une isogénie de degré N et de noyau cyclique entre elles.

Le problème classique lié aux courbes modulaires $X_0(N)$ est de trouver de bons modèles, avec peu de singularités et de petits coefficients, voir, par exemple, [84]. Allant dans ce sens, Ogg a déterminé dans [150] lesquelles de ces courbes sont hyperelliptiques, et des équations concrètes pour ces cas ont été données dans [157, 147, 174].

Mais pour les applications, on a en général besoin des j-invariants des courbes elliptiques impliquées, de sorte qu'il est souvent préférable de s'en tenir aux polynômes modulaires de la définition 3.1 qui font intervenir la fonction j, quitte à garder des singularités et souffrir de coefficients plus grands.

3.1.2 Applications cryptographiques

Une application des polynômes pour $\Gamma^0(\ell)$ avec ℓ premier est directement liée à la multiplication complexe ; il s'agit de l'algorithme de Kohel pour déterminer l'anneau des endomorphismes d'une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ [123, 75]. En combinant (1.7) et (1.8), le cardinal nous fournit un discriminant fondamental Δ tel que $u^2\Delta = (q + 1 - |E(\mathbb{F}_q)|)^2 - 4q$. Alors, l'anneau des endomorphismes de la courbe est $\mathcal{O}_D$ avec $D = f^2\Delta$ pour un $f|u$.

L'idée de base de l'algorithme est d'utiliser le lien entre le nombre de ℓ-isogénies rationnelles et le discriminant D' d'une courbe pour en déduire f. En effet, pour $D' = \Delta$,

il y a $\left(\frac{D'}{\ell}\right)+1$ isogénies de degré ℓ préservant l'anneau des endomorphismes (elles correspondent aux multiplications par les idéaux de $\mathcal{O}_\Delta$ de norme ℓ), et $\ell-\left(\frac{D'}{\ell}\right)$ qui ajoutent un facteur ℓ au conducteur. Si $D'=(f')^2\Delta$ avec $\ell|f'$, il y a une ℓ-isogénie qui enlève un facteur ℓ au conducteur, et ℓ qui en ajoutent un. Dès que le conducteur u est atteint, les isogénies ajoutant un facteur ℓ ne sont plus $\mathbb{F}_q$-rationnelles. Le nombre d'isogénies rationnelles depuis une courbe au j-invariant $\bar{\jmath}$ s'obtient en comptant le nombre de racines dans $\mathbb{F}_q$ d'un polynôme modulaire pour $\Gamma^0(\ell)$ instancié en $\bar{\jmath}$; si le polynôme $\Phi_{j(z/\ell)}$ est choisi, ces racines correspondent directement aux j-invariants des courbes ℓ-isogènes.

Une autre application, dont il sera encore question à la section 3.5, est l'accélération de l'algorithme de Schoof pour compter le nombre de points sur une courbe elliptique définie sur un corps fini [166] due à Atkin et Elkies [53, 141, 167]. Premièrement, le nombre de racines d'un polynôme modulaire pour $\Gamma^0(\ell)$, instancié en le j-invariant de la courbe, donne la valeur de $\left(\frac{D}{\ell}\right)$ comme dans le paragraphe précédent. Si le symbole de Legendre vaut 1 (cas des premiers dits d'Elkies), le calcul explicite des isogénies fournit les valeurs propres du Frobenius sur la ℓ-torsion, ce qui détermine son polynôme caractéristique (1.6) modulo ℓ ; les restes chinois permettent alors de conclure.

Finalement, comme déjà discuté à la page 18, le polynôme modulaire entre j et une fonction de classes permet de retrouver une courbe elliptique à multiplication complexe depuis une racine d'un polynôme de classes alternatif.

3.2 Équations modulaires pour un niveau premier

Les résultats originaux de cette section, travaux communs avec François Morain, ont été publiés dans [64].

Il y a essentiellement trois choix de polynômes modulaires pour $\Gamma^0(\ell)$ avec ℓ premier qui ont été proposés dans la littérature et qui permettent d'expliciter assez aisément les isogénies entre courbes elliptiques :

- les polynômes *traditionnels* ou *classiques* à partir de $j(z/\ell)$; l'isogénie se calcule à partir du développement en série de la fonction $\wp$ de Weierstraß, voir [167, sections 7 et 8] ;
- les polynômes *canoniques* à partir des quotients simples w_ℓ^{2s} de η tels que donnés à la définition 1.18 ; la clef pour expliciter l'isogénie est de noter que la série d'Eisenstein G_2 de (1.1) apparaît dans la dérivée logarithmique de la fonction, voir [141, section 3.2.1] ;
- des polynômes entre j et une fonction f sur $X_0^+(\ell)$; le lien avec les isogénies est maintenant fait par l'involution de Fricke–Atkin–Lehner qui laisse f invariante et qui transforme $j(z)$ en $j(z/\ell)$, voir [141, section 3.2.2].
  Morain décrit dans [141, Section 2.3.1] une procédure attribuée à Atkin, appelée *blanchiment*, qui est censée donner heuristiquement une fonction sur $X_0^+(\ell)$ de pôle minimal à l'infini. En vue de l'algorithme asymptotiquement optimal du chapitre 3.4, les fonctions proposées par Müller dans [145, section 5.3] sont préférables en pratique ; bien que non optimales par rapport à l'ordre du pôle à l'infini, elles s'évaluent plus rapidement. Il s'agit du quotient d'une forme modulaire de

poids 1, obtenue comme produit de deux fonctions η, et de la même forme transformée par un opérateur de Hecke de niveau r premier satisfaisant $\left(\frac{r}{\ell}\right) = \left(\frac{\ell}{r}\right) = 1$ et $24|(r-1)(\ell+1)$:

$$f_{\ell,r} = \frac{T_r(\eta(z)\eta(\ell z))}{\eta(z)\eta(\ell z)} \tag{3.2}$$

avec

$$T_r(f) = \frac{1}{r}\sum_{\nu=0}^{r-1} f\left(\frac{z+24\nu}{r}\right) + f(rz).$$

Cette fonction est modulaire pour $\Gamma_0(\ell)$; en l'évaluant en $-1/z$ on obtient une fonction pour $\Gamma^0(\ell)$.

Dans [64], nous énonçons le résultat suivant pour les polynômes canoniques :

Théorème 3.3 *Le degré de $\Phi_{w_\ell^{2s}}(X,Y)$ en Y (la variable correspondant à j, voir la définition 3.1) est de $\frac{s(\ell-1)}{12}$. Le coefficient dominant par rapport à Y est $-X$, et le coefficient constant par rapport à X est ℓ^s.*

La démonstration s'obtient aisément en notant qu'un système de représentants de $\Gamma^0(\ell)\backslash\Gamma$ pour ℓ premier est donné par les *translations* $T^b = \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix}$ pour $b = 0, \ldots, \ell-1$ et la *réflexion* $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ (cf. la démonstration de la proposition 1.3), et en remplaçant les conjuguées de la fonction dans (3.1) par leurs développements en série. Pour une fonction modulaire f avec développement $f = \sum_{\nu_0}^{\infty} c_\nu q^{\nu/\ell}$, le développement de $f \circ T^b$ s'écrit directement comme $f \circ T^b = \sum_{\nu_0}^{\infty} c_\nu \zeta_\ell^{b\nu} q^{\nu/\ell}$ avec $\zeta_\ell = e^{2\pi i/\ell}$. Pour la réflexion, on ne peut procéder symboliquement et doit tenir compte du comportement de la fonction particulière ; ici, $w_\ell^{2s} \circ S(z) = \ell^s \frac{\eta(\ell z)}{\eta(z)}$, dont le développement est facile à obtenir.

3.3 Équations modulaires pour un niveau produit de deux premiers

Cette section porte sur les résultats de [69], travaux en commun avec Reinhard Schertz.

Pour appliquer les résultats sur la hauteur des polynômes de classes, notamment le théorème 1.23, aux quotients doubles de η de la section 1.2.3, il faut connaître le polynôme modulaire qui relie cette fonction à j. Si on veut le calculer par les méthodes classiques, il faut en plus avoir à sa disposition les développements en série de Fourier de toutes ses conjuguées. C'est le sujet de [69], qui donne notamment l'équivalent du théorème 3.3.

Dans le cas de fonctions pour $\Gamma^0(\ell)$ avec ℓ premier, toutes les conjuguées sauf une sont obtenues à travers des translations, ce qui permet de facilement dériver leurs développements en q. Quand le niveau N est composé, il s'y ajoute essentiellement un ensemble de conjuguées pour chaque diviseur de N. Dans un premier temps, nous donnons dans

[69] un système de représentants de $\Gamma^0(N)\backslash\Gamma$ quand N est le produit de deux nombres premiers, non nécessairement distincts.

Pour traiter ces nouveaux conjuguées, il faut connaître le comportement des fonctions examinées sous transformations unimodulaires. À cet effet, nous dérivons de la transformation de la fonction η le résultat suivant pour les briques de bases des w_{p_1,p_2}.

Proposition 3.4 *Soit* $K \in \mathbb{N}$ *et* $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma$ *normalisée telle que* $c \geqslant 0$, *et* $d > 0$ *si* $c = 0$. *Écrivons* $c = \gamma \cdot 2^\lambda$ *avec* γ *impair et, par convention,* $\gamma = \lambda = 1$ *si* $c = 0$. *Étant donnés la relation de Bézout*

$$\delta = \operatorname{pgcd}(a, K) = \operatorname{pgcd}(a, Kc) = ua + vKc$$

et $U = \begin{pmatrix} a/\delta & -v \\ Kc/\delta & u \end{pmatrix}$, *on a*

$$\eta\left(\frac{Mz}{K}\right) = \varepsilon(U)\sqrt{\delta(cz+d)}\,\eta\left(\frac{\delta z + (uB + vKd)}{K/\delta}\right)$$

avec

$$\varepsilon(U) = \left(\frac{a}{\gamma}\right)\zeta_{24}^{ab+c(d(1-a^2)-a)+3\gamma(a-1)+\frac{3}{2}\lambda(a^2-1)}$$

pour $\zeta_{24} = e^{2\pi i/24}$.

On en déduit facilement que les w_{p_1,p_2} sont invariantes sous l'involution de Fricke–Atkin–Lehner. Nous en dérivons également les différentes conjuguées en tant que quotients de fonctions $\eta\left(\frac{az+b}{d}\right)$, ce qui permet d'obtenir leurs développements en série. En particulier, il en découle l'ordre du pôle à l'infini ainsi que son résidu, ce qui nous met en mesure de démontrer le résultat suivant sur le polynôme modulaire :

Théorème 3.5 *Toutes les conjuguées du double quotient* $w^s_{p_1,p_2}$ *de la définition 1.21 sont des séries en* $q^{1/N}$ *avec coefficients dans* $\mathbb{Z}(\zeta_N)$ *pour* $\zeta_N = e^{2\pi i/N}$, *de sorte que* $\Phi_{w^s_{p_1,p_2}} \in \mathbb{Z}[X,Y]$ *par le principe des développements en q de Hasse. En tant que polynôme en* Y, *le degré de* Φ *est* $\frac{s(p_1-1)(p_2-1)}{12}$ *et son coefficient dominant est* $X^{p_1+p_2}$ *pour* $p_1 \neq p_2$ *et* X^{p-1} *pour* $p_1 = p_2 = p$. *En tant que polynôme en* X, *son coefficient constant est* 1 *pour* $p_1 \neq p_2$ *et* $p^{s(p-1)/2}$ *pour* $p_1 = p_2 = p$.

Rappelons pour compléter la présentation que le degré en X de tout polynôme modulaire entre une fonction pour $\Gamma^0(N)$ et j est donné par

$$\psi(N) = N\prod_{p|N}\left(1+\frac{1}{p}\right) = \prod_{p^e||N}(p^{e-1}+p^e), \tag{3.3}$$

où les produits sont pris sur tous les premiers respectivement puissances de premiers divisant N.

3.4 Algorithme quasi-linéaire pour les équations modulaires

Je présente dans ce chapitre les résultats de [59].

Comme pour les polynômes de classes (voir le chapitre 1.5), on peut se demander s'il est possible de calculer également les polynômes modulaires en temps quasi-linéaire en leur taille. Pour simplifier l'exposition, contentons-nous pour le moment du cas de $\Gamma^0(\ell)$ avec ℓ premier, qui est le plus intéressant pour les applications de la section 3.1.2 ; d'autres polynômes sont évoqués vers la fin de la section 3.4.2 et à la section 3.4.3.

Dans un premier lieu, il faut avoir une idée précise de la taille de la sortie. Une excellente estimation dans le cas des polynômes traditionnels entre $j(z)$ et $j(z/\ell)$ est donnée par Cohen dans [34], qui montre que la hauteur logarithmique est bornée par

$$6(\ell+1)(\log \ell + O(1)) \subseteq O(\ell \log \ell). \tag{3.4}$$

Le polynôme ayant un degré $\ell+1$ dans les deux variables, on arrive ainsi à une taille de sortie de $O(\ell^3 \log \ell)$.

Pour d'autres fonctions, on observe une croissance semblable des coefficients, et il devrait être possible de démontrer un résultat analogue avec une constante dépendant de la classe de fonctions à la place de 6.

Dans [59], je fais un survol et entreprends une analyse de complexité des différents algorithmes pour calculer des polynômes modulaires présentés dans la littérature. À titre de comparaison, donnons rapidement l'approche classique, décrite par exemple dans [141]. Elle consiste à déterminer explicitement les développements en $q^{1/\ell}$ des différentes conjuguées et de calculer les coefficients c_ν du polynôme selon (3.1), encore sous forme d'un développement en q. Cela définit un système linéaire triangulaire qui permet de facilement reconnaître c_ν comme un polynôme en j : si la série commence par $d_k q^{-k}$, le terme dominant du polynôme est $d_k j^k$, car le développement de j commence par q^{-1} ; ainsi, on dépile les puissances décroissantes de j. Le degré des polynômes en j étant généralement en $O(\ell)$ pour les fonctions qui nous intéressent (cf. les théorèmes 3.3 et 3.5), il faut connaître les développements des conjuguées jusqu'à un exposant dans $O(\ell)$, c'est-à-dire qu'il faut $O(\ell^2)$ coefficients par série. Ces coefficients sont *a priori* dans $\mathbb{Z}[\zeta_\ell]$, et il faut les calculer avec une précision de $\tilde{O}(\ell)$ chiffres, où la tilde indique que des facteurs $\log \ell$ ont été omis. Ainsi, représenter une série demande $\tilde{O}(\ell^4)$ bits. Comme il y a $O(\ell)$ conjuguées à manipuler, l'espace mémoire utilisé est déjà de $\tilde{O}(\ell^5)$. En employant des techniques de multiplication rapide, on arrive ainsi à une complexité de $\tilde{O}(\ell^5)$. Morain observe dans [141] que les racines de l'unité dans les séries proviennent des translations, et en considérant séparément les conjuguées obtenues par translation et celle obtenu par réflexion, on peut se contenter de calculs dans $\mathbb{Z}$, ce qui fait arriver à une complexité totale de $\tilde{O}(\ell^4)$. Par rapport à la taille de la sortie, c'est un facteur de ℓ en trop.

3.4.1 L'algorithme ...

L'idée de base de l'algorithme quasi-linéaire se trouve déjà dans le livre [21, chapitre 4.5] des Borweins. Elle consiste à noter que l'équation (3.1) entre fonctions modu-

laires reste valable si on instancie les arguments des fonctions par des nombres complexes. En *évaluant* explicitement les conjuguées de la fonction f de départ en un $z_k \in \mathbb{H}$ et en multipliant les facteurs linéaires $X - (f \circ M)(z_k)$, on obtient le polynôme

$$\Phi(X)(z_k) = X^n + \sum_{\nu=1}^{n} c_\nu(z_k) X^{n-\nu} \in \mathbb{C}[X]. \tag{3.5}$$

Maintenant, les c_ν en tant que fonctions modulaires sont en fait des polynômes en j ; il faut déterminer les $c_{\nu,\mu}$ tels que $c_\nu(z_k) = \sum_\mu c_{\nu,\mu}\, j(z_k)^\mu$, ce qui est simplement un problème d'*interpolation* d'un polynôme univarié dont on connaît les valeurs en les arguments $j(z_k) \in \mathbb{C}$. Si la précision des calculs est suffisamment grande, on arrive ensuite à arrondir les $c_{\nu,\mu}$ vers leurs valeurs correctes dans $\mathbb{Z}$.

3.4.2 ... et sa complexité

Examinons séparément les deux phases de l'algorithme, l'évaluation et l'interpolation. Pendant la première phase, il faut obtenir $O(\ell^2)$ valeurs de fonctions modulaires (ou plus généralement, $O(\deg_X \Phi \cdot \deg_Y \Phi)$ valeurs) à une précision flottante de n bits avec $n \in O(\ell \log \ell)$ d'après (3.4). En évaluant naïvement les séries en $q^{1/\ell}$ jusqu'à un exposant de $O(n)$, il faut pour cela considérer $O(\ell n)$ termes. On arrive à une complexité de $O(\ell^3 n\, M(n)) = O\tilde{}(\ell^5)$, ce qui est pire que pour les calculs formels avec les séries. Cela peut expliquer en partie pourquoi l'approche des Borweins ne semble pas s'être répandue.

Mais comme à la section 1.5.2, on peut noter que toutes les fonctions qui nous intéressent peuvent être construites à partir de η. Et en remplaçant l'évaluation d'une fonction quelconque pour $\Gamma^0(\ell)$ par un nombre constant d'évaluations de η, on gagne sur deux fronts : premièrement, au lieu d'une série en $q^{1/\ell}$, on arrive à une série en q (mis à part le facteur $q^{1/24}$, qui ne gêne pas), ce qui fait gagner un facteur ℓ au nombre de termes à considérer ; deuxièmement, la série creuse s'évalue plus vite, ce qui fait gagner un facteur $O(\sqrt{n})$ comme à la section 1.5.2. Ainsi, la complexité de la phase d'évaluation devient $O(\ell^2 \sqrt{n}\, M(n)) = O\tilde{}(\ell^{3,5})$, et l'approche classique est battue.

Il n'y a aucune raison de s'arrêter là ; en employant l'une ou l'autre des techniques quasi-linéaires d'évaluation de fonctions modulaires exposées à la section 1.5.2, la complexité devient $O\tilde{}(\ell^2 M(n)) = O\tilde{}(\ell^3)$.

Il reste à considérer la reconstruction du polynôme $\Phi(X)(z_k)$ à partir de ses racines, ou autrement dit, la multiplication de ses facteurs linéaires. En organisant encore une fois les calculs dans un arbre aussi équilibré que possible comme déjà vu à la section 1.4.1, cette étape a également une complexité de $O\tilde{}(\ell^3)$; en regardant de plus près les facteurs $\log \ell$, elle devient même dominante.

De même, l'interpolation se fait par les algorithmes rapides déjà évoqués à la section 1.4.1 en temps $O\tilde{}(\ell^3)$.

Les arguments précédents ne se limitent pas au cas de $\Gamma^0(\ell)$, mais permettent en fait de traiter n'importe quel sous-groupe de congruences. En analysant plus précisément les facteurs logarithmiques apparaissant dans les complexités, j'arrive au résultat suivant dans [59] :

Théorème 3.6 *Soit $\Gamma' \subseteq \Gamma$ un sous-groupe de congruences et f une fonction modulaire pour Γ' telle que le polynôme modulaire Φ_f vit dans $\mathbb{Z}[X,Y]$. Soit n la hauteur logarithmique de Φ. Supposons qu'un système de représentants de $\Gamma'\backslash\Gamma$ soit connu et que f peut être évaluée à précision $O(n)$ en temps $O(\log^2 n\, M(n))$. Alors l'algorithme procédant par évaluation et interpolation calcule Φ en temps*

$$O\left(\deg_X \Phi \, \deg_Y \Phi \, (\log^2 \max(\deg_X \Phi, \deg_Y \Phi) + \log n)\, M(n)\right) \ ;$$

dans le cas du polynôme traditionnel pour $\Gamma^0(\ell)$ entre $j(z)$ et $j(z/\ell)$, cette complexité devient

$$O(\ell^3 \log^4 \ell \log\log \ell).$$

Comme le théorème 1.27, ce résultat a un caractère heuristique dans le sens que rien ne garantit la correction du polynôme modulaire en la présence d'erreurs d'arrondi. Néanmoins, on peut par exemple vérifier la factorisation du polynôme instancié en le j-invariant d'une courbe elliptique à cardinal connu sur un corps fini et ainsi s'assurer de façon probabiliste de sa correction.

Outre la bonne complexité de l'algorithme, son applicabilité directe à des sous-groupes de congruences quelconques en est un grand atout. Elle dispense complètement du calcul parfois difficile des développements en q des conjuguées.

3.4.3 Généralisations

L'algorithme opérant par évaluation et interpolation se prête également à calculer ce qui pourrait être appelé des « polynômes modulaires relatifs », un cas particulier des polynômes Ψ de la définition 3.1. Supposons donnés deux sous-groupes de congruences $\Gamma'' \subseteq \Gamma'$ et deux fonctions modulaires f pour Γ'' et g pour Γ'. Considérons le polynôme $\Psi(X)$ défini de façon analogue à (3.1) par

$$\Psi(X) = \prod_{M \in \Gamma''\backslash\Gamma'} (X - f \circ M) = X^n + \sum_{\nu=1}^{n} c_\nu X^{n-\nu}.$$

Les coefficients c_ν sont des fonctions modulaires invariantes sous Γ'. On ne peut plus en toute généralité en déduire qu'ils sont des polynômes en g. Par contre, dans le cas très particulier que $\mathbb{C}_{\Gamma'} = \mathbb{C}(g)$, les c_ν sont des fonctions rationnelles en g à coefficients dans $\mathbb{C}$. En examinant les pôles à l'infini et la nature des coefficients des développements en série de Fourier on peut alors espérer arriver encore à des polynômes en g à coefficients dans $\mathbb{Z}$; ainsi, le polynôme Ψ défini ci-dessus de façon analytique correspond bien au polynôme $\Psi_{f,g}$ de la définition 3.1. Pour cela, il faut au moins que le corps de fonctions $\mathbb{C}_{\Gamma'}$ soit de genre 0, ce qui n'arrive que pour un nombre fini de sous-groupes de congruences.

L'algorithme quasi-linéaire s'applique presque sans modifications pour calculer ce type de polynômes modulaires ; il suffit d'énumérer $\Gamma''\backslash\Gamma'$ dans la phase d'évaluation et de faire l'interpolation par rapport aux valeurs de g au lieu de j.

De tels polynômes ont été introduits en 1870 par Schläfli dans [165] et examinés plus systématiquement par Weber dans [180, §§73–74]. La fonction g de base est prise comme

l'une des fonctions de Weber notée par f, f_1 et f_2 dans (1.5), qui sont modulaires pour des sous-groupes de $\Gamma' = \Gamma(48)$. Avec un degré de transformation ℓ premier et différent de 2 et 3, la deuxième fonction f est prise comme $g(z/\ell)$. On vérifie aisément que f est modulaire pour $\Gamma'' = \Gamma^0(\ell) \cap \Gamma(48)$, et un système de représentants de $\Gamma''\backslash\Gamma'$ est obtenu simplement en modifiant le système de représentants standard de $\Gamma^0(\ell)\backslash\Gamma$ de sorte que les matrices sont dans $\Gamma(48)$. Les polynômes ainsi obtenus ont des coefficients nettement plus petits que ceux du polynôme traditionnel pour $\Gamma^0(\ell)$; Weber remarque dans [180, page 245] que ce dernier n'était à l'époque calculable que pour $\ell = 2$, tandis qu'il écrit aisément l'équation de Schläfli pour $\ell = 19$ en quelques lignes à la page 263. En plus, seulement un coefficient sur 24 apparaît dans les polynômes, comme démontré dans [180, page 266]. Notre algorithme en profite directement car le nombre d'évaluations ainsi que, après récriture, le degré des polynômes univariés à interpoler sont divisés par 24.

Au chapitre 5 de sa thèse [106], dont j'étais rapporteur, Hart généralise les équations de Schläfli à d'autres quotients simples de η de petit niveau tels que donnés à la définition 1.18. Il suffit de prendre $g = w_p^{2s}$ et $f(z) = g(z/\ell)$ pour un ℓ premier différent de p. Expérimentalement, j'ai constaté qu'on obtient de telles équations également pour $w_{5,7}$ et $w_{3,13}$. Ces équations peuvent servir à généraliser les algorithmes p-adiques de la multiplication complexe (voir la section 1.5.1) à des fonctions de classes alternatives, comme remarqué par Bröker au chapitre 6.8 de sa thèse [26], dont j'étais également rapporteur.

3.5 Application au comptage de points sur une courbe elliptique

Dans cette section, je donne quelques détails de mon implantation de l'algorithme quasi-linéaire, tirés de [59], et je présente les records pour le calcul du cardinal d'une courbe elliptique sur un corps premier, obtenus avec Pierrick Gaudry et François Morain [63, 62, 66].

Le logiciel `modpol` est le compagnon de `cm` présenté à la section 1.7.3, étant écrit en C et fondé sur `mpc`, `mpfr` et en dernier lieu `gmp`, et partageant des modules tels que l'évaluation de fonctions modulaires. Il n'est pas mis à la disposition du public.

L'évaluation asymptotiquement rapide de fonctions modulaires n'étant rentable que pour de très grandes précisions (voir le tableau 1.2), les fonctions sont évaluées via la série creuse pour η. Le temps pour l'interpolation s'avérant négligeable par rapport à l'évaluation, elle est implantée par un simple algorithme quadratique via les différences itérées ; elle est aidée par le choix des arguments z_k pour arriver, par exemple, à une progression arithmétique entière pour les abscisses $j(z_k)$ des points d'interpolation. Contrairement au cas des polynômes de classes, où il suffit de prendre une précision flottante correspondant directement aux approximations de (1.12) et du théorème 1.23 pour arriver au résultat correct, il paraît que l'interpolation numérique introduit des erreurs d'arrondi non négligeables – un effet classique lié au mauvais conditionnement des matrices de Vandermonde. Ainsi, il faut augmenter la précision des calculs par un petit facteur, déterminé expérimentalement entre 1,1 et 2 selon la fonction et le niveau.

L'évaluation se distribue facilement sur plusieurs machines, chaque machine calculant

une « ligne » $\Phi(X)(z_k)$ de (3.5), ce qui permet une accélération par un facteur égal au nombre de processeurs utilisés ; la communication est négligeable. L'interpolation pourrait également être distribuée, chaque processeur s'occupant d'un c_ν correspondant à une « colonne » de (3.5), mais le besoin ne s'est pas fait ressentir.

Pour les utiliser dans l'algorithme de Schoof–Elkies–Atkin (SEA) de calcul du nombre de points sur une courbe elliptique, j'ai précalculé les polynômes modulaires pour $\Gamma^0(\ell)$ utilisant les fonctions $f_{\ell,r}$ de (3.2) sans lacunes jusqu'à un niveau ℓ d'environ 6000 et quelques niveaux au-delà. Ces polynômes, stockés sous forme de fichiers texte compressés, prennent environ 800 Go.

Voici les détails du record, obtenu pour $\ell = 10\,079$, les temps de calcul se référant à l'équivalent d'un seul Opteron 64 cadencé à 2,4 GHz.

ℓ	10 079
r	5
$\deg_Y \Phi$	673
précision des calculs en bits	35 051
hauteur en bits	28 825
temps d'évaluation	10 000 000 s $\approx$ 120 d
temps d'interpolation	56 000 s $\approx$ 16 h
taille en tant que fichier texte compressé	16 Go

On voit que, comme il se doit pour un algorithme quasi-linéaire, c'est encore une fois la taille du résultat qui limite les calculs, d'autant plus que l'implantation est distribuée : les 120 jours de calcul correspondent en fait à une petite semaine sur une vingtaine de machines.

Entre autres grâce à ces calculs, nous avons battu des records pour SEA, obtenant en succession rapide le cardinal de courbes elliptiques sur un corps premier dont la caractéristique avait 1500 [63], 2000 [62] et 2500 [66] chiffres décimaux. Les autres ingrédients des records étaient des améliorations à l'intérieur de SEA dues à Bostan, Gaudry, Mihăilescu, Morain, Salvy et Schost et décrites dans [22, 88, 139].

Quand on veut répéter SEA sur plusieurs courbes, on peut considérer l'obtention des équations modulaires comme un précalcul. Concernant les records, par contre, les polynômes modulaires restent le goulot d'étranglement malgré la complexité théorique quasi-linéaire. Opposons pour le voir le temps de SEA (sans polynômes modulaires) de 195 jours pour la courbe elliptique à 2500 chiffres au temps de 120 jours pour la seule équation modulaire de niveau 10 079.

3.6 Perspectives

Un certain nombre de commentaires faits au chapitre 1.8 sur les polynômes de classes est également valable pour les polynômes modulaires.

Nous disposons d'un algorithme quasi-linéaire pour les calculer, et on pourrait penser que la recherche peut s'arrêter là-dessus. Néanmoins, des améliorations dans les détails sont envisageables. Les fonctions (3.2) pour $\Gamma^0(\ell)$ ne sont optimales que pour les petits

niveaux ℓ, et il reste à étudier si les fonctions obtenus par la méthode du blanchiment d'Atkin évoquée au chapitre 3.2 sont utilisables dans le cadre de l'algorithme asymptotiquement rapide. Il ne suffit pas de les avoir uniquement sous forme de leurs séries en q, car celles-ci ne convergent qu'extrêmement lentement autour des pointes rationnelles. Une solution pourrait être de les exprimer en tant que quotients de fonctions ϑ et η. Une approche intermédiaire est de combiner linéairement des fonctions $f_{\ell,r}$ de (3.2) pour plusieurs valeurs de r afin de réduire l'ordre du pôle à l'infini ; le temps d'évaluation d'une fonction $f_{\ell,r}$ étant proportionnel à r, cela réduirait la taille du résultat au détriment d'un temps de calcul agrandi.

Une autre solution pourrait consister à utiliser les polynômes de Schläfli généralisés de la section 3.4.3 entre une fonction $g(z)$ et sa transformée $g(z/\ell)$ pour le comptage de points. Cela devrait être possible quand g est une fonction de classes pour le discriminant de l'ordre quadratique associé à la courbe. Les polynômes à la Schläfli se calculant bien mieux que ceux pour $\Gamma^0(\ell)$, cette approche permettrait d'aller plus loin, mais à chaque fois sur une famille restreinte de courbes.

Comme pour les polynômes de classes, il se pose la question de certifier les résultats obtenus avec des calculs flottants sujets aux erreurs d'arrondi. Encore une fois, c'est un problème de principe plutôt que pratique : en spécialisant les polynômes dans des j-invariants de courbes elliptiques à multiplication complexe connue et en factorisant le résultat, on peut se convaincre aisément de la correction. Mais comme les tests de primalité, ces tests ne marchent que dans un sens : ils ne peuvent que rejeter des polynômes faux, mais laissent planer le doute sur des polynômes justes. Il serait souhaitable d'avoir une vraie preuve de correction, de préférence avec un certificat, qui s'obtienne ou au moins se vérifie plus vite que le calcul du polynôme lui-même.

Une autre possibilité serait de partir directement de calculs certifiés en utilisant l'arithmétique des intervalles. Mais celle-ci devrait être implantée sur les complexes au lieu des réels comme d'habitude, et de préférence être compatible avec l'arithmétique rapide des polynômes par la FFT, ce qui est loin d'être évident.

Finalement, il y a le genre supérieur, pour lequel presque tout reste à faire. Pour l'utilisation dans un algorithme qui détermine la fonction zêta d'une courbe de genre 2, Gaudry et Schost ont employé une méthode algébrique pour calculer des polynômes modulaires déjà instanciés en les invariants de la courbe ; ils atteignent un niveau $\ell = 19$ dans [93]. En faisant des calculs symboliques astucieux, ils arrivent à l'équation modulaire non instanciée pour le niveau $\ell = 3$ dans [89]. Au chapitre 10.4.2 de sa thèse [48], Dupont traite le calcul de polynômes modulaires pour le genre 2. Ces polynômes multivariés sont dérivés des invariants d'Igusa de la courbe et, contrairement aux polynômes de Gaudry et Schost, ont des dénominateurs. L'algorithme employé est inspiré de celui du chapitre 3.4 et procède par évaluation et interpolation. Dupont a réussi à calculer le polynôme pour $\ell = 2$, qui prend 27 Mo sous forme compressée, et le dénominateur pour $\ell = 3$. Comme dans le cas des courbes elliptiques, il paraît que les invariants les plus naturels de la courbe fournissent des polynômes impraticablement grands ; ainsi, la recherche de fonctions de classes évoquée au chapitre 1.8 rejoint la quête pour de meilleures équations modulaires.

4 Logarithmes discrets dans les jacobiennes

Logarithme, *s. m. (Arithmét.) nombre d'une progression arithmétique, lequel répond à un autre nombre dans une progression géométrique.*

— Encyclopédie de Diderot et d'Alembert

Le problème du logarithme discret dans un groupe fini est l'un des problèmes supposés difficiles qui rendent la cryptographie *asymétrique* ou *à clef publique* possible. Dans ce chapitre, je donne un survol des algorithmes connus à ce jour pour les jacobiennes de courbes de genre 2 et supérieur définies sur un corps fini ; le cas des courbes elliptiques sera effleuré, mais un traitement plus en profondeur mériterait un chapitre en soi, voir [18, chapitre V] et [112].

Soit donné un groupe cyclique $(G, +)$ d'ordre N, engendré par un élément P. Pour un élément $Q \in G$, on désigne par $\log Q = \log_P Q$ l'entier x (unique modulo N) tel que $Q = xP$, son *logarithme discret* ; le *problème du logarithme discret (DLP)* dans G est de calculer x étant donné Q.

Un cryptosystème est dit *fondé sur le DLP* si résoudre le DLP casserait le cryptosystème ; on souhaiterait que la réciproque soit vraie aussi, mais elle n'est démontrée pour aucun système. Des réductions de la sécurité cryptographique se font généralement à des problèmes potentiellement plus faciles que le DLP, tels que le problème de Diffie–Hellman calculatoire (CDH) ou décisionnel (DDH), voir leur définition et la discussion à la section 2.2.2.

La figure 4.1 montre la complexité du problème du logarithme discret en fonction de N telle qu'elle se présente typiquement dans un certain nombre de groupes. Dans la suite de ce chapitre, nous allons examiner successivement ces algorithmes de plus en plus performants, mais en même temps de plus en plus limités quant aux groupes auxquels ils s'appliquent.

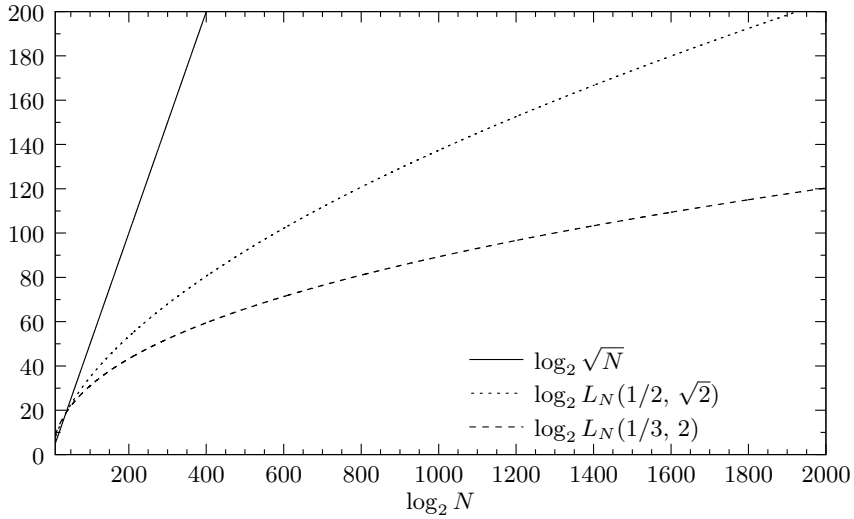


FIG. 4.1 – Complexités de différents DLP

4.1 Algorithmes exponentiels

4.1.1 Algorithmes génériques

Un certain nombre d'algorithmes permettent de calculer des logarithmes discrets en utilisant uniquement des opérations dans le groupe G, indépendamment de la représentation de ses éléments.

Notons pour commencer que la version décisionnelle du DLP est simple : étant donnés Q et un candidat x pour son logarithme, il suffit de calculer xP et de le comparer à Q pour décider si la valeur de x est bonne. Cette observation rend possible une recherche exhaustive en $O(N)$ opérations dans le groupe.

Si de plus tout élément admet un représentant unique, on peut passer à $O(\sqrt{N})$. L'algorithme des *pas de bébé et pas de géants* de Shanks [172] calcule et stocke d'abord les pas de bébé iP pour $0 \leqslant i < \lceil\sqrt{N}\rceil$; puis, les pas de géants $Q - j\lceil\sqrt{N}\rceil$ pour $0 \leqslant j < \lceil\sqrt{N}\rceil$ sont calculés et comparés aux éléments stockés. Dès qu'une *collision* $iP = Q - j\lceil\sqrt{N}\rceil$ se produit, on en déduit que $x = i + j\lceil\sqrt{N}\rceil$. Cet algorithme déterministe demande de stocker $O(\sqrt{N})$ éléments.

Une approche probabiliste due à Pollard [154] permet en fin de compte de se passer des besoins de stockage. L'idée de base est de calculer des combinaisons linéaires aléatoires $R_i = a_i P + b_i Q$; quand on observe une collision $R_i = R_j$, on en déduit que $x = -\frac{a_j - a_i}{b_j - b_i} \bmod N$ si $b_j - b_i$ est inversible modulo N ; sinon, on obtient l'information partielle $x \bmod N/\operatorname{pgcd}(N, b_j - b_i)$. Tel quel, cet algorithme marche en temps attendu de $O(\sqrt{N})$, mais a encore besoin de $O(\sqrt{N})$ en espace. En remplaçant le choix aléatoire des a_i et b_i par une marche pseudo-aléatoire de sorte que R_{i+1} dépend uniquement de R_i et en cherchant une collision uniquement de la forme $R_i = R_{2i}$, on arrive à l'al-

gorithme ϱ de Pollard, prenant heuristiquement un temps de $O(\sqrt{N})$ en stockant un nombre constant d'éléments du groupe. Pour une analyse plus poussée, voir [177].

Alternativement, on peut utiliser une méthode admettant une parallélisation facile due à Oorschot et Wiener [151]. On définit d'abord les *points distingués* par une propriété facilement reconnaissable et arrivant avec une probabilité bien contrôlée, comme un certain nombre de zéros dans leur représentation binaire. Plusieurs marches pseudo-aléatoires sont alors démarrées en parallèle à partir de points de départ différents. Dès qu'un point distingué est atteint, il est communiqué à une machine centrale qui le stocke et qui fait la recherche de collisions uniquement sur ces éléments. Notons que l'existence d'un représentant unique est cruciale pour les algorithmes en $\sqrt{N}$: si la recherche d'une collision devrait se faire à l'aide de tests d'égalité pour chaque couple d'éléments stockés, la complexité monterait de nouveau à $O(N)$.

Une astuce classique, décrite dans [153], consiste à réduire le DLP en une suite de logarithmes discrets dans les sous-groupes d'ordre p de G pour p premier divisant N. Notons d'abord que si $p^e||N$, alors $x \bmod p^e = \log_{(N/p^e)\cdot P} \frac{N}{p^e} Q$; les restes chinois permettent alors de recoller les logarithmes discrets obtenus dans les sous-groupes de Sylow de G. Sans perte de généralité, on peut alors supposer que $N = p^e$. De la même façon, $x_0 = x \bmod p$ s'obtient comme $\log_{p^{e-1}P}(p^{e-1}Q)$; puis, $x_1 = \frac{x-x_0}{p} \bmod p$ est égal à $\log_{p^{e-1}P}(p^{e-2}(Q - x_0 P))$ et ainsi de suite, de sorte que la décomposition en base p de x est obtenue de proche en proche en calculant des logarithmes discrets dans le sous-groupe d'ordre p de G. Combiné avec les algorithmes de complexité racine carrée du paragraphe précédent, des logarithmes discrets sont obtenus avec

$$O\left(\sum_{p^e||N} e\sqrt{p}\right)$$

opérations dans le groupe. C'est donc essentiellement la racine carrée du plus grand facteur premier de N qui détermine la sécurité maximum qu'un cryptosystème fondé sur le logarithme discret peut potentiellement atteindre. Pour N premier, on arrive à la droite de la figure 4.1.

4.1.2 Bornes inférieures

Il se pose alors la question de savoir si la difficulté du DLP peut être garantie. Nechaev et Shoup dans [148, 175] donnent une réponse partielle : si on n'admet que des opérations de groupe et que N est premier, alors il faut $\Omega(\sqrt{N})$ opérations pour obtenir un logarithme discret avec une probabilité non négligeable. Dès lors, pour dépasser cette borne, un algorithme doit tenir compte de la représentation particulière des éléments qui distingue G du groupe cyclique de cardinal N abstrait.

Ouvrons une petite parenthèse pour le problème de Diffie–Hellman (CDH), qui consiste à calculer abP étant donnés P, aP et bP. Dans le même article [175] Shoup montre qu'un algorithme générique, ne faisant qu'exécuter la loi de groupe, nécessite aussi $\Omega(\sqrt{N})$ opérations dans un groupe de cardinal premier N. Même le problème décisionnel DDH a alors la même complexité minimum.

Maurer et Wolf ont démontré l'équivalence entre CDH et DLP indépendamment de leur difficulté dans [134]. Ils considèrent le cas de N premier tel qu'il existe un groupe auxiliaire H, algébrique sur $\mathbb{F}_N$, dans lequel $\mathbb{F}_N$ se plonge dans un sens probabiliste (l'image d'un élément a le droit de ne pas exister, mais dans ce cas, l'élément légèrement perturbé doit posséder une image). Par exemple, H peut être une courbe elliptique définie sur $\mathbb{F}_N$, et l'image de $x \in \mathbb{F}_N$ est donnée par un point sur H d'abscisse x, s'il existe ; on peut perturber en passant à $x + e$. Maintenant, si le DLP dans H se résout par un algorithme algébrique faisant n opérations de groupe, on peut résoudre le DLP dans G par essentiellement n appels à un oracle résolvant CDH dans G. Si l'ordre de H est suffisamment friable (c'est-à-dire qu'il n'a que de facteurs premiers de l'ordre d'une puissance de $\log N$), il y a alors par les algorithmes de la section 4.1.1 appliqués à H une équivalence polynomiale entre le CDH et le DLP dans G. Remarquons que la démonstration utilise uniquement le cardinal du groupe, et non la représentation concrète de ses éléments. En acceptant l'heuristique que les entiers dans l'intervalle de Hasse autour de N se factorisent comme des entiers aléatoires de cette taille, et en utilisant que tous les cardinaux dans l'intervalle de Hasse apparaissent effectivement d'après le corollaire 1.14, Maurer et Wolf montrent l'existence d'un groupe auxiliaire elliptique H tel que la réduction devient polynomiale. Le trouver par multiplication complexe, par contre, pourrait prendre un temps exponentiel. En revanche, en admettant une réduction sous-exponentielle en $L(1/2)$, il devrait être possible de trouver un groupe auxiliaire dans le même temps.

Pour conclure ce chapitre sur les algorithmes exponentiels, mentionnons que ce sont pour le moment les seuls à s'appliquer à toutes les courbes elliptiques. Quelques courbes elliptiques particulières admettent des plongements dans d'autres groupes où le logarithme discret est plus facile à obtenir, mais elles sont de très faible densité : il s'agit des courbes supersingulières et autres courbes à faible degré de plongement dans le groupe multiplicatif d'un corps fini [136, 78], déjà rencontrées au chapitre 2.3 ; des sous-groupes d'ordre p de courbes définies sur $\mathbb{F}_q$ de caractéristique p, qui se plongent dans le groupe additif $(\mathbb{F}_q, +)$ [160, 169, 176] ; et de courbes elliptiques se plongeant par descente de Weil dans une courbe hyperelliptique de petit genre suivant une suggestion de Frey [77], voir [86, 45] et [112] et sa bibliographie. Ce dernier résultat donne une autre motivation pour regarder de plus près les courbes de genre plus grand que 1.

4.2 Algorithmes sous-exponentiels en $L(1/2)$

Dans ce chapitre, je reprend essentiellement des résultats de ma thèse de doctorat [54], publiés également en tant que [57, 70, 60, 56].

4.2.1 La fonction sous-exponentielle

Par une *fonction sous-exponentielle* on pourrait comprendre une fonction qui croît moins vite que toute exponentielle, mais plus vite que tout polynôme. Dans le cas qui nous concerne, une définition plus restrictive s'impose.

Définition 4.1 La fonction sous-exponentielle avec paramètres $\alpha \in (0,1)$ et $c > 0$ par rapport à l'argument N est donnée par

$$L_N(\alpha, c) = e^{c(\log N)^\alpha (\log\log N)^{1-\alpha}}.$$

Pour simplifier, nous noterons

$$L_N(\alpha) = \{L_N(\alpha, c) : c > 0\}$$

et omettrons le N quand aucune confusion n'est possible.

Notre attention se focalisera dans la suite sur le paramètre α, qui a la plus grande influence sur la croissance de la fonction. Le paramètre c sera fréquemment intitulé la *constante* de la fonction sous-exponentielle, bien qu'il apparaisse à l'exposant, de sorte que son influence est loin d'être négligeable.

La notation traditionnelle L_N est un peu malheureuse, car en termes de complexité, il faut s'imaginer un problème ayant N entrées étant spécifiées par $\log N$ bits, et la sous-exponentialité est donnée par rapport à $\log N$ plutôt qu'à N :

- pour le cas extrême exclu par la définition $\alpha = 0$, on aurait le polynôme $\log^c N$;
- l'autre cas extrême $\alpha = 1$ mène à l'exponentielle N^c ;
- en tant que valeurs intermédiaires pour α, essentiellement $1/2$ et $1/3$ apparaissent dans le cadre du logarithme discret et de la factorisation. Deux fonctions typiques sont tracées à la figure 4.1.

On vérifie aisément les règles de calcul suivantes :

$$\begin{aligned} &L_N(\alpha, c_1) \cdot L_N(\alpha, c_2) = L_N(\alpha, c_1 + c_2) \\ &\log^k N \in L(\alpha, o(1)) \text{ pour tout } k, \text{ et plus généralement,} \\ &L_N(\beta, d) \in L_N(\alpha, o(1)) \text{ pour } \beta < \alpha. \end{aligned} \tag{4.1}$$

En particulier, si une opération polynomiale est répétée $L_N(\alpha, c)$ fois, la complexité en résultant est dans $L_N(\alpha, c + o(1))$; c'est pourquoi la définition 4.1 se fait souvent en ajoutant ce $o(1)$.

4.2.2 Un algorithme pour les corps finis

Les algorithmes sous-exponentiels pour le logarithme discret se déroulent généralement en deux étapes : dans la phase de *crible* ou *collecte de relations*, une matrice entière est remplie de *relations* ; la phase de l'*algèbre linéaire* résout le système modulo le cardinal du groupe et donne certains logarithmes ; éventuellement, une troisième phase, plus légère, est nécessaire pour calculer des *logarithmes individuels*. Ce type d'algorithme est communément appelé « calcul d'index » (« index calculus » en anglais), une dénomination malheureuse. En fait, « index » est traditionnellement utilisé comme synonyme de « logarithme » ; déjà l'encyclopédie de Diderot et d'Alembert, publiée entre 1751 et 1772, donne la définition suivante : « *Index*, en terme d'Arithmétique, est la même chose que la caractéristique ou l'exposant d'un logarithme. Voyez Logarithme. » [44].

L'idée de base de créer des relations et de les combiner linéairement pour calculer des logarithmes discrets (et pour factoriser) a été publiée par Kraïtchik dans les années 20 [124, chapitre 5, §§14-16]. En 1979, l'algorithme est redécouvert par Adleman et présenté avec l'analyse de sa complexité sous-exponentielle pour les corps finis premiers. Il se généralise facilement aux corps $\mathbb{F}_{2^m}$ (la raison en est expliquée à la section 4.2.5), plus proche de la situation que nous allons rencontrer pour les courbes. En voici une version légèrement remaniée.

Le problème est donc, étant donnés P un élément primitif de $\mathbb{F}_{2^m}$ et $Q \in \mathbb{F}_{2^m}^\times$, de renvoyer x tel que $Q = P^x$. Il convient de représenter $\mathbb{F}_{2^m}$ comme $\mathbb{F}_2[X]/(f)$, où f est un polynôme irréductible sur $\mathbb{F}_2$ de degré m. Ainsi, tout élément du corps $\mathbb{F}_{2^m}$ peut être vu comme un polynôme binaire de degré plus petit que m. Cette représentation du corps par des polynômes introduit des notions qui *a priori* n'ont pas de sens dans un corps : on peut désormais parler d'éléments irréductibles, le degré des polynômes induit une notion de taille sur les éléments, et on a une factorisation unique des éléments en éléments irréductibles. En fait, la factorisation n'est plus unique dès qu'on lève la restriction sur les degrés, plusieurs éléments de $\mathbb{F}_2[X]$ pouvant représenter le même élément du corps fini.

Algorithme 4.2
ENTRÉE: *P élément primitif de $\mathbb{F}_{2^m} = \mathbb{F}_2[x]/(f)$, $Q \in \mathbb{F}_{2^m}^\times$*
SORTIE: *x tel que $Q = P^x$*

0. *soit $N = 2^m - 1$; fixer une* borne de friabilité $B \in \mathbb{N}$, *et calculer la* base des facteurs $\mathcal{F} = \{p_0, \ldots, p_n\}$ *des polynômes irréductibles sur $\mathbb{F}_2$ de degré au plus B augmentés de $P = p_0$; préparer une matrice vide A avec n colonnes et r lignes et un vecteur vide b avec r lignes pour $r \in O(n)$*
1. ***répéter*** *pour $i = 1, \ldots, r$*
   répéter
   tirer aléatoirement des exposants $e_{ij} \in \{0, \ldots, N-1\}$ pour $j = 0, \ldots, n$
   calculer $\prod_{j=0}^n p_j^{e_{ij}} \bmod f$
   si *le résultat se factorise sur $\mathcal{F}$ en tant que $\prod_{j=0}^n p_j^{f_{ij}}$,*
   on a la relation $\prod_{j=1}^n p_j^{a_{ij}} = P^{-a_{i0}}$ *dans $\mathbb{F}_{2^m}$ avec $a_{ij} = e_{ij} - f_{ij}$;*
   ajouter $(a_{ij})_{j=1}^n$ à la matrice A et $-a_{i0}$ au vecteur b
   jusqu'à *succès pour une nouvelle relation*
2. *résoudre le système $Ay = b$ modulo N, de sorte que $y_j = \log_P p_j$*
3. *créer une relation supplémentaire $Q \prod p_j^{e_j} = \prod p_j^{f_j}$; renvoyer $x = \sum (f_j - e_j) y_j$*

Cette version sépare les phases 2 de l'algèbre linéaire de la phase 3 du calcul d'un logarithme individuel, qui peut être répétée autant de fois que souhaitée. Alternativement, il suffit d'ajouter Q dans la base des facteurs et de s'arrêter après l'étape 2.

Concernant la complexité de l'algorithme, elle dépend essentiellement de la probabilité qu'un polynôme aléatoire de degré au plus $m-1$ se décompose sur la base des facteurs, autrement dit, qu'il est *B-friable*. Si la taille n de la base des facteurs est polynomiale,

cette probabilité décroît exponentiellement ; pour qu'elle ne décroisse que polynomialement, il faut une base des facteurs de taille exponentielle. L'optimum se trouve entre les deux ; plus précisément, pour $n \in L(1/2)$, la probabilité d'avoir une relation est dans $1/L(1/2)$. Ainsi, le nombre d'itérations pour obtenir une relation est dans $L(1/2)$, et il faut répéter ce processus $O(n) \subseteq L(1/2)$ fois pour remplir la matrice. Toutes les opérations de base de l'algorithme étant polynomiales ou en $L(1/2)$ (par exemple l'algèbre linéaire, qui est polynomiale en n), les règles de calcul (4.1) montrent que la complexité totale de l'algorithme est en $L(1/2)$. Pour une discussion plus générale de la friabilité et une analyse plus fine de la complexité, voir la section 4.2.5.

Notons encore que ce résultat n'est pas en contradiction avec les bornes inférieures exponentielles de la section 4.1.2, l'algorithme sous-exponentiel étant loin d'être générique : il utilise bien des propriétés particulières de la représentation concrète du groupe $\mathbb{F}_{2^m}^\times \simeq \mathbb{Z}/N\mathbb{Z}$ par des polynômes binaires.

4.2.3 Arithmétique des jacobiennes de courbes

Par les résultats de la section 4.1.2, il est clair qu'il faut regarder de plus près la représentation des éléments et la loi du groupe associé à une courbe algébrique. Pour arriver à l'équivalent de l'algorithme 4.2, notre but sera de montrer que les éléments se comportent essentiellement comme des polynômes.

Partons pour le moment d'une courbe $\mathcal{C}$ définie par une équation $C(X, Y) = 0$ sur un corps *algébriquement clos* K. On lui associe un entier g, son *genre*, qui mesure en quelque sorte à quel point la courbe est complexe ; il est étroitement lié au degré du polynôme C si celui-ci est « raisonnable ». Par exemple, une courbe *hyperelliptique* est donnée en caractéristique différente de 2 par un polynôme non singulier $C = Y^2 - X^{2g+1} + f(X)$ avec f de degré au plus $2g$; le cas $g = 1$ correspond à une courbe elliptique. Une généralisation est donnée par les courbes *superelliptiques*, définies par un polynôme non singulier $Y^a - X^b + f(X)$ avec f de degré plus petit que b et $\operatorname{pgcd}(a, b) = 1$ quand la caractéristique de K est première avec a. En admettant certains termes mixtes en X et Y, on arrive au cas le plus général considéré ici, les courbes $\mathcal{C}_{a,b}$, définies par un polynôme irréductible non singulier

$$Y^a - X^b + \sum_{(i,j):ai+bj<ab} c_{ij} X^i Y^j$$

avec $\operatorname{pgcd}(a, b) = 1$ en caractéristique ne divisant ni a, ni b. Le genre de ces courbes est $g = \frac{(a-1)(b-1)}{2}$.

Hormis pour les courbes elliptiques, le groupe associé ne s'exprime plus directement sur les points. À la place, il faut travailler dans la *jacobienne* $J(\mathcal{C})$ de la courbe, une variété abélienne. En pratique, on préfère travailler avec le groupe isomorphe (que nous noterons encore $J(\mathcal{C})$) des classes de diviseurs de degré 0. Définissions pour cela le groupe des *diviseurs* de $\mathcal{C}$,

$$\operatorname{Div}(\mathcal{C}) = \left\{ \sum_{P \in \mathcal{C}} m_P P : m_P \in \mathbb{Z} \text{ presque tous nuls} \right\},$$

par des sommes formelles finies, avec multiplicités potentiellement négatives, de points sur un modèle projectif non singulier de la courbe ; alternativement, les points peuvent être considérés comme les places du corps de fonctions $K(\mathcal{C}) = K(X)[Y]/(C)$. Le *degré* d'un diviseur est donné par

$$\deg\left(\sum m_P P\right) = \sum m_P.$$

Associons à une fonction rationnelle f de $K(\mathcal{C})$ son *diviseur principal*, contenant ses zéros avec multiplicités positives et ses pôles avec multiplicités négatives ; c'est un diviseur de degré 0. Si $\mathrm{Div}^0(\mathcal{C})$ désigne le groupe des diviseurs de degré 0 et $\mathrm{Prin}(\mathcal{C})$ son sous-groupe de diviseurs principaux, la jacobienne est donnée par

$$J(\mathcal{C}) = \mathrm{Div}^0(\mathcal{C})/\,\mathrm{Prin}(\mathcal{C}).$$

Fixons un point ∞ sur la courbe ; il y a alors un isomorphisme naturel

$$\mathrm{Div}^0(\mathcal{C}) \to \mathrm{Div}'(\mathcal{C}), \quad \sum_P m_P P \mapsto \sum_{P\neq\infty} m_P P,$$

où $\mathrm{Div}'(\mathcal{C})$ est le sous-groupe de $\mathrm{Div}(\mathcal{C})$ de diviseurs ne contenant pas ∞ dans leur support. L'isomorphisme inverse s'obtient en ajoutant la bonne multiplicité de ∞ pour arriver à un degré 0. Par le théorème de Riemann–Roch, chaque classe de $J(\mathcal{C})$ peut alors être représentée par un unique diviseur *effectif* ou *positif* (c'est-à-dire aux coefficients positifs ou nuls) de $\mathrm{Div}'(\mathcal{C})$ de degré minimal, appelé *réduit*, et ce degré n'excède pas g.

Dans le cas des courbes hyperelliptiques ou plus généralement $\mathcal{C}_{a,b}$, on peut prendre comme ∞ l'unique place à l'infini (c'est justement la spécificité de ces courbes), et alors se contenter de ne travailler qu'avec des points affines. En termes de corps de fonctions, cela veut dire que $J(\mathcal{C})$ est le groupe de classes de l'anneau des entiers $K[\mathcal{C}] = K[X,Y]/(C)$ de $K(\mathcal{C})$.

Cette observation permet d'utiliser la représentation standard des idéaux dans un anneau de Dedekind ; tout diviseur D de Div' s'écrit en tant qu'idéal sous la forme

$$(d)(u,w)$$

avec d, $u \in K[X]$ et $w \in K[\mathcal{C}]$ unitaire et de degré moins de a en Y pour une courbe $\mathcal{C}_{a,b}$; comme nous travaillons modulo l'équivalence induite par les diviseurs principaux et que (d) est principal, seuls les diviseurs de la forme (u,w) apparaîtront dans les algorithmes. Le polynôme u s'annule (avec multiplicités) dans les coordonnées X des points dans D, tandis que le polynôme bivarié w interpole (encore avec multiplicités) ces points.

L'algorithme réalisant l'addition dans une jacobienne manipule alors des polynômes et procède en deux étapes : la *composition* effectue l'addition des diviseurs respectivement la multiplication des idéaux, en enlevant éventuellement des idéaux (d) de $K[X]$ qui apparaissent ; il s'agit essentiellement de l'interpolation de Lagrange. La *réduction* calcule pour le diviseur génériquement de degré $2g$ qui en résulte son représentant réduit de degré au plus g ; cette étape dépend fortement de la courbe.

Passons maintenant aux courbes définies sur un corps fini $K = \mathbb{F}_q$. Contrairement à ce qu'on pourrait penser, il ne suffit pas de faire la même construction que pour K algébriquement clos ; en additionnant deux éléments de la jacobienne ne contenant que des points définis sur $\mathbb{F}_q$, la réduction peut faire apparaître des points définis sur une extension. Pour y remédier, il faut considérer l'automorphisme de Frobenius de $\mathbb{F}_q$, $x \mapsto x^q$, qui donne de façon évidente l'endomorphisme $\varphi : (x, y) \mapsto (x^q, y^q)$ sur les points de la courbe définis sur $\overline{\mathbb{F}}_q$, tel qu'il a déjà été considéré à la section 1.1.5. Il s'étend aux diviseurs et préserve la principalité. Nous pouvons donc définir les groupes Div, Div^0 et Prin comme ensembles de diviseurs définis sur $\overline{\mathbb{F}}_q$ comme ci-dessus, avec la restriction supplémentaire que les éléments sont invariants sous le Frobenius. Le passage à Div' nécessite en plus que ∞ soit un point $\mathbb{F}_q$-rationnel, ce qui est le cas pour les courbes qui nous intéressent. Nous arrivons ainsi encore une fois au groupe de classes de $K[\mathcal{C}]$. Les éléments de la jacobienne seront représentés par des idéaux (u, w) comme ci-dessus, avec maintenant u et w des polynômes à coefficients dans $\mathbb{F}_q$. Les algorithmes de composition et de réduction restent les mêmes ; leur nature algébrique entraîne qu'ils n'ont aucune « conscience » du corps sur lequel ils travaillent.

Des algorithmes d'addition efficaces pour des courbes générales ont été développés par Heß et par Khuri-Makdisi [110, 120]. Concernant l'arithmétique hyperelliptique, voir [29, 55, 126, 92], les courbes superelliptiques, [82, 14], les courbes $\mathcal{C}_{a,b}$ et en particulier $\mathcal{C}_{3,4}$, [6, 7, 73, 13, 1].

Notons encore une simple conséquence de la généralisation de Weil [181] du théorème de Hasse (1.6), que le cardinal de la jacobienne d'une courbe $\mathcal{C}$ de genre g définie sur $\mathbb{F}_q$ satisfait

$$(\sqrt{q} - 1)^{2g} \leqslant |J(\mathcal{C})| \leqslant (\sqrt{q} + 1)^{2g}.$$

On en déduit que l'immense majorité des éléments de la jacobienne est représentée par (u, w) avec $\deg u = g$, $w = Y - v(X)$, et $\deg v = g - 1$.

4.2.4 L'algorithme d'Adleman–DeMarrais–Huang pour les courbes hyperelliptiques

Le premier algorithme sous-exponentiel pour calculer des logarithmes discrets dans les courbes hyperelliptiques de grand genre définies sur un corps fini $K = \mathbb{F}_q$ est dû à Adleman, DeMarrais et Huang [3]. Il diffère de l'algorithme 4.2 essentiellement par la génération de relations. En effet, les diviseurs principaux étant nuls dans la jacobienne, il suffit de tirer au hasard des polynômes de la forme $Y - v(X)$ et d'en calculer le diviseur ; si celui-ci est friable, on a directement une relation. C'est le cas quand la norme de $Y - v$ par rapport à l'extension de corps de fonctions $K(\mathcal{C})/K(X)$ est friable ; en supposant heuristiquement que ces normes se comportent comme des polynômes aléatoires, les auteurs arrivent à une complexité de $L_{q^g}(1/2)$ quand $(2g + 1)^{0,98} \geqslant \log q$.

Le résultat est heuristique pour une deuxième raison. Implicitement, l'algorithme 4.2 décrit le groupe G comme $\mathbb{Z}^n$ quotienté par le réseau formé par les lignes de la matrice. Il n'est pas clair si les bornes imposées sur le degré de v permettent d'obtenir un réseau suffisamment dense pour qu'il y ait un isomorphisme.

4.2.5 Un cadre général

J'ai donné dans [57] le premier algorithme pour le logarithme discret dans les courbes hyperelliptiques de grand genre avec une complexité sous-exponentielle prouvée, fondée sur le théorème de friabilité de [70]. Son temps de calcul dépend de la croissance du genre g par rapport à la taille du corps $\mathbb{F}_q$. Un résultat semblable pour l'infrastructure d'un corps de fonctions quadratique réel se trouve dans [146], lui aussi ayant besoin du résultat de friabilité de [70].

Dans [60], nous développons une approche générale pour décrire les groupes admettant un algorithme de logarithme discret en $L(1/2)$. Ce cadre s'applique à un grand nombre de groupes proposés en cryptographie, notamment les courbes hyperelliptiques.

Soit donné un ensemble $\mathcal{P}$ d'éléments dits *premiers*, et appelons $\mathbb{M}$ le monoïde libre sur $\mathcal{P}$. Pour une relation d'équivalence $\sim$ sur $\mathbb{M}$ compatible avec l'addition, soit $G = \mathbb{M}/\sim$ un groupe. Supposons en plus une fonction de taille $\deg : \mathcal{P} \to \mathbb{R}^{\geqslant 1}$, étendue de façon évidente à $\mathbb{M}$, ce qui permet de définir la base des facteurs $\mathcal{F}$ comme l'ensemble des petits éléments premiers de taille bornée par une borne de friabilité B. Si chaque élément de G possède un représentant canonique dans $\mathbb{M}$ (en général, l'élément de plus petite taille), la décomposition trivialement unique en éléments premiers de $\mathbb{M}$ se transfère à G. Si de plus quelques conditions techniques concernant par exemple la calculabilité, la taille en bits des éléments et la génération de G par $\mathcal{F}$ sont satisfaites, l'algorithme 4.2 s'applique directement.

Ces notions ont été introduites par Knopfmacher dans [122], qui appelle $\mathbb{M}$ un *semi-groupe arithmétique* et G une *formation arithmétique*. Des exemples concrets sont fournis par les corps premiers $\mathbb{F}_p$, où $\mathbb{M} = \mathbb{Z}$, deg est le logarithme et $\sim$ est l'équivalence modulo p ; et les corps $\mathbb{F}_{p^m} = \mathbb{F}_p[X]/(f)$, où $\mathbb{M} = \mathbb{F}_p[X]$, deg est le degré, et $\sim$ l'équivalence modulo f. Mais aussi les groupes de classes de corps de nombres, où $\mathbb{M}$ est l'ensemble des idéaux entiers, deg est le logarithme de la norme et $\sim$ l'équivalence modulo idéaux principaux. Et finalement les jacobiennes de courbes $\mathcal{C}$ avec un seul point à l'infini sur un corps fini $\mathbb{F}_q$, où $\mathbb{M} = \mathrm{Div}'(\mathcal{C})$, deg est le degré d'un diviseur et $\sim$ l'équivalence modulo diviseurs principaux. Concernant l'ensemble $\mathcal{P}$, il s'agit ici des *diviseurs premiers*, qui sont les orbites sous le Frobenius d'un point défini sur une extension de $\mathbb{F}_q$.

Il reste à s'assurer du temps de calcul de l'algorithme. Pour cela, il faut qu'une base des facteurs de taille dans $L(1/2)$ implique une probabilité de friabilité dans $1/L(1/2)$. Des résultats correspondants se trouvent par exemple dans [155] pour $\mathbb{F}_p$, dans [15] pour $\mathbb{F}_{2^m}$, dans [170] pour les groupes de classes de corps quadratiques imaginaires (sous l'hypothèse de Riemann généralisée) et dans [70] pour les courbes hyperelliptiques de grand genre. Nous démontrons alors dans [60], en nous servant de l'uniformité des relations créées dans l'algorithme 4.2, que dans tous ces cas l'algorithme a une complexité en $L(1/2)$.

En regardant de plus près les théorèmes de friabilité, on réalise qu'il s'agit en fait toujours du même résultat : pour une base de friabilité de cardinal $L_N(1/2, c)$, un élément de taille $\log N$ a une chance de $1/L_N(1/2, 1/(2c) + o(1))$ d'être friable. Ce résultat peut être démontré dans $\mathbb{M}$ si on suppose l'équivalent du théorème des nombres premiers : il faut que le nombre d'éléments premiers de taille bornée par k soit de l'ordre de $\frac{q^k}{k}$

pour un q ; voir [133, 132]. Dans ce cas, la proportion d'éléments friables par rapport à la borne y parmi tous les éléments de taille bornée par x est asymptotiquement (avec quelques contraintes sur la croissance de y par rapport à x) donnée par la valeur de la fonction ϱ de Dickmann–De Bruijn en $u = \frac{x}{y}$; et de Bruijn a montré en 1951 dans [27] que $1/\varrho(u) \in e^{(1+o(1))u \log u}$, ce qui fait le lien avec la sous-exponentialité.

À cause de la relation d'équivalence, le résultat de friabilité pour $\mathbb{M}$ ne se transfère pas directement à G. Dans une courbe, par exemple, il y a des diviseurs de degré g qui ne sont pas réduits. Néanmoins, les résultats de [170, 70] donnent des exemples de formations arithmétiques dans lesquelles on observe ce même comportement de friabilité ; il est donc raisonnable de l'accepter comme heuristique aussi dans d'autres contextes.

Étant donné le résultat de friabilité, la complexité de l'algorithme 4.2 se vérifie aisément. Prenons la taille de la base des facteurs comme $n = L_N(1/2, d + o(1))$ avec un paramètre d à déterminer, et N le cardinal du groupe ; pour les courbes, il convient de poser $N = q^g$ comme approximation du cardinal. Si la factorisation d'un élément sur la base des facteurs se fait en temps $L_N(1/2, o(1))$ (ce qui est le cas pour tous les groupes considérés), le temps de création de $O(n)$ relations de la première étape est de

$$L_N(1/2, o(1))L_N(1/2, 1/(2d) + o(1))L_N(1/2, d + o(1)) = L_N(1/2, d + 1/(2d) + o(1))$$

par (4.1). L'algèbre linéaire de la phase 2 se fait sur une matrice creuse de l'ordre de $L_N(1/2, d + o(1))$ avec $L_N(1/2, d + o(1))$ entrées, chaque relation ayant de l'ordre de $\log N$ premiers. Les algorithmes de Lanczos ou Wiedemann [125, 183] se déroulent alors en temps $L_N(1/2, 2d + o(1))$. Ainsi, le temps total de l'algorithme devient

$$L_N(1/2, \max(d + 1/(2d), d) + o(1)).$$

Cette quantité est minimisée pour $d = \sqrt{2}/2$, ce qui résulte en la complexité

$$L_N\left(1/2, \sqrt{2} + o(1)\right).$$

Pour les courbes hyperelliptiques, ce temps de calcul est valable quand la taille q du corps reste fixe et que le genre tend vers l'infini. On peut admettre une croissance maîtrisée du corps ; pour $g \geqslant \vartheta \log q$, suivant l'analyse de [57], un temps de calcul de

$$L_N\left(1/2, \sqrt{2} + \frac{2}{\sqrt{\vartheta}} + o(1)\right)$$

est obtenu dans [60].

Une supposition implicite dans l'algorithme 4.2 n'est pas forcément satisfaite pour les courbes hyperelliptiques ; il faut que le groupe soit cyclique et d'ordre N connu. Si ce n'est pas le cas, il est possible de remplacer la résolution d'un système linéaire par le calcul des formes normales d'Hermite et de Smith de la matrice, ce qui résulte en une complexité en $L_N(1/2, c)$ pour une moins bonne constante c, voir [56].

Un algorithme de complexité sous-exponentielle prouvée en $L_{q^g}(1/2 + \varepsilon)$ pour une grande classe de courbes, non limitée aux courbes hyperelliptiques, est donnée par Couveignes dans [37] ; il fait l'hypothèse que la courbe contient un point $\mathbb{F}_q$-rationnel et que le

cardinal de la jacobienne est bornée par $q^{g+O(\sqrt{g})}$. L'approche est assez différente de celle présentée ci-dessus, reposant sur une double randomisation, à la fois dans la combinaison d'éléments de la base de facteurs et dans le choix d'une fonction d'un certain espace de Riemann–Roch. Un algorithme sans restrictions sur la courbe d'entrée est donné par Heß dans [111], qui arrive ainsi à une complexité prouvée en $L_{q^g}(1/2)$ pour toutes les courbes de grand genre.

À première vue, ces algorithmes ne peuvent marcher en genre petit. Pour les courbes elliptiques, par exemple, ils perdent complètement leur sens : comme tous les diviseurs sont de degré 1, la base des facteurs est soit vide, soit elle contient tous les points de la courbe, et la complexité devient de l'ordre de q^2, bien pire que la recherche exhaustive. Mais la situation change quand on considère des courbes de genre fixé, mais un peu plus grand, une approche développée d'abord par Gaudry dans [91], puis dans [178, 90]. Il s'avère que pour un genre 3 et supérieur, les algorithmes sont exponentiels, mais plus rapide que des algorithmes génériques en la racine carrée du cardinal. Ainsi, pour obtenir un cryptosystème de sécurité équivalente à un système elliptique, ou hyperelliptique avec une courbe de genre 2, il faut augmenter la taille du groupe d'un facteur $\frac{g^2}{4g-4}$ dès le genre 3, ce qui rend les courbes de genre 4 et supérieur inintéressantes pour la cryptographie ; l'intérêt du genre 3 est douteux.

4.3 Algorithmes sous-exponentiels en $L(1/3)$

Dans cette section, je donne un aperçu de mes derniers résultats obtenus en collaboration avec Pierrick Gaudry. Notre article [61] a été sélectionné parmi les trois meilleurs soumissions au colloque Eurocrypt 2007, et nous avons été sollicités de soumettre une version étendue au Journal of Cryptology*.*

4.3.1 Le crible des corps de fonctions

Suivant le progrès dans les algorithmes de factorisation, une complexité de $L(1/3)$ a été établie également pour le calcul de logarithmes dans les corps finis. Il s'agit d'abord de l'algorithme de Coppersmith pour $\mathbb{F}_{2^m}$ de [35], qui peut-être vu comme un cas particulier du *crible des corps de fonctions* d'Adleman [2], adapté aux corps $\mathbb{F}_{p^m}$ avec p petit. Le cas de $\mathbb{F}_p$ respectivement $\mathbb{F}_{p^m}$ avec m petit est traité par le *crible des corps de nombres* ou *crible algébrique* de Gordon [97]. Récemment, il a été montré dans [119] que les domaines d'application des deux algorithmes se recoupent, de sorte qu'on dispose d'un algorithme en $L(1/3)$ pour tous les corps finis.

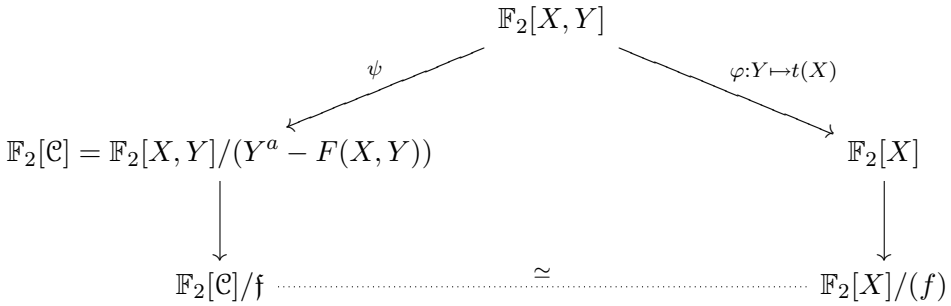
Le crible des corps de fonctions est d'un intérêt particulier dans notre contexte, car il fera surgir des liens avec l'algorithme pour les courbes de la section 4.3.2. L'observation de départ pour arriver à $L(1/3)$ est que les résultats de friabilité de la section 4.2.5 se généralisent en faisant varier la taille des éléments à décomposer et la borne de friabilité. Le théorème suivant est démontré dans [61] pour les courbes algébriques dont le genre croît suffisamment vite par rapport à une puissance de $\log q$, mais il est vrai en toute généralité.

Théorème 4.3 *Supposons une formation arithmétique de cardinal N comme à la section 4.2.5 dans laquelle la friabilité est régie par la fonction de Dickmann–De Bruijn. Soient $0 < \beta < \alpha \leqslant 1$ et $c,\ d > 0$. La probabilité qu'un élément de taille au plus $\log L_N(\alpha, c)$ soit friable par rapport à la base des facteurs des $L_N(\beta, d)$ plus petits premiers est donnée par*

$$1/L_N\left(\alpha - \beta, \frac{c}{d}(\alpha - \beta) + o(1)\right).$$

Le cas $\alpha = c = 1$ et $\beta = 1/2$ correspond à la situation de la section précédente, où il a servi pour démontrer des complexités en $L(1/2)$. Pour arriver à une complexité en $L(1/3)$, ce théorème n'ouvre qu'une voie : puisqu'il faut écrire la base des facteurs, on ne peut dépasser $\beta = 1/3$; alors, il faut baisser la taille des éléments à factoriser à $\log L(2/3)$.

Le crible des corps de fonctions y arrive en représentant le corps fini, disons $\mathbb{F}_{2^m}$, de deux façons différentes : premièrement, comme avant, par $\mathbb{F}_2[X]/(f)$ avec f irréductible de degré m. Deuxièmement, comme corps résiduel d'une place dans un corps de fonctions défini sur $\mathbb{F}_2$, donné par une courbe $\mathcal{C} : Y^a - F(X, Y) = 0$ de type $\mathcal{C}_{a,b}$ avec $b \approx a$. Supposons que (f) est totalement scindé dans $\mathbb{F}_2(\mathcal{C})$, et $\mathfrak{f} = (f(X), Y - t(X))$ un idéal de $\mathbb{F}_2[\mathcal{C}]$ au-dessus de (f). Alors, les homomorphismes partant de $\mathbb{F}_2[X, Y]$ et donnés d'une part par la réduction $\psi : \mathbb{F}_2[X, Y] \to \mathbb{F}_2[\mathcal{C}]$ modulo l'équation de la courbe, d'autre part par l'évaluation $\varphi : \mathbb{F}_2[X, Y] \to \mathbb{F}_2[X]$, $Y \mapsto t(X)$, sont compatibles avec les réductions modulo $\mathfrak{f}$ et f :



En prenant un polynôme w dans $\mathbb{F}_2[X, Y]$ dont les images à la fois sous ψ et φ sont friables, on arrive alors à une relation dans $\mathbb{F}_{2^m}$. (Des complications techniques sont introduites par le fait que $\mathbb{F}_2[\mathcal{C}]$ n'est pas principal, de sorte qu'au lieu de décomposer $\psi(w)$, on ne peut décomposer que l'idéal engendré.) La décomposition au niveau du corps de fonctions se fait aisément en notant qu'il suffit de tester la friabilité et de factoriser le cas échéant la norme de $\psi(w)$.

Le degré a de la courbe procure un nouveau degré de liberté ; quand les paramètres sont soigneusement choisis, le degré de la norme de $\psi(w)$ ainsi que celui de $\varphi(w)$ sont dans $\log L_{2^m}(2/3)$. À première vue, il y a une perte car il faut maintenant satisfaire deux conditions de friabilité simultanément ; mais cela ne joue que sur la constante de la fonction sous-exponentielle, et on arrive bien à une complexité en $L(1/3)$. Ce résultat repose sur l'heuristique selon laquelle la norme d'un polynôme dans $\mathbb{F}_2[\mathcal{C}]$ se comporte comme un polynôme aléatoire du même degré vis-à-vis de la friabilité.

4.3.2 Le cas des courbes

La question naturelle qui se pose alors est de savoir s'il est possible de passer à $L(1/3)$ également pour les jacobiennes de courbes sur un corps fini. Les parallèles entre corps finis et jacobiennes de courbes utilisées à la section 4.2.5 pour développer le cadre général des algorithmes en $L(1/2)$ laissent espérer arriver à une généralisation semblable pour $L(1/3)$. Or, la deuxième représentation de $\mathbb{F}_{2^m}$ comme corps résiduel dans un corps de fonctions (ou bien de $\mathbb{F}_p$ comme corps résiduel dans un corps de nombres) n'a pas de parallèle pour les jacobiennes. Il ne semble, par exemple, pas possible d'empiler une deuxième courbe au-dessus de la courbe donnée.

La solution que nous avons trouvée dans [61] retourne cet inconvénient à notre avantage : en effet, nous optons pour travailler *directement* dans les courbes telles qu'elles apparaissent dans le crible des corps de fonctions. L'algorithme n'est pas limité aux courbes $\mathcal{C}_{a,b}$. Soient a_0 et b_0 deux constantes positives quelconques. Nous considérons des familles de courbes absolument irréductibles de genre g sur un corps fini $\mathbb{F}_q$, de la forme

$$\mathcal{C} : Y^a + F(X, Y)$$

avec $F(X, Y) \in \mathbb{F}_q[X, Y]$ de degré b en X et au plus $a - 1$ en Y, où a et b sont bornés par

$$a < a_0\, g^{1/3}\mathcal{M}^{-1/3} \text{ et } b < b_0\, g^{2/3}\mathcal{M}^{1/3} \tag{4.2}$$

avec $\mathcal{M} = \frac{\log(g \log q)}{\log q} = \log_q(g \log q)$. Pour pouvoir appliquer les résultats de friabilité, il faut en plus que $g \geqslant (\log q)^\delta$ pour un $\delta > 2$.

Par exemple, on peut choisir $a_0 > 0$ arbitrairement, fixer $b_0 = \frac{2}{a_0}$ et considérer des courbes $C_{a,b}$ satisfaisant (4.2) ; ainsi, nous ne parlons pas de l'ensemble vide.

Les relations sont créées comme dans l'algorithme d'Adleman–DeMarrais–Huang de la section 4.2.4 : les diviseurs principaux étant nuls dans la jacobienne, il suffit de tirer au hasard des fonctions $w = r(X) + s(X)Y$ et de vérifier si leur diviseur est friable ; cela se fait en calculant la norme dans $\mathbb{F}_q[X]$ et en vérifiant que celle-ci est friable. Il suffit d'inclure dans la base des facteurs les diviseurs premiers de degré relatif 1 par rapport à l'extension de corps de fonctions $\mathbb{F}_q(\mathcal{C})/\mathbb{F}_q(X)$, qui ont une densité de Dirichlet égale à 1 (un joli parallèle avec la théorie des corps de classes, dont il a été question au chapitre 1).

En choisissant comme base des facteurs les $L_{q^g}(1/3, d)$ plus petits diviseur premiers et le degré de r et s comme $c\, g^{1/3}\mathcal{M}^{2/3}$, on s'assure de deux choses :

- Premièrement, la probabilité de friabilité de la norme est (heuristiquement) de $1/L(1/3, e/d + o(1))$ avec $e = (a_0 c + b_0)/3$.
- Deuxièmement, l'espace de crible est suffisamment grand. En effet, il serait possible d'augmenter la probabilité de friabilité en choisissant r et s de degré encore plus petit (dans l'extrême, comme des constantes) ; mais alors, le nombre de choix pour w serait tellement restreint qu'en moyenne, on n'obtiendrait pas une seule relation. Comme dans d'autres algorithmes sous-exponentiels, il faut donc s'assurer que le nombre de w disponibles est au moins égal au nombre de tests de friabilité à effectuer. C'est ce qui empêche de passer sous la barre de $L(1/3)$.

En calculant la forme normale de Smith de la matrice de relations, nous obtenons un algorithme pour le cardinal et la structure (en tant que produit de groupes cycliques) de la jacobienne. Après optimisation des paramètres libres d et c, la complexité devient

$$L_{q^g}\left(1/3, \frac{4}{3}\sqrt{a_0 c + b_0} + o(1)\right)$$

avec c la solution positive de l'équation quadratique $c^2 - \frac{4}{9}a_0 c - \frac{4}{9}b_0 = 0$.

Il reste à voir comment calculer des logarithmes discrets. Pour cela, il faut (comme dans la troisième phase de l'algorithme 4.2) obtenir une relation supplémentaire qui fait intervenir l'élément Q dont le logarithme est cherché. Mais on n'a aucun contrôle sur la taille de Q, de $\log L(1)$ plutôt que de $\log L(2/3)$. En perturbant aléatoirement Q par des éléments de la base des facteurs, on arrive par le théorème de friabilité 4.3 en temps $L(1/3)$ a une relation contenant des premiers Q_i d'une taille en $\log L(2/3)$. On peut continuer de façon semblable à l'approche baptisée « descente par rapport à Q_i » pour la factorisation, c'est-à-dire créer pour chaque Q_i une relation qui le contient en considérant les fonctions $w = r(X) + s(X)Y$ qui passent par Q_i. Mais pour avoir une chance de trouver une relation, il faut comme avant laisser un peu de marge pour le degré de r et s ; avec la contrainte de passer par Q_i, on arrive encore une fois à un diviseur de degré en $L(1)$, et le processus tourne en rond.

La solution retenue dans [61] est de relâcher un peu la contrainte sur la complexité. Soit donc $\varepsilon > 0$ fixé. En un temps de $L(1/3 + \varepsilon)$, on peut créer une relation contenant Q et d'autres diviseurs premiers Q_i de taille $\log L(2/3 - \varepsilon)$. Pour chaque Q_i, on fait une descente par des fonctions passant par Q_i ; cela permet de le remplacer par une combinaison linéaire de diviseurs premiers $Q_{i,j}$ de taille $\log L(2/3 - 2\varepsilon)$, et ainsi de suite. Quand le degré d'un $Q_{i,j,\ldots}$ passe sous la barre de $\log L(1/3 + \varepsilon)$, la descente renvoie des premiers de taille $\log L(1/3)$, qui se trouvent donc dans la base des facteurs.

Ce processus de descente crée un arbre de degré en $O(g)$ et de hauteur bornée par $1/(3\varepsilon)$ dont les feuilles sont dans la base des facteurs ; comme ε est supposé fixe, le nombre de nœuds est polynomial en g et donc largement couvert par toute fonction sous-exponentielle. Ainsi, nous démontrons le résultat suivant :

Théorème 4.4 (heuristique) *Soit donnée une famille de courbes $\mathcal{C}$ comme ci-dessus, satisfaisant en particulier (4.2) et $g \geqslant (\log q)^\delta$ pour un $\delta > 2$, et soit $\varepsilon > 0$. En supposant heuristiquement que les diviseurs rencontrés au cours de l'algorithme ont la même probabilité d'être friables que des diviseurs aléatoires du même degré, des logarithmes discrets dans la jacobienne de $\mathcal{C}$ se calculent en temps $L_{q^g}(1/3 + \varepsilon, o(1))$.*

En ce qui concerne la constante de la fonction sous-exponentielle, il suffit de remarquer que l'existence d'un algorithme en $L(1/3+\varepsilon/2, c)$ pour n'importe quelle constante c permet de passer à $L(1/3 + \varepsilon, o(1))$ par (4.1).

Il est possible d'équilibrer différemment les degrés a et b en X et Y de la courbe. En posant $a \approx g^\alpha$ et $b \approx g^{1-\alpha}$ pour α entre $1/3$ et $1/2$, l'algorithme pour calculer la structure du groupe reste en $L(1/3)$ (avec une constante différente), tandis que le temps pour calculer des logarithmes croît vers $L(\alpha + \varepsilon)$. Quand α devient plus petit que $1/3$,

le calcul de la structure de groupe ne se fait plus en temps $L(1/3)$; il passe à $L(x(\alpha))$ avec $x(\alpha) \in [1/3, 1/2]$, et redevient $L(1/2)$ pour les courbes hyperelliptiques. Notons que c'est apparemment la première fois que des algorithmes sous-exponentiels apparaissent naturellement avec un premier paramètre différent de $1/2$ et de $1/3$.

Notre analyse sous-exponentielle jette aussi une nouvelle lumière sur le résultat de Diem dans [47], qui a montré, encore une fois en considérant le cas d'un petit genre fixé, que les courbes ni elliptiques ni hyperelliptiques nécessitent à sécurité égale des tailles de groupes plus élevées et sont à éviter en cryptographie.

4.4 Perspectives

Notre premier algorithme de [61] pour calculer des logarithmes discrets en temps $L(1/3+\varepsilon)$ dans les jacobiennes d'une certaine classe de courbes ouvre une toute nouvelle direction de recherche. Nous avons par exemple réussi à éliminer le ε en modifiant le processus de descente, un résultat qui sera soumis sur invitation au Journal of Cryptology. Au colloque « 10th Workshop on Elliptic Curve Cryptography (ECC 2006) » Diem a annoncé un algorithme en $L(1/3)$ inspiré par nos idées, mais avec un point de vue assez différent [46] ; pour l'instant, il n'est pas clair si sa classe de courbes diffère de la nôtre. Évidemment, un problème ouvert est de classifier toutes les courbes auxquelles s'appliquent ces algorithmes.

Concernant une implantation, je viens d'encadrer le stage d'option scientifique de l'École polytechnique (première année de master) de Guillaume Guerpillon sur les formes normales d'Hermite et de Smith, nécessaires pour déterminer la structure de groupe [100] ; et le stage de master de Jean-François Biasse sur les différentes approches à la recherche de relations [16].

Notre algorithme en $L(1/3)$ s'inspire du crible des corps de fonctions pour attaquer directement des courbes (c'est-à-dire, des corps de fonctions). Dans un esprit semblable, il devrait être possible de s'inspirer du crible des corps de nombres pour calculer le groupe de classes et les unités fondamentales ou au moins le régulateur de corps de nombres.

Ainsi que les algorithmes de logarithme discret en $L(1/2)$ permettent, par la descente de Weil (voir le dernier paragraphe de la section 4.1.2), d'attaquer certains cryptosystèmes elliptiques, il se pose la question si les algorithmes en $L(1/3)$ permettent d'aller plus loin et d'étendre la portée des attaques par descente de Weil à d'autres courbes elliptiques ou même hyperelliptiques.

S'il paraît difficile de dépasser la barrière de $L(1/3)$, l'apparition d'encore un algorithme avec cette complexité, dans le contexte des courbes, différent de celui des corps finis, laisse espérer pouvoir comprendre si un algorithme en, disons, $L(1/4)$ serait envisageable ou quelle est l'obstruction fondamentale à son existence.

Faisant le lien entre les chapitres 4 et 1, il se pose toujours la question si les courbes elliptiques obtenues avec les méthodes de la multiplication complexe (c'est-à-dire avec un discriminant du corps quadratique associé particulièrement petit, et donc une structure d'endomorphismes particulièrement riche), admettent un meilleur algorithme pour calculer des logarithmes discrets que des courbes elliptiques aléatoires.

Bibliographie

[1] Fatima K. Abu Salem et Kamal Khuri-Makdisi. Fast Jacobian group operations for $C_{3,4}$ curves over a large finite field. *LMS Journal of Computation and Mathematics*, 10:307–328, 2007.

[2] Leonard M. Adleman. The Function Field Sieve. Dans Leonard M. Adleman et Ming-Deh Huang, éditeurs, *Algorithmic Number Theory*, volume 877 de *Lecture Notes in Computer Science*, pages 108–121, Berlin, 1994. Springer-Verlag.

[3] Leonard M. Adleman, Jonathan DeMarrais et Ming-Deh Huang. A Subexponential Algorithm for Discrete Logarithms over the Rational Subgroup of the Jacobians of Large Genus Hyperelliptic Curves over Finite Fields. Dans Leonard M. Adleman et Ming-Deh Huang, éditeurs, *Algorithmic Number Theory*, volume 877 de *Lecture Notes in Computer Science*, pages 28–40, Berlin, 1994. Springer-Verlag.

[4] Manindra Agrawal, Neeraj Kayal et Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[5] Sattam S. Al-Riyami et Kenneth G. Paterson. Certificateless Public Key Cryptography. Dans Chi Sung Laih, éditeur, *Advances in Cryptology — ASIACRYPT 2003*, volume 2894 de *Lecture Notes in Computer Science*, pages 452–473, Berlin, 2003. Springer-Verlag.

[6] Seigo Arita. Algorithms for Computations in Jacobian Group of C_{ab} Curve and Their Application to Discrete-Log Based Public Key Cryptosystems. *IEICE Transactions*, J82-A(8):1291–1299, 1999. En Japonais. Traduction anglaise dans les actes de *Conference on The Mathematics of Public Key Cryptography*, Toronto 1999, ou [7].

[7] Seigo Arita. An Addition Algorithm in Jacobian of C_{ab} Curves. *Discrete Applied Mathematics*, 130(1):13–31, 2003.

[8] Emil Artin. Beweis des allgemeinen Reziprozitätsgesetzes. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 5:353–363, 1927.

[9] A. O. L. Atkin et F. Morain. Elliptic Curves and Primality Proving. *Mathematics of Computation*, 61(203):29–68, 1993.

[10] Eric Bach. Explicit Bounds for Primality Testing and Related Problems. *Mathematics of Computation*, 55(191):355–380, 1990.

[11] Paolo Barreto. The Pairing-Based Crypto Lounge, 2005. `http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html`.

[12] Paulo S. L. M. Barreto, Ben Lynn et Michael Scott. Constructing Elliptic Curves with Prescribed Embedding Degrees. Dans Stelvio Cimato, Clemente Galdi et Giuseppe Persiano, éditeurs, *Security in Communication Networks — Third International Conference, SCN 2002, Amalfi, Italy, September 2002*, volume 2576 de *Lecture Notes in Computer Science*, pages 257–267, Berlin, 2003. Springer-Verlag.

[13] Abdolali Basiri, Andreas Enge, Jean-Charles Faugère et Nicolas Gürel. Implementing the Arithmetic of $C_{3,4}$ Curves. Dans Duncan Buell, éditeur, *Algorithmic Number Theory — ANTS-VI*, volume 3076 de *Lecture Notes in Computer Science*, pages 87–101, Berlin, 2004. Springer-Verlag.

[14] Abdolali Basiri, Andreas Enge, Jean-Charles Faugère et Nicolas Gürel. The Arithmetic of Jacobian Groups of Superelliptic Cubics. *Mathematics of Computation*, 74(249):389–410, 2005.

[15] Renet Lovorn Bender et Carl Pomerance. Rigorous Discrete Logarithm Computations in Finite Fields Via Smooth Polynomials. Dans D. A. Buell et J. T. Teitelbaum, éditeurs, *Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A.O.L. Atkin*, volume 7 de *Studies in Advanced Mathematics*, pages 221–232. American Mathematical Society, 1998.

[16] Jean-François Biasse. Algorithmes sous-exponentiels de résolution du logarithme discret sur les jacobiennes de courbes algébriques. Mémoire de master, Master Parisien de Recherche en Informatique, Paris, 2007.

[17] B. J. Birch. Weber's Class Invariants. *Mathematika*, 16:283–294, 1969.

[18] Ian Blake, Gadiel Seroussy et Nigel Smart. *Elliptic Curves in Cryptography*, volume 265 de *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1999.

[19] Dan Boneh et Matt Franklin. Identity-Based Encryption from the Weil Pairing. Dans Joe Kilian, éditeur, *Advances in Cryptology — CRYPTO 2001*, volume 2139 de *Lecture Notes in Computer Science*, pages 213–229, Berlin, 2001. Springer-Verlag.

[20] Dan Boneh, Ben Lynn et Hovav Shacham. Short Signatures from the Weil Pairing. Dans Colin Boyd, éditeur, *Advances in Cryptology — ASIACRYPT 2001*, volume 2248 de *Lecture Notes in Computer Science*, pages 514–532, Berlin, 2001. Springer-Verlag.

[21] Jonathan M. Borwein et Peter B. Borwein. *Pi and the AGM*. Wiley, New York, 1987.

[22] A. Bostan, F. Morain, B. Salvy et É. Schost. Fast algorithms for computing isogenies between elliptic curves. HAL-INRIA 91441, INRIA, 2006. À paraître dans *Mathematics of Computation*, `http://hal.inria.fr/inria-00091441`.

[23] Richard P. Brent. Fast Multiple-Precision Evaluation of Elementary Functions. *Journal of the ACM*, 23(2):242–251, 1976.

[24] Friederike Brezing et Annegret Weng. Elliptic Curves Suitable for Pairing Based Cryptography. *Designs, Codes and Cryptography*, 37(1):133–141, 2005.

[25] N. Brisebarre et G. Philibert. Effective lower and upper bounds for the Fourier coefficients of powers of the modular invariant j. *Journal of the Ramanujan Mathematical Society*, 20:255–282, 2005.

[26] Reinier Bröker. *Constructing elliptic curves of prescribed order*. Proefschrift, Universiteit Leiden, 2006.

[27] N. G. de Bruijn. On the number of positive integers $\leqslant x$ and free of prime factors $> y$. *Indagationes Mathematicae*, 13:50–60, 1951.

[28] Ran Canetti, Oded Goldreich et Shai Halevi. The Random Oracle Methodology, Revisited (preliminary version). Dans *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 209–218. Association for Computing Machinery, 1998.

[29] David G. Cantor. Computing in the Jacobian of a Hyperelliptic Curve. *Mathematics of Computation*, 48(177):95–101, 1987.

[30] M. Cipolla. Un metodo per la risoluzione della congruenza di secondo grado. *Napoli Rend.*, 9:153–163, 1903.

[31] Clifford Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. Dans Bahram Honary, éditeur, *Cryptography and Coding: 8th IMA International Conference Cirencester, UK, December 17–19, 2001*, volume 2260 de *Lecture Notes in Computer Science*, pages 360–363, Berlin, 2001. Springer-Verlag.

[32] H. Cohen et H. W. Lenstra Jr.. Heuristics on class groups of number fields. Dans H. Jager, éditeur, *Number Theory Noordwijkerhout 1983*, volume 1068 de *Lecture Notes in Mathematics*, pages 33–62, Berlin, 1984. Springer-Verlag.

[33] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen et Frederik Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Discrete mathematics and its applications. Chapman & Hall, Boca Raton, 2006.

[34] Paula Cohen. On the coefficients of the transformation polynomials for the elliptic modular function. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95:389–402, 1984.

[35] Don Coppersmith. Fast Evaluation of Logarithms in Fields of Characteristic Two. *IEEE Transactions on Information Theory*, 30(4):587–594, juillet 1984.

[36] Jean-Marc Couveignes. Computing l-Isogenies Using the p-Torsion. Dans Henri Cohen, éditeur, *Algorithmic Number Theory — ANTS-II*, volume 1122 de *Lecture Notes in Computer Science*, pages 59–65, Berlin, 1996. Springer-Verlag.

[37] Jean-Marc Couveignes. Algebraic Groups and Discrete Logarithm. Dans K. Alster, J. Urbanowicz et H. C. Williams, éditeurs, *Public-Key Cryptography and Computational Number Theory*, pages 17–27, Berlin, 2001. De Gruyter.

[38] Jean-Marc Couveignes et Thierry Henocq. Action of Modular Correspondences around CM Points. Dans Claus Fieker et David R. Kohel, éditeurs, *Algorithmic Number Theory — ANTS-V*, volume 2369 de *Lecture Notes in Computer Science*, pages 234–243, Berlin, 2002. Springer-Verlag.

[39] Jean-Marc Couveignes et François Morain. Schoof's Algorithm and Isogeny Cycles. Dans Leonard M. Adleman et Ming-Deh Huang, éditeurs, *Algorithmic Number Theory*, volume 877 de *Lecture Notes in Computer Science*, pages 43–58, Berlin, 1994. Springer-Verlag.

[40] David A. Cox. *Primes of the Form $x^2 + ny^2$ — Fermat, Class Field Theory, and Complex Multiplication.* John Wiley & Sons, New York, 1989.

[41] Max Deuring. Arithmetische Theorie der Korrespondenzen algebraischer Funktionenkörper II. *Journal für die reine und angewandte Mathematik*, 183:25–36, 1941.

[42] Max Deuring. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abhandlungen aus dem mathematischen Seminar der hamburgischen Universität*, 14:197–272, 1941.

[43] Max Deuring. Die Klassenkörper der komplexen Multiplikation. Dans *Enzyklop. d. math. Wissenschaften*, volume I 2 Heft 10. Teubner, Stuttgart, 2e édition, 1958.

[44] Diderot et d'Alembert, éditeurs. *Encyclopédie, ou dictionnaire raisonné des sciences, des arts et des métiers, par une société de gens de lettres.* Briasson, David, Le Breton et Durand, 1751–1772.

[45] Claus Diem. The GHS Attack in Odd Characteristic. *Journal of the Ramanujan Mathematical Society*, 18(1):1–32, 2003.

[46] Claus Diem. An index calculus algorithm for non-singular plane curves of high genus, 2006. Transparents de la conférence au 10th Workshop on Elliptic Curve Cryptography, Toronto, September 18–20, `http://www.cacr.math.uwaterloo.ca/conferences/2006/ecc2006/diem.pdf`.

[47] Claus Diem. An Index Calculus Algorithm for Plane Curves of Small Degree. Dans Florian Hess, Sebastian Pauli et Michael Pohst, éditeurs, *Algorithmic Number Theory — ANTS-VII*, volume 4076 de *Lecture Notes in Computer Science*, pages 543–557, Berlin, 2006. Springer-Verlag.

[48] Régis Dupont. *Moyenne arithmético-géométrique, suites de Borchardt et applications.* Thèse de doctorat, École polytechnique, Palaiseau, 2006.

[49] Régis Dupont. Fast evaluation of modular functions using Newton iterations and the AGM. À paraître dans *Mathematics of Computation*, `http://www.lix.polytechnique.fr/Labo/Regis.Dupont/preprints/Dupont/FastEvalMod.ps.gz`, 2007.

[50] Régis Dupont, Andreas Enge et François Morain. Building curves with arbitrary small MOV degree over finite prime fields. *Journal of Cryptology*, 18(2):79–89, 2005.

[51] Régis Dupont et Andreas Enge. Provably Secure Non-Interactive Key Distribution Based on Pairings. Dans Daniel Augot, Pascale Charpin et Grigory Kabatianski, éditeurs, *WCC 2003 — Proceedings of the International Workshop on Coding and Cryptography*, pages 165–174. École Supérieure et d'Application des Transmissions, 2003.

[52] Régis Dupont et Andreas Enge. Provably Secure Non-Interactive Key Distribution Based on Pairings. *Discrete Applied Mathematics*, 154(2):270–276, 2006.

[53] Noam D. Elkies. Elliptic and Modular Curves over Finite Fields and Related Computational Issues. Dans D. A. Buell et J. T. Teitelbaum, éditeurs, *Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A.O.L. Atkin*, volume 7 de *Studies in Advanced Mathematics*, pages 21–76. American Mathematical Society, 1998.

[54] Andreas Enge. *Hyperelliptic Cryptosystems: Efficiency and Subexponential Attacks*. Books on Demand, Norderstedt, 2000.

[55] Andreas Enge. The Extended Euclidian Algorithm on Polynomials, and the Computational Efficiency of Hyperelliptic Cryptosystems. *Designs, Codes and Cryptography*, 23(1):53–74, 2001.

[56] Andreas Enge. A General Framework for Subexponential Discrete Logarithm Algorithms in Groups of Unknown Order. Dans A. Blokhuis, J. W. P. Hirschfeld, D. Jungnickel et J. A. Thas, éditeurs, *Finite Geometries*, volume 3 de *Developments in Mathematics*, pages 133–146, Dordrecht, 2001. Kluwer Academic Publishers.

[57] Andreas Enge. Computing Discrete Logarithms in High-Genus Hyperelliptic Jacobians in Provably Subexponential Time. *Mathematics of Computation*, 71(238):729–742, 2002.

[58] Andreas Enge. The complexity of class polynomial computation via floating point approximations. HAL-INRIA 1040 et ArXiv cs.CC/0601104, INRIA, 2006. `http://hal.inria.fr/inria-00001040`.

[59] Andreas Enge. Computing modular polynomials in quasi-linear time. HAL-INRIA 143084 et ArXiv 0704.3177, INRIA, 2007. `http://hal.inria.fr/inria-00143084`.

[60] Andreas Enge et Pierrick Gaudry. A General Framework for Subexponential Discrete Logarithm Algorithms. *Acta Arithmetica*, 102(1):83–103, 2002.

[61] Andreas Enge et Pierrick Gaudry. An $L(1/3 + \varepsilon)$ algorithm for the discrete logarithm problem for low degree curves. Dans Moni Naor, éditeur, *Advances in Cryptology — Eurocrypt 2007*, volume 4515 de *Lecture Notes in Computer Science*, pages 367–382, Berlin, 2007. Springer-Verlag.

[62] Andreas Enge, Pierrick Gaudry et François Morain. Computing #E(GF(10^2004+4863)), décembre 2005. Communication sur la Number Theory List, `http://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind0512&L=nmbrthry&T=0&F=&S=&P=3123`.

[63] Andreas Enge, Pierrick Gaudry et François Morain. Computing #E(GF(p)) for large prime p – an update, mars 2005. Communication sur la Number Theory List, `http://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind0503&L=nmbrthry&T=0&P=324`.

[64] Andreas Enge et François Morain. Comparing Invariants for Class Fields of Imaginary Quadratic Fields. Dans Claus Fieker et David R. Kohel, éditeurs, *Algorithmic Number Theory — ANTS-V*, volume 2369 de *Lecture Notes in Computer Science*, pages 252–266, Berlin, 2002. Springer-Verlag.

[65] Andreas Enge et François Morain. Fast Decomposition of Polynomials with Known Galois Group. Dans Marc Fossorier, Tom Høholdt et Alain Poli, éditeurs, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes — AAECC-15*, volume 2643 de *Lecture Notes in Computer Science*, pages 254–264, Berlin, 2003. Springer-Verlag.

[66] Andreas Enge et François Morain. SEA in genus 1: 2500 decimal digits, décembre 2006. Communication sur la Number Theory List, `http://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind0612&L=NMBRTHRY&P=R125&I=-3`.

[67] Andreas Enge, Michael Pohst et Reinhard Schertz. Verfahren zur Konstruktion elliptischer Kurven über endlichen Körpern, 2005. Patentschrift DE 103 29 885 B4 2005.10.06.

[68] Andreas Enge et Reinhard Schertz. Constructing elliptic curves over finite fields using double eta-quotients. *Journal de Théorie des Nombres de Bordeaux*, 16:555–568, 2004.

[69] Andreas Enge et Reinhard Schertz. Modular Curves of Composite Level. *Acta Arithmetica*, 118(2):129–141, 2005.

[70] Andreas Enge et Andreas Stein. Smooth Ideals in Hyperelliptic Function Fields. *Mathematics of Computation*, 71(239):1219–1230, 2002.

[71] Andreas Enge et Paul Zimmermann. `mpc` — A library for multiprecision complex arithmetic with exact rounding. Version 0.4.5, `http://www.lix.polytechnique.fr/Labo/Andreas.Enge/Software.html`.

[72] Leonhard Euler. Evolutio Producti Infiniti $(1-x)(1-xx)(1-x^3)(1-x^4)(1-x^5)(1-x^6)$ etc. in Seriem Simplicem. *Acta academiae scientiarum Petropolitanae*, 1780:I:125–169, 1783. *Opera Omnia* I.3:472–479.

[73] Stéphane Flon et Roger Oyono. Fast Arithmetic on Jacobians of Picard Curves. Dans Feng Bao, Robert Deng et Jianying Zhou, éditeurs, *Public Key Cryptography — PKC 2004*, volume 2947 de *Lecture Notes in Computer Science*, pages 55–68, Berlin, 2004. Springer-Verlag.

[74] Mireille Fouquet, Pierrick Gaudry et Robert Harley. Finding Secure Curves with the Satoh-FGH Algorithm and an Early-Abort strategy. Dans Birgit Pfitzmann, éditeur, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 de *Lecture Notes in Computer Science*, pages 14–29, Berlin, 2001. Springer-Verlag.

[75] Mireille Fouquet et François Morain. Isogeny Volcanoes and the SEA Algorithm. Dans Claus Fieker et David R. Kohel, éditeurs, *Algorithmic Number Theory — ANTS-V*, volume 2369 de *Lecture Notes in Computer Science*, pages 276–291, Berlin, 2002. Springer-Verlag.

[76] David Freeman, Michael Scott et Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. Preprint, Cryptology ePrint Archive 2006/372, `http://eprint.iacr.org/2006/372/`, 2006.

[77] Gerhard Frey. Applications of Arithmetical Geometry to Cryptographic Constructions. Dans Dieter Jungnickel et Harald Niederreiter, éditeurs, *Finite Fields and Applications — Proceedings of The Fifth International Conference on Finite Fields and Applications F_q5, held at the University of Augsburg, Germany, August 2–6, 1999*, pages 128–161, Berlin, 2001. Springer-Verlag.

[78] Gerhard Frey et Hans-Georg Rück. A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. *Mathematics of Computation*, 62(206):865–874, avril 1994.

[79] Robert Fricke. *Lehrbuch der Algebra*, volume III — Algebraische Zahlen. Vieweg, Braunschweig, 1928.

[80] Philipp Furtwängler. Allgemeiner Existenzbeweis für den Klassenkörper eines beliebigen algebraischen Zahlkörpers. *Mathematische Annalen*, 63:1–37, 1907.

[81] S. Galbraith, F. Hess et F. Vercauteren. Aspects of Pairing Inversion. `http://eprint.iacr.org/2007/256/`, 2007.

[82] S. D. Galbraith, S. M. Paulus et N. P. Smart. Arithmetic on Superelliptic Curves. *Mathematics of Computation*, 71(237):393–405, 2002.

[83] Steven Galbraith. Pairings. Dans Ian F. Blake, Gadiel Seroussi et Nigel P. Smart, éditeurs, *Advances in Elliptic Curve Cryptography*, Chapitre 9, pages 183–213. Cambridge University Press, Cambridge, 2005.

[84] Steven D. Galbraith. *Equations for Modular Curves*. PhD thesis, University of Oxford, 1996.

[85] Joachim von zur Gathen et Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[86] P. Gaudry, F. Hess et N. P. Smart. Constructive and Destructive Facets of Weil Descent on Elliptic Curves. *Journal of Cryptology*, 15:19–46, 2002.

[87] P. Gaudry, T. Houtmann, D. Kohel, C. Ritzenthaler et A. Weng. The 2-Adic CM Method for Genus 2 Curves with Application to Cryptography. Dans Xuejia Lai et Kefei Chen, éditeurs, *Advances in Cryptology — ASIACRYPT 2006*, volume 4284 de *Lecture Notes in Computer Science*, pages 114–129, Berlin, 2006. Springer-Verlag.

[88] P. Gaudry et F. Morain. Fast algorithms for computing the eigenvalue in the Schoof–Elkies–Atkin algorithm. Dans Jean-Guillaume Dumas, éditeur, *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computations (sic!) — ISSAC MMVI*, pages 109–115, New York, 2006. ACM Press.

[89] P. Gaudry et É. Schost. Modular Equations for Hyperelliptic Curves. *Mathematics of Computation*, 74(249):429–454, 2005.

[90] P. Gaudry, E. Thomé, N. Thériault et C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Mathematics of Computation*, 76(257):475–492, 2007.

[91] Pierrick Gaudry. An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves. Dans Bart Preneel, éditeur, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 de *Lecture Notes in Computer Science*, pages 19–34, Berlin, 2000. Springer-Verlag.

[92] Pierrick Gaudry. Fast genus 2 arithmetic based on Theta functions. *Journal of Mathematical Cryptology*, 1(3):243–265, 2007.

[93] Pierrick Gaudry et Éric Schost. Construction of secure random curves of genus 2 over prime fields. Dans Duncan Buell, éditeur, *Algorithmic Number Theory — ANTS-VI*, volume 3076 de *Lecture Notes in Computer Science*, pages 239–256, Berlin, 2004. Springer-Verlag.

[94] Alice Gee. Class Invariants by Shimura's Reciprocity Law. *Journal de Théorie des Nombres de Bordeaux*, 11(1):45–72, 1999.

[95] Alice Gee et Peter Stevenhagen. Generating Class Fields Using Shimura Reciprocity. Dans J. P. Buhler, éditeur, *Algorithmic Number Theory — ANTS-III*, volume 1423 de *Lecture Notes in Computer Science*, pages 441–453, Berlin, 1998. Springer-Verlag.

[96] Shafi Goldwasser et Joe Kilian. Almost All Primes Can be Quickly Certified. Dans *Proc. 18th Annual ACM Symp. on Theory of Computing*, pages 316–329, 1986.

[97] Daniel M. Gordon. Discrete Logarithms in $GF(p)$ using the number field sieve. *SIAM Journal on Discrete Mathematics*, 6(1):124–138, 1993.

[98] De la Grange. Recherches d'arithmétique. *Nouveaux Mémoires de l'Académie Royale des Sciences et Belles-Lettres*, pages 265–312, 1773.

[99] Torbjörn Granlund *et al.* `gmp` — GNU multiprecision library. Version 4.2.1, `http://www.swox.com/gmp`.

[100] Guillaume Guerpillon. Calcul du groupe de classes dans un corps quadratique imaginaire. Rapport de stage d'option scientifique, École polytechnique, Palaiseau, 2007.

[101] Louis Claude Guillou et Jean-Jacques Quisquater. A "Paradoxical" Identity-Based Signature Scheme Resulting from Zero-Knowledge. Dans Shafi Goldwasser, éditeur, *Advances in Cryptology — CRYPTO '88*, volume 403 de *Lecture Notes in Computer Science*, pages 216–231, Berlin, 1990. Springer-Verlag.

[102] Peter Gutmann. X.509 Style Guide, octobre 2000. `http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt`.

[103] James L. Hafner et Kevin S. McCurley. A Rigorous Subexponential Algorithm for Computation of Class Groups. *Journal of the American Mathematical Society*, 2(4):837–850, 1989.

[104] G. Hanrot et F. Morain. Solvability by Radicals from an Algorithmic Point of View. Dans Bernard Mourrain, éditeur, *ISSAC 2001 — Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*, pages 175–182, New York, 2001. Association for Computing Machinery.

[105] Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier et Paul Zimmermann *et al*. `mpfr` — A library for multiple-precision floating-point computations with exact rounding. Version 2.2.1, `http://www.mpfr.org`.

[106] William B. Hart. *Evaluation of the Dedekind Eta Funktion*. PhD thesis, Macquarie University, 2004.

[107] Helmut Hasse. Beweis des Analogons der Riemannschen Vermutung für die Artinschen und F. K. Schmidtschen Kongruenzzetafunktionen in gewissen elliptischen Fällen. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Phyikalische Klasse*, pages 253–262, 1933.

[108] Helmut Hasse. Zur Theorie der abstrakten elliptischen Funktionenkörper III. Die Struktur des Meromorphismenrings. Die Riemannsche Vermutung. *Journal für die reine und angewandte Mathematik*, 175:193–208, 1936.

[109] Erich Hecke. *Vorlesungen über die Theorie der algebraischen Zahlen*. Akademische Verlagsgesellschaft, Leipzig, 1923. `http://dml.math.uni-bielefeld.de/dml/misc/BOOKS/hecke.ocr.djvu`.

[110] Florian Hess. Computing Riemann–Roch Spaces in Algebraic Function Fields and Related Topics. *Journal of Symbolic Computation*, 33(4):425–445, 2002.

[111] Florian Hess. Computing Relations in Divisor Class Groups of Algebraic Curves over Finite Fields. Manuscrit, `http://www.math.tu-berlin.de/~hess/personal/dlog.ps.gz`, 2004.

[112] Florian Hess. Weil Descent Attacks. Dans Ian F. Blake, Gadiel Seroussi et Nigel P. Smart, éditeurs, *Advances in Elliptic Curve Cryptography*, Chapitre 8, pages 151–180. Cambridge University Press, Cambridge, 2005.

[113] Marc Hindry et Joseph H. Silverman. *Diophantine Geometry — An Introduction*. Springer-Verlag, New York, 2000.

[114] R. Housley, W. Ford, W. Polk et D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459, Internet Engineering Task Force, 1999. `http://www.ietf.org/rfc/rfc2459.txt`.

[115] Carl Gustav Jacob Jacobi. Fundamenta Nova Theoriae Functionum Ellipticarum. Dans *Gesammelte Werke*, pages 49–239. Chelsea, New York, 2 (1969) édition, 1829.

[116] Michael J. Jacobson Jr.. Applying Sieving to the Computation of Quadratic Class Groups. *Mathematics of Computation*, 68(226):859–867, 1999.

[117] Michael J. Jacobson Jr., Shantha Ramachandran et Hugh C. Williams. Numerical Results on Class Groups of Imaginary Quadratic Fields. Dans Florian Hess, Sebastian Pauli et Michael Pohst, éditeurs, *Algorithmic Number Theory — ANTS-VII*, volume 4076 de *Lecture Notes in Computer Science*, pages 87–101, Berlin, 2006. Springer-Verlag.

[118] Antoine Joux. A One Round Protocol for Tripartite Diffie–Hellman. Dans Wieb Bosma, éditeur, *Algorithmic Number Theory — ANTS-IV*, volume 1838 de *Lecture Notes in Computer Science*, pages 385–393, Berlin, 2000. Springer-Verlag.

[119] Antoine Joux, Reynald Lercier, Nigel Smart et Frederik Vercauteren. The Number Field Sieve in the Medium Prime Case. Dans Cynthia Dwork, éditeur, *Advances in Cryptology — CRYPTO 2006*, volume 4117 de *Lecture Notes in Computer Science*, pages 326–344, Berlin, 2006. Springer-Verlag.

[120] Kamal Khuri-Makdisi. Linear Algebra Algorithms for Divisors on an Algebraic Curve. *Mathematics of Computation*, 73(245):333–357, 2004.

[121] Felix Klein. Über die Transformationsgleichung der elliptischen Funktionen und die Auflösung der Gleichungen fünften Grades. *Math. Annalen*, 14:111–172, 1878. *Gesammelte Mathematische Abhandlungen* III:13–75.

[122] John Knopfmacher. *Abstract Analytic Number Theory*, volume 12 de *North-Holland Mathematical Library*. North-Holland Publishing Company, Amsterdam, 1975.

[123] David Kohel. *Endomorphism Rings of Elliptic Curves over Finite Fields*. PhD thesis, University of California at Berkeley, 1996.

[124] M. Kraïtchik. *Théorie des nombres*. Gauthier-Villars, Paris, 1922.

[125] Cornelius Lanczos. Solution of Systems of Linear Equations by Mimimized Iterations. *Journal of Research of the National Bureau of Standards*, 49(1):33–53, juillet 1952.

[126] Tanja Lange. Formulae for Arithmetic on Genus 2 Hyperelliptic Curves. *Applicable Algebra in Engineering, Communication and Computing*, 15(5):295–328, 2005.

[127] R. Lercier et E. Riboulet-Deyris. Elliptic curves with complex multiplication. Communication sur la Number Theory List, `http://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind0401&L=nmbrthry&P=R305`, 2004.

[128] Reynald Lercier. Computing Isogenies in $GF(2^n)$. Dans Henri Cohen, éditeur, *Algorithmic Number Theory — ANTS-II*, volume 1122 de *Lecture Notes in Computer Science*, pages 197–212, Berlin, 1996. Springer-Verlag.

[129] Benoît Libert. *New Secure Applications of Bilinear Maps in Cryptography*. Thèse de doctorat, Université catholique de Louvain, Louvain-la-Neuve, 2006.

[130] Stéphane Louboutin. Computation of Class Numbers of Quadratic Number Fields. *Mathematics of Computation*, 71(240):1735–1743, 2002.

[131] Florian Luca et Igor E. Shparlinski. Elliptic Curves with Low Embedding Degree. *Journal of Cryptology*, 19:553–562, 2006.

[132] E. Manstavičius. Remarks on the Semigroup Elements Free of Large Prime Factors. *Lithuanian Mathematical Journal*, 32(4):400–409, 1992.

[133] E. Manstavičius. Semigroup Elements Free of Large Prime Factors. Dans F. Schweiger et E. Manstavičius, éditeurs, *New Trends in Probability and Statistic*, pages 135–153, 1992.

[134] Ueli M. Maurer et Stefan Wolf. The Relationship between Breaking the Diffie–Hellman Protocol and Computing Discrete Logarithms. *SIAM Journal on Computing*, 28(5):1689–1721, 1999.

[135] Ueli M. Maurer et Yacov Yacobi. A Non-Interactive Public-Key Distribution System. *Designs, Codes and Cryptography*, 9(3):305–316, 1996.

[136] Alfred J. Menezes, Tatsuaki Okamoto et Scott A. Vanstone. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, septembre 1993.

[137] Jean-François Mestre. Lettre adressée à Gaudry et Harley, décembre 2000. `http://www.institut.math.jussieu.fr/~mestre/lettreGaudryHarley.ps`.

[138] C. Meyer. Bemerkungen zum Satz von Heegner–Stark über die imaginärquadratischen Zahlkörper mit der Klassenzahl Eins. *Journal für die reine und angewandte Mathematik*, 242:179–214, 1970.

[139] P. Mihăilescu, F. Morain et É. Schost. Computing the Eigenvalue in the Schoof–Elkies–Atkin Algorithm using Abelian Lifts. Dans C. W. Brown, éditeur, *Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation — ISSAC 2007*, pages 285–292, New York, 2007. Association for Computing Machinery.

[140] Atsuko Miyaji, Masaki Nakabayashi et Shunzou Takano. New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. *IEICE Trans. Fundamentals*, E84-A(5):1234–1243, mai 2001.

[141] François Morain. Calcul du nombre de points sur une courbe elliptique dans un corps fini: aspects algorithmiques. *Journal de Théorie des Nombres de Bordeaux*, 7(1):111–138, 1995.

[142] François Morain. ECPP, 2001. Version 6.4.5, `http://www.lix.polytechnique.fr/~morain/Prgms/getecpp645.english.html`.

[143] François Morain. La primalité en temps polynomial. *Astérisque*, 294:205–230, 2004.

[144] François Morain. La barre des 20000 chiffres est franchie, juin 2006. `http://www.lix.polytechnique.fr/~morain/Primes/mills2.txt`.

[145] Volker Müller. *Ein Algorithmus zur Bestimmung der Punktanzahl elliptischer Kurven über endlichen Körpern der Charakteristik größer drei.* Dissertation, Universität des Saarlandes, Saarbrücken, 1995. `ftp://ftp.informatik.tu-darmstadt.de/pub/TI/reports/vmueller.diss.ps.gz`.

[146] Volker Müller, Andreas Stein et Christoph Thiel. Computing Discrete Logarithms in Real Quadratic Congruence Function Fields of Large Genus. *Mathematics of Computation*, 68(226):807–822, 1999.

[147] Naoki Murabayashi. On Normal Forms of Modular Curves of Genus 2. *Osaka Journal of Mathematics*, 29:405–418, 1992.

[148] V. I. Nechaev. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

[149] Henri J. Nussbaumer. *Fast Fourier transform and convolution algorithms*. Springer, Berlin, 1981.

[150] Andrew P. Ogg. Hyperelliptic Modular Curves. *Bulletin de la Société Mathématique de France*, 102:449–462, 1974.

[151] P. C. van Oorschot et M. J. Wiener. Parallel Collision Search With Cryptanalytic Applications. *Journal of Cryptology*, 12(1):1–28, 1999.

[152] D. Page, N. P. Smart et F. Vercauteren. A comparison of MNT curves and supersingular curves. *Applicable Algebra in Engineering, Communication and Computing*, 17:379–392, 2006.

[153] Stephen C. Pohlig et Martin E. Hellman. An Improved Algorithm for Computing Logarithms over $GF(p)$ and Its Cryptographic Significance. *IEEE Transactions on Information Theory*, 24(1):106–110, janvier 1978.

[154] J. M. Pollard. Monte Carlo Methods for Index Computation (mod p). *Mathematics of Computation*, 32(143):918–924, juillet 1978.

[155] Carl Pomerance. Fast, Rigorous Factorization and Discrete Logarithm Algorithms. Dans David S. Johnson, Takao Nishizeki, Akihiro Nozaki et Herbert S. Wolf, éditeurs, *Discrete Algorithms and Complexity, Proceedings of the Japan–US Joint Seminar, June 4–6, 1986, Kyoto, Japan*, volume 15 de *Perspectives in Computing*, pages 119–143, Orlando, 1987. Academic Press.

[156] Regulierungsbehörde für Telekommunikation und Post. Geeignete Kryptoalgorithmen gemäß § 17 (2) SigV, novembre 1999. `http://www.bundesnetzagentur.de/media/archive/1501.pdf`.

[157] Josep Gonzàlez Rovira. Equations of Hyperelliptic Modular Curves. *Ann. Inst. Fourier Grenoble*, 41(4):779–795, 1991.

[158] R. Sakai, K. Ohgishi et M. Kasahara. Cryptosystems based on pairing, 2000. SCIS 2000, The 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, January 26–28.

[159] Takakazu Satoh. The Canonical Lift of an Ordinary Elliptic Curve over a Finite Field and its Point Counting. *Journal of the Ramanujan Mathematical Society*, 15:247–270, 2000.

[160] Takakazu Satoh et Kiyomichi Araki. Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves. *Commentarii Mathematici Universitatis Sancti Pauli*, 47(1):81–92, 1998. Errata in vol. 48 (2):211–213, 1999.

[161] Takakazu Satoh, Berit Skjernaa et Yuichiro Taguchi. Fast computation of canonical lifts of elliptic curves and its application to point counting. *Finite Fields and Their Applications*, 9(1):89–101, 2003.

[162] Reinhard Schertz. Die singulären Werte der Weberschen Funktionen $\mathfrak{f}$, $\mathfrak{f}_1$, $\mathfrak{f}_2$, γ_2, γ_3. *Journal für die reine und angewandte Mathematik*, 286/287:46–74, 1976.

[163] Reinhard Schertz. Zur expliziten Berechnung von Ganzheitsbasen in Strahlklassenkörpern über einem imaginär-quadratischen Zahlkörper. *Journal of Number Theory*, 34(1):41–53, janvier 1990.

[164] Reinhard Schertz. Weber's Class Invariants Revisited. *Journal de Théorie des Nombres de Bordeaux*, 14(1):325–343, 2002.

[165] L. Schläfli. Beweis der Hermiteschen Verwandlungstafeln für die elliptischen Modulfunktionen. *Journal für die reine und angewandte Mathematik*, 72:360–369, 1870.

[166] René Schoof. Elliptic Curves Over Finite Fields and the Computation of Square Roots mod p. *Mathematics of Computation*, 44(170):483–494, avril 1985.

[167] René Schoof. Counting Points on Elliptic Curves Over Finite Fields. *Journal de Théorie des Nombres de Bordeaux*, 7(1):219–254, 1995.

[168] René Schoof. The exponents of the groups of points on the reductions of an elliptic curve. Dans G. van der Geer, F. Oort et J. Steenbrink, éditeurs, *Arithmetic Algebraic Geometry*, pages 325–335, Boston, 1991. Birkhäuser.

[169] I. A. Semaev. Evaluation of Discrete Logarithms in a Group of p-Torsion Points of an Elliptic Curve in Characteristic p. *Mathematics of Computation*, 67(221):353–356, 1998.

[170] Martin Seysen. A Probabilistic Factorization Algorithm with Quadratic Forms of Negative Discriminant. *Mathematics of Computation*, 48(178):757–780, 1987.

[171] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. Dans G. R. Blakley et David Chaum, éditeurs, *Advances in Cryptology – CRYPTO '84*, volume 196 de *Lecture Notes in Computer Science*, pages 47–53, Berlin, 1985. Springer-Verlag.

[172] Daniel Shanks. The Infrastructure of a Real Quadratic Number Field and its Applications. Dans *Proc. 1972 Number Th. Conf.*, pages 217–224, Boulder (Colorado), 1972.

[173] Goro Shimura. *Introduction to the Arithmetic Theory of Automorphic Functions*. Iwanami Shoten and Princeton University Press, 1971.

[174] Mahoro Shimura. Defining Equations of Modular Curves $X_0(N)$. *Tokyo J. Math.*, 18(2):443–456, 1995.

[175] Victor Shoup. Lower Bounds for Discrete Logarithms and Related Problems. Dans Walter Fumy, éditeur, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 de *Lecture Notes in Computer Science*, pages 256–266, Berlin, 1997. Springer-Verlag.

[176] Nigel P. Smart. The Discrete Logarithm Problem on Elliptic Curves of Trace One. *Journal of Cryptology*, 12(3):193–196, 1999.

[177] Edlyn Teske. On Random Walks for Pollard's Rho Method. *Mathematics of Computation*, 70(234):809–825, 2001.

[178] Nicolas Thériault. Index Calculus Attack for Hyperelliptic Curves of Small Genus. Dans Chi Sung Laih, éditeur, *Advances in Cryptology — ASIACRYPT 2003*, volume 2894 de *Lecture Notes in Computer Science*, pages 75–92, Berlin, 2003. Springer-Verlag.

[179] William C. Waterhouse. Abelian Varieties Over Finite Fields. *Annales Scientifiques de l'École Normale Supérieure, 4^e Série*, 2:521–560, 1969.

[180] Heinrich Weber. *Lehrbuch der Algebra*, volume 3: *Elliptische Funktionen und algebraische Zahlen*. Vieweg, Braunschweig, 2e édition, 1908. `http://www-gdz.sub.uni-goettingen.de/cgi-bin/digbib.cgi?PPN234719087`.

[181] André Weil. Sur les courbes algébriques et les variétés qui s'en déduisent, 1948. Dans *Courbes algébriques et variétés abéliennes*. Hermann, Paris, 1971.

[182] Annegret Weng. Class polynomials of CM-fields, 2001. `http://www.exp-math.uni-essen.de/zahlentheorie/classpol/class.html`.

[183] Douglas H. Wiedemann. Solving Sparse Linear Equations Over Finite Fields. *IEEE Transactions on Information Theory*, 32(1):54–62, janvier 1986.

[184] N. Yui et D. Zagier. On the Singular Values of Weber Modular Functions. *Mathematics of Computation*, 66(220):1645–1662, 1997.