Attaques algébriques du problème du logarithme discret sur courbes elliptiques

Vanessa Vitse

Université Versailles Saint-Quentin en Yvelines

UFR des Sciences
École Doctorale STV
Laboratoire PRiSM

THÈSE

présentée en vue de l'obtention du grade de

Docteur de l'Université Versailles Saint-Quentin en Yvelines

Spécialité :
Informatique

par

Vanessa VITSE

Attaques algébriques du problème du logarithme discret sur courbes elliptiques

Soutenue le jeudi 20 octobre 2011 devant le jury composé de :

M.	P. Elbaz-Vincent	professeur, université Joseph Fourier	président
M.	A. Enge	directeur de recherche, INRIA Bordeaux-Sud-Ouest	examinateur
M.	P. Gaudry	directeur de recherche, CNRS et université Nancy	rapporteur
M.	L. Goubin	professeur, université Versailles Saint-Quentin	co-directeur
M.	A. Joux	professeur, DGA et université Versailles Saint-Quentin	directeur
M.	G. Lecerf	chargé de recherche, CNRS et École Polytechnique	examinateur
M.	A. Lenstra	professeur, École Polytechnique Fédérale de Lausanne	rapporteur
M.	R. Lercier	professeur, DGA et université de Rennes	examinateur

À mon petit garçon

Remerciements

Mes premiers remerciements s'adressent tout naturellement à Antoine Joux, qui a indéniablement su me transmettre son goût pour la recherche et me donner l'élan dont j'avais besoin pour découvrir (enfin !) toutes les richesses du milieu. Je le remercie donc pour les très bons sujets qu'il m'a confiés, mais aussi pour ses incroyables intuitions qu'il partage toujours avec grande générosité et qui ont été une source d'inspiration inestimable durant ces trois années de thèse. Ses précieux conseils en programmation m'ont permis d'aborder sans effroi des calculs difficiles et aujourd'hui encore, je reste impressionnée par l'aisance et la rapidité avec laquelle il surmonte les difficultés liées à l'exercice. Je remercie également avec grand plaisir Louis Goubin pour m'avoir encadrée et co-encadrée depuis mon stage de Master. C'est d'abord grâce à lui que je me suis orientée en cryptographie, la qualité de ses cours et la clarté de ses explications m'ayant immédiatement convaincue d'intégrer son équipe. Sans son appui et sa confiance, il ne m'aurait certainement pas été possible d'aborder cette thèse dans d'aussi bonnes conditions tout en conservant mon statut de PRAG.

Je suis très reconnaissante à Pierrick Gaudry et Arjen Lenstra d'avoir accepté de rapporter cette thèse. L'étendue de leurs travaux dans le domaine m'inspirent la plus grande admiration, et je suis aujourd'hui très fière qu'ils m'aient fait l'honneur de relire ce manuscrit. Je tiens à remercier particulièrement Pierrick pour l'intérêt constant qu'il a porté à mes travaux, ainsi que l'ensemble de l'équipe CARAMEL pour les invitations chaleureuses (!) à Nancy. Un grand merci à Philippe Elbaz-Vincent, Andreas Enge, Grégoire Lecerf et Reynald Lercier d'avoir accepté de participer à ce jury de thèse. Je profite de l'occasion pour remercier encore une fois Reynald de m'avoir concocté durant mon mémoire de Master un package SEA adapté à mes besoins ; celui-ci me rend encore bien des services lorsque Magma peine à me trouver des bons exemples de courbes.

Mes remerciements vont également à Emmanuel Thomé, Dimitar Jetchev, Laurent Imbert, Andreas Enge, Elisa Gorla, Jérémie Detrey, Eleonora Guerrini, Thomas Sirvent, Romain Lebreton et l'équipe du LIP6 qui m'ont invitée aux workshops, écoles d'été, conférences et différents séminaires qu'ils ont organisés, me motivant ainsi à finaliser mes travaux. Non loin de ces équipes, je souhaite remercier particulièrement Damien Robert, Jean-Gabriel Kammerer, Jérémy Berthomieu et Maike Maissierer d'avoir relu attentivement articles ou parties de cette thèse ; merci pour leurs questions et remarques pertinentes, et pour les conversations enrichissantes que nous avons eues.

Parmi la "jeunesse versaillaise", j'aimerais saluer spécialement Sorina Ionica, Jérôme Plût, Michael Quisquater et Nicolas Gama pour les moments plus ou moins sérieux que nous avons passés à discuter de polynômes de Hilbert, pose de parquets, relèvements p-adiques, faire-part de mariage, séminaire d'équipe, desserts à l'azote liquide... Ils participent fortement à la bonne ambiance de ce laboratoire, à l'instar des célèbres cryptogirls désormais dispersées dans d'autres lieux.

Enfin, une mention spéciale à Gaëtan Bisson, Nicolas Estibals et Damien Robert, les cryptologues voileux qui m'ont embarquée dans leur galère charentaise. C'est lors de ce séjour et de la conférence qui a suivi que j'ai appris à faire de la recherche tout en ayant le mal de mer... une sensation bien retrouvée pendant les quatre mois qui ont suivi !

Il m'aurait été bien entendu impossible de réaliser cette thèse en seulement trois ans tout en effectuant mon (demi-)service de PRAG si mes collègues de l'IUT n'avaient pas contribué à mon projet. Je tiens donc à remercier sincèrement Stéphane Delaplace, Frédéric Plumet ainsi qu'Etienne Huot pour avoir accepté de m'attribuer chaque année l'allégement de service maximal auquel je pouvais prétendre. Laurent Dalmas, Pierre Doyen et Stéphan Soulayrol ont supporté sans (trop !) se plaindre mes contraintes hebdomadaires ainsi que les nombreux déplacements en conférence, merci

à eux d'avoir pris en compte tous mes desiderata. Je remercie également Jérôme Morio, Régine Delay, Jean-Claude Pissondes, Éric Fekete et Malika Izabachène qui ont assuré toutes les heures que je n'ai pas pu faire.

Je souhaite enfin témoigner toute ma gratitude à ma famille, notamment mon père qui participe toujours de près ou de loin à tous mes projets sans se lasser de mes changements de carrière. Les dernières lignes de ces remerciements seront consacrées à Grégoire : c'est grâce à son soutien et ses encouragements quotidiens que j'ai pu entreprendre cette tâche aussi lourde, il est certain que sans lui cette thèse n'aurait jamais pu voir le jour.

Table des matières

Introduction

C'est dans les années 1970 qu'émerge pour la première fois le concept de cryptographie à clef publique, permettant de s'affranchir du délicat problème de la distribution de clefs. L'article fondateur de Diffie et Hellman [DH76], basé sur des idées de Merkle, introduit la notion de fonction à sens unique qui est au cœur de ce nouveau paradigme cryptographique. Avec le développement concomitant de la puissance de calcul des ordinateurs, c'est naturellement du côté des mathématiques que les cryptologues se sont tournés pour trouver de telles fonctions. Deux principaux candidats ont été retenus à l'époque : la fonction exponentielle modulaire, dont la réciproque est appelée logarithme discret, qui est à la base de l'échange de clef Diffie-Hellman, du cryptosystème ElGamal et des schémas de signature qui en dérivent [DH76, ElG85] ; et la multiplication de deux nombres premiers, sur laquelle repose le très répandu cryptosystème RSA [RSA78].

Confrontés aux progrès des algorithmes de calcul du logarithme discret dans le groupe multiplicatif d'un corps fini, Koblitz et Miller réalisent indépendamment que les cryptosystèmes basés sur cette primitive peuvent en fait s'instancier sur tout type de groupe, et proposent d'utiliser le groupe des points rationnels d'une courbe elliptique, ou plus généralement de la jacobienne d'une courbe algébrique, définie sur un corps fini [Kob87, Mil86a]. Au moment où les cryptologues commencent à s'y intéresser, la géométrie algébrique vient de connaître plusieurs décennies de développement considérable. Si l'étude des courbes elliptiques remonte au XIXème siècle avec les travaux d'Abel et Weierstrass notamment, et que les propriétés des corps finis sont bien comprises dès le début du XXème siècle, c'est principalement à partir des années 1950 que la géométrie algébrique moderne prend son envol à partir des fondements posés par Serre et Grothendieck, culminant avec la démonstration complète des conjectures de Weil par Deligne en 1973. Cependant, peu de géomètres algébristes se doutaient à l'époque que ce domaine puisse avoir une quelconque application en dehors des mathématiques ; en particulier, peu de méthodes de calcul effectif existaient. Les années suivantes ont donc vu l'émergence d'une nouvelle discipline, la théorie algorithmique des nombres, et la publication d'importants algorithmes permettant l'utilisation concrète de la cryptographie basée sur courbes. On peut citer notamment Cantor pour la loi de groupe sur les jacobiennes de courbes hyperelliptiques [Can87], Schoof, Elkies, Atkin, Mestre et Satoh pour le comptage des points rationnels d'une courbe elliptique [Sch85, Sch95, Sat00], Atkin et Morain pour la multiplication complexe [AM93], et Miller pour les couplages [Mil04].

Parallèlement, les cryptanalystes s'efforcent depuis bientôt trente ans de développer des attaques permettant d'évaluer la sécurité des systèmes basés sur le problème du logarithme discret (DLP) sur variétés jacobiennes. Jusqu'à présent, toutes les attaques produites se regroupent en deux familles : les méthodes de transfert, et les méthodes de calcul d'indices imitant les techniques utilisées avec succès sur les corps finis. Ainsi pour des familles de courbes très spécifiques, il est possible de transférer le DLP sur un groupe plus faible. Dans cette catégorie d'attaques se trouvent la méthode de transfert par couplage de Menezes-Okamoto-Vanstone et Frey-Rück [MOV93, FR94] montrant la vulnérabilité des courbes supersingulières initialement proposées pour la simplicité du calcul de leur cardinalité, l'attaque des courbes anomales [SA98, Sem98, Sma99], ainsi que les techniques liées à la descente de Weil [Fre98, GHS02b, Die03]. Les méthodes de calcul d'indices, quant à elles, s'appliquent aux variétés jacobiennes de courbes de genre $g \geq 3$ [ADH94, Gau00, GTTD07, Die06], ou définies sur des extensions de petit degré de corps finis [Gau08, Nag10, Die11] ; dans ce deuxième cas, on préférera parler de *décompositions* plutôt que de calcul d'indices. Au final, la majorité des courbes elliptiques et hyperelliptiques de genre 2 semble résister à toutes ces attaques. L'objectif principal de cette thèse est d'améliorer les techniques existantes pour élargir le nombre de courbes vulnérables.

Une difficulté soulevée par les méthodes de décompositions concerne la résolution de systèmes polynomiaux multivariés. Elles font ainsi partie du domaine de la cryptanalyse algébrique, qui consiste à modéliser un cryptosystème sous la forme d'équations polynomiales, réduisant la sécurité de celui-ci à la difficulté de la résolution du système associé. Ici encore, les cryptologues héritent d'un important bagage mathématique. La théorie de l'élimination, que l'on peut faire remonter à Bézout et Gauss, a été principalement développée dans la deuxième moitié du XIXème siècle par des mathématiciens comme Sylvester et Cayley avant de subir une éclipse au début du XXème siècle, à l'exception toutefois des travaux de Macaulay et Gröbner. L'intérêt pour les méthodes effectives s'est réactivé avec les possibilités de calculs grandissantes des ordinateurs, et les bases de Gröbner introduites par Buchberger en 1965 se sont imposées comme un outil incontournable en calcul formel [Buc65]. Leur apparition en cryptologie arrive au moment où sont proposés les premiers schémas utilisant l'évaluation de systèmes polynomiaux multivariés comme fonction à sens unique [MI88, Pat96], et leur emploi pour la résolution du challenge HFE [FJ03] marque le premier grand succès de la cryptanalyse algébrique, faisant entrer définitivement les bases de Gröbner dans l'arsenal du cryptologue. Bien que des progrès importants aient été enregistrés ces dernières décennies [Buc85, Laz83, GM88, FGLM93] avec notamment les algorithmes F4 et F5 de Faugère [Fau99, Fau02], le calcul des bases de Gröbner reste un problème en général difficile et un domaine de recherche ouvert. C'est dans ce contexte que l'on propose dans cette thèse des méthodes adaptées à la résolution des systèmes polynomiaux qui interviennent dans les attaques par décompositions.

Organisation de la thèse et résultats

Ce mémoire se divise en deux parties. La première partie de cette thèse, plus orientée vers le calcul formel, résume les principaux outils utilisés par le cryptanalyste algébrique, notamment les algorithmes principaux de calcul de bases de Gröbner. Un nouvel algorithme dédié à la résolution de nombreux systèmes polynomiaux "similaires", et ayant fait l'objet de la publication [JV11b], y est présenté. La deuxième partie s'articule autour du problème du logarithme discret sur courbes algébriques en cryptographie. On s'intéresse spécifiquement au groupe défini par l'ensemble des points rationnels de la jacobienne d'une courbe algébrique définie sur un corps fini, et on liste les quelques attaques connues sur ce type de courbes, avec les méthodes de calcul d'indices et de transfert du DLP. Au centre de cette thèse sont alors détaillées les attaques sur courbes elliptiques définies sur des extensions de corps finis : après une analyse complète des techniques GHS et des méthodes de décomposition initialement introduites par Gaudry et Diem, on propose des variantes de ces méthodes permettant de fragiliser le DLP, ainsi que des problèmes reliés, pour des courbes elliptiques définies sur une gamme plus large d'extensions de corps finis. Ces résultats ont fait l'objet d'un deuxième article [JV10], d'un exposé invité [Vit10] ainsi que d'une collaboration [GJV10]. Enfin, on présente une nouvelle approche combinant les attaques par recouvrement avec les méthodes de décompositions : cette attaque permet entre autres de calculer complètement le logarithme discret sur courbes elliptiques définies sur des extensions sextiques de taille jamais atteinte auparavant. Ce dernier résultat fait l'objet d'une pré-publication [JV11a] et d'un deuxième exposé invité [Vit11].

Plus précisément, le chapitre 1 est dédié aux prérequis de la résolution de systèmes polynomiaux sur corps finis. On commence par rappeler les propriétés fondamentales des bases de Gröbner et les résultats principaux de la théorie de l'élimination pour les idéaux de dimension 0. On donne alors une méthode explicite pour résoudre les systèmes polynomiaux sur corps finis admettant un nombre fini de solutions ainsi que les outils utilisés pour l'estimation de la complexité des calculs

de base de Gröbner dans le cas homogène puis affine.

Le deuxième chapitre résume les techniques existantes de calcul de bases de Gröbner qui, historiquement, se regroupent en deux familles : la première se développe autour des idées de Buchberger présentées initialement dans sa thèse, alors que la seconde remonte à la théorie de l'élimination et des résultants et s'appuie sur des résultats de Lazard basés sur des calculs d'élimination gaussienne de matrices de Macaulay. Le problème des réductions à zéro étant fondamental dans ce type de calculs, on détaille les deux critères déjà donnés par Buchberger permettant d'éliminer a priori un maximum de ces réductions inutiles, ainsi que leur mise en place dans une version de l'algorithme de Buchberger proposée par Gebauer et Möller ; on donne en particulier une preuve rigoureuse de la correction de cet algorithme avec critères. On fait ensuite une description détaillée des algorithmes F4 et F5 de Faugère, qui s'inscrivent dans la lignée de ces idées. Ils sont considérés actuellement comme les plus efficaces pour le calcul de bases de Gröbner, avec notamment l'implantation de F4 en Magma qui reste une référence majeure dans le domaine. La programmation pratique de ces deux algorithmes, très peu détaillée dans la littérature, pose un certain nombre de challenges. On présente les solutions adoptées dans nos implantations qui utilisent entre autres les instructions SIMD pour l'élimination gaussienne, ainsi qu'une optimisation de la procédure de simplification utilisée par F4. Une reformulation plus simple de l'algorithme F5, comprenant une stratégie de sélection plus performante, est également proposée. On mentionne les difficultés restantes liées à l'implantation de F5 en suggérant des voies d'améliorations possibles. Après une analyse de complexité, ce chapitre se termine par une présentation de l'algorithme FGLM permettant d'effectuer un changement d'ordre en dimension 0, et d'accélérer les calculs de bases de Gröbner pour l'ordre lexicographique.

Le chapitre 3 est entièrement dédié à l'algorithme `F4Remake` conçu pour la résolution de nombreux systèmes polynomiaux issus d'une même famille de systèmes paramétrés. Ce type de systèmes apparaît naturellement en cryptanalyse algébrique, comme en témoignent les nombreux exemples d'applications donnés dans ce chapitre. On analyse en détail ce nouvel algorithme probabiliste basé sur le calcul de "traces" de F4, en donnant les probabilités de réussite sous des hypothèses réalistes en pratique. On illustre ses performances aussi bien sur des problèmes concrets issus de la cryptanalyse algébrique que sur des tests de référence, mettant ainsi en évidence des performances meilleures que celles de F4 et F5 dans ce contexte.

Les chapitres suivants concernent les attaques du problème du logarithme discret sur courbes algébriques. Le chapitre 4 se veut introductif et présente les principaux protocoles cryptographiques basés sur ce problème et ses variantes, ainsi que les algorithmes génériques applicables sur n'importe quel groupe et qui serviront de points de comparaison pour toutes les autres attaques proposées dans la suite. On y résume également le principe du calcul d'indices, qui est la méthode à l'origine des attaques les plus efficaces connues du DLP sur les groupes multiplicatifs de corps finis aussi bien que sur les jacobiennes de courbes algébriques.

Le chapitre 5 résume rapidement les prérequis essentiels de géométrie algébrique permettant d'appréhender les méthodes présentées dans les chapitres suivants : on détaille notamment la structure des courbes elliptiques et variétés jacobiennes de courbes hyperelliptiques définies sur des corps finis, ainsi que l'arithmétique existante sur ce type de courbes. Les attaques spécifiques à ces groupes et basées sur des méthodes de transfert sont ensuite listées, montrant que les courbes (hyper-)elliptiques de genre 1 ou 2 restent les plus sûres.

On se concentre alors dans les chapitres 6, 7 et 8 sur ces courbes lorsqu'elles sont définies sur des extensions de corps finis. On commence dans le chapitre 6 par un état de l'art des méthodes de transfert utilisant la descente de Weil, initialement proposée par Frey et à l'origine des techniques

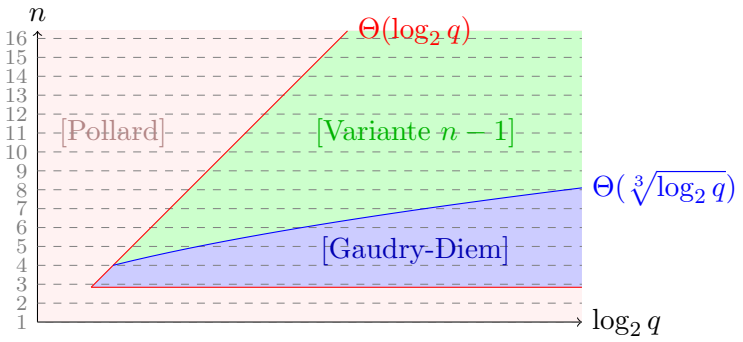
GHS. Des exemples de calculs explicites de recouvrements en caractéristique paire et impaire sont également donnés, principalement pour des extensions cubiques.

Dans le chapitre suivant sont présentées les prémices d'une généralisation de la méthode de calcul d'indices au contexte des courbes elliptiques, avec la première tentative de Semaev et ses polynômes de sommation. Ces polynômes constituant un outil important pour la recherche de relations dans les calculs d'indices présentés ensuite, on propose deux nouvelles méthodes permettant de les obtenir plus efficacement. On explique alors comment Gaudry et Diem ont pu concevoir à partir des idées de Semaev et des techniques de restriction de Weil, la première méthode de calcul d'indices sur courbe elliptique définie sur une extension d'un corps fini de petit degré. Plus précisément, la complexité de leur algorithme sur $E(\mathbb{F}_{q^n})$ est en $\tilde{O}(q^{2-2/n})$ à n fixé et $q \to \infty$, mais avec une constante cachée en n qui croît de façon sur-exponentielle en $2^{O(n^2)}$, rendant impraticable la résolution dès que $n \geq 5$. L'approche de Nagao est alors détaillée : elle permet de généraliser les outils introduits par Semaev pour la recherche de relations au cas des courbes de genre supérieur. Afin d'élargir la plage des corps finis sur lesquels les courbes sont plus vulnérables aux attaques par décomposition, on propose une première variante, appelée "$n-1$", de la méthode de Gaudry et Diem, aboutissant au résultat suivant :

Théorème. *Soit E une courbe elliptique définie sur $\mathbb{F}_{q^n}$ et G un sous-groupe du groupe des points rationnels de la courbe. Lorsque les hypothèses 7.3.2 et 7.3.4 sont vérifiées, il existe un algorithme permettant de résoudre le DLP défini sur G avec une complexité en*

$$\tilde{O}\left((n-1)!\left(2^{(n-1)(n-2)}e^n\, n^{-1/2}\right)^{\omega} q^2\right). \tag{1}$$

L'exposant ω intervenant dans le théorème est tel que la complexité de la multiplication de deux matrices de taille n est en $O(n^\omega)$ opérations. On montre ainsi que, sous des hypothèses standards, cette approche est asymptotiquement meilleure que les attaques génériques lorsque n reste inférieur à un multiple donné de $\log_2 q$, et est plus performante que celle de Gaudry et Diem lorsque n est plus grand qu'un multiple donné de $\sqrt[3]{\log_2 q}$.



Comparaison entre les méthodes de Pollard-rho, Gaudry and Diem et variante $n-1$.

Une autre utilisation de ces méthodes de décompositions est également donnée dans ce chapitre. On propose en effet un algorithme de résolution du problème Diffie-Hellman statique (SDHP) assisté d'un oracle sur $E(\mathbb{F}_{q^n})$; cet algorithme permet après $q/2$ appels à l'oracle de calculer une instance arbitraire de SDHP en un temps raisonnable. Ainsi, il devient possible, sous ces hypothèses d'accès à un oracle, de mener une attaque complète de SDHP sur une courbe standard de 155 bits en moins de 2 semaines, moyennant un accès à 1 000 processeurs de type Intel Xeon à 2.93 GHz. On termine enfin le chapitre 7 en proposant une autre variante de l'approche de Nagao. Celle-ci permet en

pratique d'accélérer la recherche de relations en genre $g > 1$; elle s'accompagne d'une technique de crible et est particulièrement bien adaptée au cas des courbes hyperelliptiques définies sur des extensions de degré pair.

Dans le dernier chapitre de cette thèse, on propose une nouvelle attaque du DLP spécifique aux courbes elliptiques définies sur des extensions de degré composé dont on compare l'efficacité à celle des techniques existantes. L'idée consiste à combiner les méthodes de recouvrement présentées en chapitre 6 pour transférer le DLP sur la jacobienne d'une courbe $\mathcal{C}$ définie sur une extension intermédiaire, avec les méthodes de décompositions pour attaquer le DLP sur la courbe ainsi obtenue. Cette technique est particulièrement efficace sur les courbes définies sur les extensions sextiques, et permet de réaliser une attaque complète du DLP d'une courbe de 132 bits a priori résistante à toute autre attaque connue en à peine 30 h de temps de calcul réel sur moins 200 coeurs (soit environ 3 700 h·CPU). On donne les détails de l'implantation faite en fin de chapitre.

Première partie

Algorithmique du calcul des bases de Gröbner

Chapitre 1

Bases de Gröbner et résolution de systèmes polynomiaux

Les bases de Gröbner constituent un outil essentiel pour les calculs faisant intervenir des idéaux d'anneaux de polynômes à plusieurs variables, notamment pour la résolution de systèmes polynomiaux. Elles permettent entre autres de généraliser à l'anneau $\mathbb{K}[X_1, \ldots, X_n]$ des opérations comme la division euclidienne et l'algorithme d'Euclide étendu qui n'existent que sur l'anneau des polynômes en une variable, ou encore l'élimination gaussienne qui ne s'applique qu'aux systèmes polynomiaux linéaires en plusieurs variables. En particulier, les bases de Gröbner sont indispensables pour la résolution de problèmes classiques comme le test d'appartenance d'un polynôme à un idéal I (ou problème "Ideal Membership"), les calculs de formes normales dans l'anneau des polynômes à plusieurs variables quotienté par I, l'étude des solutions d'un système polynomial, l'élimination, etc. Cependant l'analyse de la complexité des calculs de bases de Gröbner reste un sujet délicat. Le problème "Ideal Membership" faisant partie de la classe des problèmes EXPSPACE complets, les bornes supérieures de la complexité sont au moins exponentielles ; d'ailleurs, il existe des exemples particuliers pour lesquels la complexité du calcul est doublement exponentielle en le nombre de variables des polynômes engendrant l'idéal [MM84]. On verra néanmoins qu'il existe des familles d'idéaux pour lesquels les calculs sont plus accessibles.

Dans ce chapitre, on présente les définitions et principales propriétés des bases de Gröbner utiles pour la résolution de systèmes polynomiaux. On commence par rappeler dans la première section, la notion d'ordre monomial admissible en vue de présenter l'algorithme classique de division de polynômes à plusieurs variables. On étudie ensuite les idéaux monomiaux, plus faciles à appréhender que les idéaux polynomiaux, ainsi que leur représentation sous forme d'escalier, qui permet de déterminer facilement une base de l'anneau quotient $\mathbb{K}[X_1, \ldots, X_n]/I$. On montre alors comment les bases de Gröbner permettent de ramener l'étude des idéaux de polynômes à celle des idéaux monomiaux via la notion d'idéal initial. Les cas particuliers des calculs de bases de Gröbner pour les idéaux homogènes et de dimension 0 sont ensuite détaillés. La section suivante est dédiée aux techniques de résolution des systèmes polynomiaux : on commence par rappeler les principaux résultats de la théorie de l'élimination pour les idéaux de dimension 0, puis on donne une méthode explicite pour résoudre les systèmes polynomiaux sur corps finis admettant un nombre fini de solutions. Enfin, on présente dans la dernière section les principaux outils pour l'estimation de la complexité des calculs de base de Gröbner dans le cas homogène puis affine. Dans tout le chapitre, $\mathbb{K}$ désigne un corps quelconque. La plupart des résultats seront donnés sans démonstration. Le lecteur pourra se reporter à [CLO07] pour plus de détails.

1.1 Définitions et propriétés

1.1.1 Ordres monomiaux et divisions de polynômes

Définition 1.1.1. *Un ordre monomial admissible $\preccurlyeq$ sur $\mathbb{K}[X_1, \ldots, X_n]$ est une relation d'ordre sur l'ensemble des monômes de $\mathbb{K}[X_1, \ldots, X_n]$ qui est*

(i) totale (deux monômes peuvent toujours être comparés)

(ii) compatible avec la multiplication de $\mathbb{K}[X_1, \ldots, X_n]$: soient m_1 et m_2 deux monômes tels que $m_1 \preccurlyeq m_2$, alors $m_1 m_3 \preccurlyeq m_2 m_3$ pour tout monôme m_3

(iii) bien ordonnée : un ensemble non vide de monômes admet toujours un plus petit élément pour l'ordre $\preccurlyeq$.

Remarque 1.1.2. *Si $\preccurlyeq$ est un ordre monomial total vérifiant la condition (ii), alors la condition (iii) est équivalente à dire que tout monôme m vérifie $1 \preccurlyeq m$.*

Avant de donner des exemples d'ordre monomiaux, on introduit quelques notations : pour tout multi-degré $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, on définit

$$X^\alpha = X_1^{\alpha_1} \cdots X_n^{\alpha_n} \quad \text{et} \quad |\alpha| = \alpha_1 + \cdots + \alpha_n.$$

Les principaux ordres admissibles que l'on considère dans la suite sont l'ordre lexicographique, bien utile pour la résolution de systèmes polynomiaux sur corps finis, l'ordre lexicographique gradué renversé pour lequel les calculs de bases de Gröbner sont souvent les plus accessibles, ainsi que l'ordre lexicographique à poids qui permet parfois de rééquilibrer les degrés de façon à rendre les systèmes homogènes ou plus faciles à résoudre.

Ordres admissibles

1. *Ordre lexicographique (lex)* (ou ordre du dictionnaire) :
   $X^\alpha \prec_{lex} X^\beta$ si le premier coefficient non nul en partant de la gauche dans $\alpha - \beta$ est strictement négatif.
2. *Ordre lexicographique gradué (glex)* :
   $X^\alpha \prec_{glex} X^\beta$ si $|\alpha| < |\beta|$ ou $\big[|\alpha| = |\beta|$ et $X^\alpha \prec_{lex} X^\beta\big]$.
3. *Ordre lexicographique gradué renversé (grevlex)* :
   $X^\alpha \prec_{grevlex} X^\beta$ si $|\alpha| < |\beta|$ ou $\big[|\alpha| = |\beta|$ et le premier coefficient non nul en partant de la droite dans $\alpha - \beta$ est strictement positif$\big]$.
4. *Ordre d'élimininination (elim)* : Le k-ième ordre d'élimination $\prec_{elim_k}$ est défini par
   $X^\alpha \prec_{elim_k} X^\beta$ si $(\alpha_1 + \cdots + \alpha_k) < (\beta_1 + \cdots + \beta_k)$ ou $\big[(\alpha_1 + \cdots + \alpha_k) = (\beta_1 + \cdots + \beta_k)$ et $X^\alpha \prec_{grevlex} X^\beta\big]$.
5. *Ordre grevlex à poids (wgrevlex)* :
   On attribue aux variables $X_1, \ldots, X_n$ les poids respectifs $\omega_1, \ldots, \omega_n$ et on note $\omega \cdot \alpha = (\omega_1 \alpha_1, \ldots, \omega_n \alpha_n)$.
   $X^\alpha \prec_{wgrevlex} X^\beta$ si $|\omega \cdot \alpha| < |\omega \cdot \beta|$ ou $\big[|\omega \cdot \alpha| = |\omega \cdot \beta|$ et le premier coefficient non nul en partant de la droite de $\omega \cdot \alpha - \omega \cdot \beta$ est strictement positif$\big]$.

En particulier, pour les quatre premiers ordres on a $X_n \prec X_{n-1} \prec \cdots \prec X_1$. Une propriété intéressante du k-ième ordre d'élimination est que tout monôme contenant l'une des variables $X_1, \ldots, X_k$ est supérieur pour $\prec_{elim_k}$ à tout monôme ne contenant aucune de ces variables.

Définition 1.1.3. *Un ordre $\prec$ est* gradué par le degré *si $|\alpha| < |\beta|$ implique $X^\alpha \prec X^\beta$.*

Les ordres 2 et 3 sont des exemples d'ordres gradués par le degré.

Exemple 1.1.4. *On considère les monômes $m_1 = X_1X_3$, $m_2 = X_2^2$ et $m_3 = X_3^4$, leurs multi-degrés respectifs sont $\alpha = (1,0,1)$, $\beta = (0,2,0)$ et $\gamma = (0,0,4)$ et on a*

- $m_3 \prec_{lex} m_2 \prec_{lex} m_1$;
- $m_1 \prec_{grevlex} m_2 \prec_{grevlex} m_3$;
- $m_1 \prec_{wgrevlex} m_3 \prec_{wgrevlex} m_2$ *pour les poids* $\omega = (2,2,1)$.

Cette notion d'ordre sur les monômes à plusieurs variables permet de généraliser les définitions de terme de tête et de coefficient dominant naturellement définis par le degré pour les polynômes en une variable.

Dans la suite, on ne considère plus que des ordres monomiaux admissibles. Soit $\prec$ un ordre monomial défini sur l'ensemble des monômes de $\mathbb{K}[X_1, \ldots, X_n]$.

Définition 1.1.5. *Soit $f = \sum_\alpha c_\alpha X^\alpha$ un polynôme non nul de $\mathbb{K}[X_1, \ldots, X_n]$. On note $\gamma = \max\{\alpha \in \mathbb{N}^n : c_\alpha \neq 0\}$ le multi-degré du plus grand monôme de f pour l'ordre $\prec$.*

(i) Le monôme de tête de f est défini par $LM(f) = X^\gamma$.

(ii) Le coefficient dominant de f est défini par $LC(f) = c_\gamma$.

(iii) Le terme de tête de f est défini par $LT(f) = LC(f) \cdot LM(f) = c_\gamma X^\gamma$.

La queue du polynôme f fait référence au polynôme obtenu en supprimant dans f le terme de tête.

Ces notions, qui dépendent clairement de l'ordre choisi, se comportent comme attendu vis-à-vis de la somme et du produit.

Division de polynômes à plusieurs variables

Étant donné un ordre monomial admissible $\prec$ sur $\mathbb{K}[X_1, \ldots, X_n]$, il devient maintenant possible de créer pour les polynômes à plusieurs variables un analogue de la division euclidienne en une variable. Comme l'anneau $\mathbb{K}[X_1, \ldots, X_n]$ n'est plus principal, il est naturel de considérer la division d'un polynôme f non par rapport à un polynôme, mais par rapport à une liste ordonnée de polynômes $\{g_1, \ldots, g_s\}$. L'objectif est alors d'écrire f sous la forme :

$$f = q_1g_1 + \cdots + q_sg_s + r \text{ avec } \begin{cases} LM(q_ig_i) \preccurlyeq LM(f), \\ r = 0 \text{ ou } [\forall m \text{ monôme de } r,\ \forall i \in [\![1; s]\!],\ LM(g_i) \nmid m]. \end{cases} \tag{1.1}$$

Cette division s'obtient facilement en utilisant l'algorithme 1. La terminaison de cet algorithme est assurée, la suite des monômes de tête de f étant strictement décroissante et l'ordre $\prec$ étant admissible.

Algorithme 1: Algorithme de division dans $\mathbb{K}[X_1, \ldots, X_n]$.

Entrées : f polynôme à diviser, $G = \{g_1, \ldots, g_s\}$ famille de polynômes, $\prec$ ordre admissible
Sortie : r le reste et $\{q_1, \ldots, q_s\}$ les quotients tels que définis dans (1.1)
$r \leftarrow 0$, $q_i \leftarrow 0$ pour $i = 1, \ldots, s$
tant que $f \neq 0$ **faire**
  $i \leftarrow 1$
  tant que $i \leq s$ **faire**
    si $LM(g_i) | LM(f)$ **alors**
      $f \leftarrow f - \frac{LT(f)}{LT(g_i)} g_i, \quad q_i \leftarrow q_i + \frac{LT(f)}{LT(g_i)}$
      si $f = 0$ **alors retourner** r et $\{q_1, \ldots, q_s\}$
      $i \leftarrow 1$
    sinon $i \leftarrow i + 1$
  $r \leftarrow r + LT(f)$
  $f \leftarrow f - LT(f)$
retourner r et $\{q_1, \ldots, q_s\}$

Exemple 1.1.6. *On considère la division du polynôme* $f(X_1, X_2) = X_1^2X_2 + X_1X_2^2 + X_2^2$ *par* $G = \{X_1X_2 - 1, X_2^2 - 1\}$ *pour l'ordre lexicographique* $\prec_{lex}$.

$$
\begin{array}{rcrcrcrcr}
X_1^2X_2 & + & X_1X_2^2 & & & + & X_2^2 & & \\
-X_1^2X_2 & & & + & X_1 & & & & \\
\hline
 & & X_1X_2^2 & + & X_1 & + & X_2^2 & & \\
 & & -X_1X_2^2 & & & & & + & X_2 \\
\hline
 & & & & [X_1] & + & X_2^2 & + & X_2 \\
 & & & & & & X_2^2 & + & X_2 \\
 & & & & & & -X_2^2 & & \\
\hline
 & & & & & & & & [X_2]
\end{array}
\qquad
\begin{array}{rcr}
+ & 1 & \\
\hline
 & & \\
 & & \\
\hline
 & & \\
 & & \\
+ & 1 & \\
\hline
+ & 1 & \\
 & & [1]
\end{array}
$$

$X_1X_2 - 1$	$X_2^2 - 1$	r
X_1	0	0
$X_1 + X_2$	0	0
$X_1 + X_2$	0	X_1
$X_1 + X_2$	1	
$X_1 + X_2$	1	$X_1 + X_2$
$X_1 + X_2$	1	$X_1 + X_2 + 1$

Par conséquent, la division de f *par la famille ordonnée* $G = \{X_1X_2 - 1, X_2^2 - 1\}$ *est* $f = (X_1 + X_2) \cdot [X_1X_2 - 1] + [X_2^2 - 1] + (X_1 + X_2 + 1)$. *Si l'on inverse l'ordre des polynômes dans* G, *on obtient la division suivante :*

$$
\begin{array}{rcrcrcrcr}
X_1^2X_2 & + & X_1X_2^2 & & & + & X_2^2 & & \\
-X_1^2X_2 & & & + & X_1 & & & & \\
\hline
 & & X_1X_2^2 & + & X_1 & + & X_2^2 & & \\
 & & -X_1X_2^2 & + & X_1 & & & & \\
\hline
 & & & & [2X_1] & + & X_2^2 & & \\
 & & & & & & X_2^2 & & \\
 & & & & & & -X_2^2 & + & 1 \\
\hline
 & & & & & & & & [1]
\end{array}
$$

$X_2^2 - 1$	$X_1X_2 - 1$	r
0	X_1	0
X_1	X_1	0
X_1	X_1	$2X_1$
$X_1 + 1$	X_1	$2X_1$
$X_1 + 1$	X_1	$2X_1 + 1$

En particulier, $f = (X_1 + 1) \cdot [X_2^2 - 1] + (X_1) \cdot [X_1X_2 - 1] + (2X_1 + 1)$.

On notera $\overline{f}^G$ ce reste qui dépend de l'ordre des polynômes dans la famille G. Il est immédiat que si $\overline{f}^G$ est nul, alors f est dans l'idéal engendré par la famille G. En revanche la réciproque, fausse

en général, ne sera vérifiée que lorsque la famille G constitue un "bon" système de représentants de l'idéal ; en particulier, pour ce système spécifique, la condition d'appartenance d'un polynôme à l'idéal engendré par G sera équivalente à la nullité de $\overline{f}^G$ et la division sera indépendante de l'ordre choisi pour les polynômes de G.

1.1.2 Idéaux monomiaux et initiaux

Avant d'introduire les bases de Gröbner, on définit la notion d'idéal initial qui est un cas particulier d'idéal monomial.

Définition 1.1.7. *Un idéal $I \subset \mathbb{K}[X_1, \ldots, X_n]$ est appelé* idéal monomial *s'il existe une famille de monômes l'engendrant.*

Soit $I = \langle (X^\alpha)_{\alpha \in A} \rangle$ un idéal monomial engendré par la famille des monômes dont les multi-degrés sont dans $A \subset \mathbb{N}^n$. Alors il est clair que X^β est un monôme de I si et seulement X^β est divisible par l'un des monômes engendrant I. En particulier, l'ensemble $S = \{\gamma \in \mathbb{N}^n : X^\gamma \in I\} = \{\alpha + \beta : \alpha \in A, \beta \in \mathbb{N}^n\}$ est une partie stable de $\mathbb{N}^n$ pour l'addition, dont on peut donner une représentation schématique sous forme d' "escalier" :

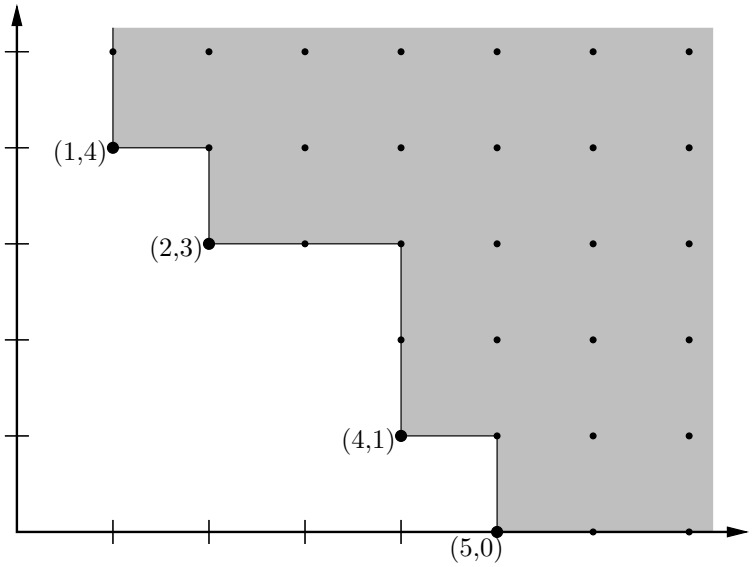


FIGURE 1.1 – Escalier représentant l'idéal monomial $I = \langle X_1^5,\ X_1^4 X_2,\ X_1^2 X_2^3,\ X_1 X_2^4 \rangle$

Une famille minimale de générateurs de I s'obtient alors comme les "coins" de l'escalier, et cette famille est nécessairement finie d'après le lemme de Dickson (ou la noethérianité de $\mathbb{K}[X_1, \ldots, X_n]$). Les monômes sous l'escalier, ou plus précisément leurs images par l'application quotient, forment une base en tant que $\mathbb{K}$-espace vectoriel de l'anneau quotient $\mathbb{K}[X_1, \ldots, X_n]/I$.

Pour la classe des idéaux monomiaux, il est très facile de résoudre des problèmes liés à la division de polynômes, comme par exemple le problème d'appartenance à un idéal. Ainsi, un polynôme f appartient à I si et seulement si tous ses termes appartiennent à I, ou autrement dit si et seulement si f est une combinaison $\mathbb{K}$-linéaire de monômes de I. On montre dans la suite comment exploiter ces résultats en associant à tout idéal muni d'un ordre, un idéal monomial :

Définition 1.1.8. *Soient $I \subset \mathbb{K}[X_1, \ldots, X_n]$ et $\prec$ un ordre monomial admissible sur $\mathbb{K}[X_1, \ldots, X_n]$. On appelle* idéal initial *de I l'idéal monomial*

$$LT(I) = \langle LM(f) : f \in I \rangle.$$

L'escalier associé à I et $\prec$ est l'ensemble $S = \{\gamma \in \mathbb{N}^n : X^\gamma \in LT(I)\}$.

En particulier, un monôme m appartient à $LT(I)$ si, et seulement si, il existe $f \in I$ tel que $m = LM(f)$. Comme précédemment, l'ensemble des monômes sous l'escalier forme une famille génératrice et donc une base de $\mathbb{K}[X_1, \ldots, X_n]/I$. En effet, soit f un représentant d'un élément du quotient admettant au moins un terme dans $LT(I)$, et soit m le plus grand de ces termes. Alors, il existe $g \in I$ tel que $m = LT(g)$. On peut remplacer f par $f - g$ et faire ainsi décroître strictement le plus grand terme présent dans $LT(I)$. On peut recommencer jusqu'à ce que f n'ait plus de terme dans $LT(I)$, ce qui arrive après un nombre fini d'étapes car l'ordre est admissible ; f s'écrit alors comme une combinaison linéaire de monômes sous l'escalier.

Pour $I = \langle f_1, \ldots, f_s \rangle$, on a nécessairement $\langle LM(f_1), \ldots, LM(f_s) \rangle \subset LT(I)$, mais l'inclusion est stricte en général :

Exemple 1.1.9. *Soit $I = \langle X_1 - X_2, X_1 - X_2^2 \rangle$ un idéal de $\mathbb{R}[X_1, X_2]$ muni de l'ordre lexicographique, alors $LT(I) = \langle X_1, X_2^2 \rangle \neq \langle X_1 \rangle$.*

En revanche, si l'on choisit des polynômes de I dont les monômes de tête engendrent l'idéal initial $LT(I)$, alors ces polynômes sont des générateurs de I (voir preuve du théorème 4 de [CLO07]) :

Lemme 1.1.10.
Soit $I \subset \mathbb{K}[X_1, \ldots, X_n]$ un idéal. Pour tout idéal $J \subset I$, si $LT(J) = LT(I)$ alors $J = I$.

1.1.3 Bases de Gröbner

Dans ce qui suit I désigne un idéal de $\mathbb{K}[X_1, \ldots, X_n]$ muni d'un ordre monomial $\prec$.

Définition 1.1.11. *On appelle* base de Gröbner *de l'idéal I toute famille $G = \{g_1, \ldots, g_s\}$ de polynômes de I telle que $\langle LM(g_1), \ldots, LM(g_s) \rangle = LT(I)$.*

Il est clair qu'une telle famille de polynômes existe (théorème de la base incomplète de Hilbert) : on a vu en section 1.1.2 que l'idéal initial $LT(I)$ est finiment engendré par des monômes $m_1, \ldots, m_s$ et que pour chaque m_i, il existe $g_i \in I$ tel que $m_i = LM(g_i)$; cette famille $G = \{g_1, \ldots, g_s\}$ est alors une base de Gröbner de l'idéal I. Enfin, le lemme 1.1.10 implique que toute base de Gröbner est bien une base de l'idéal.

Exemple 1.1.12. *Soit I un idéal de $\subset \mathbb{R}[X_1, X_2, X_3]$ défini par*

$$I = \langle X_1^2 + X_2^2 + X_3^2 - 25, X_1^2 - X_2^2 - (X_3 - 4)^2 + 9, (X_1 - 3)^2 + X_2^2 - 10 \rangle.$$

On vérifie sans difficulté que $G = \{X_1^2 + 4X_3 - 16, X_2^2 - 6X_1 - 4X_3 + 15, X_3^2 + 6X_1 - 24\}$ est une base de Gröbner de I pour l'ordre $grevlex_{X_1 \succ X_2 \succ X_3}$.

Une première propriété intéressante des bases de Gröbner est d'offrir via la division un test valide d'appartenance à l'idéal ("Ideal Membership") :

Lemme 1.1.13. *Soient $G = \{g_1, \ldots, g_s\}$ une base de Gröbner de l'idéal I et f un polynôme de $\mathbb{K}[X_1, \ldots, X_n]$. Alors il existe un unique polynôme r tel que*

(i) aucun terme de r n'est divisible par un monôme de $LT(G) = \{LM(g_1), \ldots, LM(g_s)\}$;

(ii) le polynôme $f - r$ est dans I.

On obtient explicitement le polynôme r en calculant la division de f par G, qui ne dépend donc plus de l'ordre choisi pour les polynômes de G. Ce reste, noté $\overline{f}^G$, est appelé *forme normale* de f modulo G.

Théorème 1.1.14. *Soient G une base de Gröbner de l'idéal I pour l'ordre $\prec$ et $f \in \mathbb{K}[X_1, \ldots, X_n]$. Alors $f \in I$ si et seulement si $\overline{f}^G = 0$.*

Telle qu'on l'a définie, une base de Gröbner n'est pas nécessairement unique pour un idéal donné. Certains polynômes de la base peuvent en effet être redondants : si $g \in G$ a son terme de tête divisible par un monôme de $LT(G \setminus \{g\})$, alors $\langle LT(G)\rangle = \langle LT(G \setminus \{g\})\rangle$, donc $G \setminus \{g\}$ est encore une base de Gröbner. En éliminant ces polynômes redondants et en normalisant les polynômes restant dans la base G, on obtient une base de cardinalité minimale :

Définition 1.1.15. *Une* base de Gröbner minimale *de I est une base de Gröbner G de I telle que*

(i) pour tout $g \in G$, $LC(g) = 1$;

(ii) pour tout couple de polynômes distincts $g, g' \in G$, $LM(g) \nmid LM(g')$.

En particulier, toutes les bases minimales ont la même cardinalité. Pour répondre au problème de l'unicité, on introduit la notion plus restrictive de base de Gröbner réduite :

Définition 1.1.16. *Une* base de Gröbner réduite *de I est une base de Gröbner G de I telle que*

(i) pour tout $g \in G$, $LC(g) = 1$;

(ii) pour tout couple de polynômes distincts $g, g' \in G$, aucun monôme de g n'est divisible par $LM(g')$.

Il devient alors facile de déterminer si deux ensembles de polynômes engendrent le même idéal en comparant les bases de Gröbner réduites.

1.1.4 Idéaux homogènes

Les idéaux homogènes sont des objets importants en géométrie projective. En particulier, l'homogénéisation d'un idéal affine (resp. un système polynomial) permet l'étude des points d'une variété (resp. les solutions du système) à l'infini.

Dans cette section, on explicite le lien entre les procédures d'homogénéisation et les calculs de bases de Gröbner. On introduit également la notion de d-base de Gröbner pour un ensemble de polynômes homogènes de degré inférieur ou égal à d ; cette notion, d'intérêt indépendant (puisqu'elle permet par exemple de résoudre le problème "Ideal Membership" pour un polynôme de degré inférieur à d sans avoir à calculer la base complète), sera reprise dans le chapitre 2, en remarque 2.1.6 ou encore en section 2.2 pour présenter un algorithme de calcul de base de Gröbner dû à Lazard.

Définition 1.1.17.

(i) Un polynôme $f \in \mathbb{K}[X_0, \ldots, X_n]$ est homogène *de degré total d si tous ses termes sont de degré total d.*

(ii) Un idéal de $\mathbb{K}[X_0, \ldots, X_n]$ est dit homogène *s'il existe un nombre fini de polynômes homogènes $f_1, \ldots, f_s$ tels que $I = \langle f_1, \ldots, f_s \rangle$.*

Si l'idéal $I = \langle f_1, \ldots f_s \rangle$ est homogène alors tout polynôme $f \in I$ a ses composantes homogènes dans I. On en déduit en particulier que $LT(I) = \langle LT(f) : f \text{ homogène} \rangle$, et donc que I admet une base de Gröbner formée de polynômes homogènes avec les mêmes arguments que précédemment.

On fait maintenant le lien entre la base de Gröbner d'un idéal $I = \langle f_1, \ldots, f_s \rangle \subset \mathbb{K}[X_1, \ldots, X_n]$ et celle de l'idéal homogène $I^h = \langle f_1^h, \ldots, f_s^h \rangle \subset \mathbb{K}[X_0, X_1, \ldots, X_n]$ associé au système de générateurs de I. On introduit à cet effet un ordre monomial $\prec_h$ sur $\mathbb{K}[X_0, X_1, \ldots, X_n]$ compatible à l'ordre $\prec$ sur $\mathbb{K}[X_1, \ldots, X_n]$.

Définition 1.1.18. *Soit $g \in \mathbb{K}[X_1, \ldots, X_n]$ de degré total d.*

(i) On appelle homogénéisé *de g par rapport à X_0 le polynôme $g^h \in \mathbb{K}[X_0, \ldots, X_n]$ tel que*

$$g^h(X_0, \ldots, X_n) = X_0^d \, g\left(\frac{X_1}{X_0}, \ldots, \frac{X_n}{X_0}\right).$$

(ii) Si $f \in \mathbb{K}[X_0, \ldots, X_n]$ est homogène, alors $f^*(X_1, \ldots, X_n) = f(1, X_1, \ldots, X_n)$ est le déshomogénéisé *de f.*

(iii) À un ordre monomial $\prec$ sur $\mathbb{K}[X_1, \ldots, X_n]$, on associe l'ordre $\prec_h$ défini sur $\mathbb{K}[X_0, \ldots, X_n]$ par : $m_1 \prec_h m_2$ si $\deg m_1 < \deg m_2$ ou $[\deg m_1 = \deg m_2$ et $m_1^* \prec m_2^*]$.

On vérifie aisément que l'ordre défini en (iii) est bien un ordre admissible. En pratique, on considérera souvent l'ordre lex (resp. grevlex) sur $\mathbb{K}[X_1, \ldots, X_n]$ avec $X_1 \succ \ldots \succ X_n$; l'ordre associé $\prec_h$ sur $\mathbb{K}[X_0, \ldots, X_n]$ est alors l'ordre glex (resp. grevlex) avec $X_1 \succ_h \ldots \succ_h X_n \succ_h X_0$.

Propriété 1.1.19.

1. *Pour tout $g \in \mathbb{K}[X_1, \ldots, X_n]$, $g = (g^h)^*$.*
2. *Pour tout $f \in \mathbb{K}[X_0, X_1, \ldots, X_n]$ homogène, $f = X_0^\ell \, (f^*)^h$ où X_0^ℓ est la plus grande puissance de X_0 divisant f.*
3. *$f \in I^h \Rightarrow f^* \in I$, mais la réciproque est fausse en général.*
4. *Pour tout $f \in \mathbb{K}[X_0, X_1, \ldots, X_n]$ homogène, $LT_\prec(f^*) = LT_{\prec_h}(f)^*$.*

Avec le point 4 de cette propriété, il est donc possible d'obtenir la base de Gröbner de I pour un ordre donné à partir de celle de I^h pour l'ordre correspondant.

Proposition 1.1.20.
Soit $I = \langle f_1, \ldots, f_s \rangle$ un idéal de $\mathbb{K}[X_1, \ldots, X_n]$ et $I^h = \langle f_1^h, \ldots, f_s^h \rangle \subset \mathbb{K}[X_0, X_1, \ldots, X_n]$ l'idéal homogène associé aux générateurs de I.
Si $G = \{g_1, \ldots, g_r\}$ est une base de Gröbner de l'idéal I^h pour un ordre $\prec_h$ formée de polynômes homogènes, alors la famille $G^* = \{g_1^*, \ldots, g_r^*\} \subset I$ est également un base de Gröbner de I pour l'ordre $\prec$.

La réciproque de ce résultat est clairement fausse dans le cas général :

Exemple 1.1.21. *Soit $I = \langle X, X^2 + X + 1\rangle$ muni de l'unique ordre admissible donné par le degré ; il est facile de voir que $I = \mathbb{K}[X]$ et que donc une base de Gröbner est $\{1\}$. L'idéal homogène associé est $I^h = \langle X, X^2 + XY + Y^2\rangle$, et l'ordre $\prec_h$ correspond à l'unique ordre gradué en 2 variables avec $X \succ Y$. La base de Gröbner de I^h est $\{X, Y^2\}$ donc distincte de l'homogénéisation de la base de Gröbner de I. En revanche, on retrouve bien que la déshomogénéisation de la base de Gröbner de I^h est une base de Gröbner de I (non minimale).*

Dans ce qui suit, on note $\mathbb{K}_d[X_0, \ldots, X_n]$ l'ensemble des polynômes homogènes de degré inférieur ou égal à d. On introduit la notion de base de Gröbner tronquée à un degré d pour les idéaux homogènes :

Définition 1.1.22. *Soit $I \subset \mathbb{K}[X_0, \ldots, X_n]$ un idéal homogène.*
On dit que $G \subset \mathbb{K}[X_0, \ldots, X_n]$ est une d-base de Gröbner *de I si G engendre I et vérifie l'une des propriétés équivalentes suivantes :*

(i) pour tout $s \in LT(I \cap \mathbb{K}_d[X_0, \ldots, X_n])$, il existe $t \in LT(G)$ tel que $t \mid s$;

(ii) $\overline{f}^G = 0$ pour tout $f \in I \cap \mathbb{K}_d[X_0, \ldots, X_n]$.

On constate que les polynômes de G de degré supérieur strict à d ne sont pas utiles pour la vérification des deux propriétés équivalentes données dans la définition précédente. Si l'on considère pour tout entier positif d, une d-base de Gröbner G_d minimale réduite d'un idéal homogène I donné, la suite croissante $(G_d)_{d\in\mathbb{N}}$ est stationnaire à partir d'un certain rang D ; en particulier, G_D est une base de Gröbner de l'idéal I.

1.1.5 Idéaux de dimension 0

L'objectif étant de résoudre des systèmes polynomiaux, il est naturel de s'intéresser au cas où les solutions sont en nombre fini. On introduit à cet effet les idéaux de dimension 0 qui sont ceux dont la variété associée se compose d'un nombre fini de points dans $\overline{\mathbb{K}}[X_1, \ldots, X_n]$.

Définition 1.1.23. *Un idéal I est de* dimension 0 *si le quotient $\mathbb{K}[X_1, \ldots, X_n]/I$ est de dimension finie en tant que $\mathbb{K}$-espace vectoriel. Dans ce cas, on appelle* degré de l'idéal *la dimension de ce quotient.*

Un tel idéal correspond donc à la notion de système polynomial bien déterminé, admettant un nombre fini de solutions. Le degré de l'idéal correspond alors au nombre de solutions (ou de points de la variété) dans une clôture algébrique, comptées avec multiplicité.

Par ailleurs, on a vu que les monômes sous l'escalier de l'idéal initial forme une base du quotient $\mathbb{K}[X_1, \ldots, X_n]/I$. Un idéal est donc de dimension 0 si et seulement si le nombre de monômes sous l'escalier est fini, ce qui est équivalent à dire que l'escalier "touche" les axes : pour toute variable X_i, il existe un entier $n_i \in \mathbb{N}$ tel que $X_i^{n_i} \in LT(I)$. Le nombre de monômes sous l'escalier correspond alors au degré de l'idéal.

Exemple 1.1.24. *On a vu en exemple 1.1.12 que l'idéal*

$$I = \langle X_1^2 + X_2^2 + X_3^2 - 25, X_1^2 - X_2^2 - (X_3 - 4)^2 + 9, (X_1 - 3)^2 + X_2^2 - 10\rangle \subset \mathbb{R}[X_1, X_2, X_3]$$

muni de l'ordre l'ordre $grevlex_{X_1 \succ X_2 \succ X_3}$ admet pour base de Gröbner

$$G = \{X_1^2 + 4X_3 - 16, X_2^2 - 6X_1 - 4X_3 + 15, X_3^2 + 6X_1 - 24\}.$$

En comptant le nombre de monômes sous l'escalier associé à I, on trouve que le degré de l'idéal I est égal à 8 : ceci correspond au nombre de points d'intersection de trois quadriques dans la clôture algébrique $\mathbb{C}$ de $\mathbb{R}$.

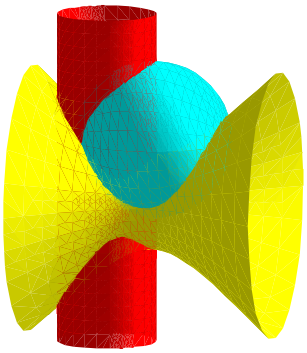


FIGURE 1.2 – Intersection des trois quadriques intervenant dans les exemples 1.1.12, 1.1.24 et 1.2.5.

On voit qu'un calcul de bases de Gröbner permet de déterminer facilement quand un système admet un nombre fini de solutions, et d'en déterminer la cardinalité. Dans la suite, on montre comment ces bases permettent en fait de résoudre le système.

1.2 Technique de résolution des systèmes polynomiaux sur corps finis

Soit $f_1, \ldots, f_s \in \mathbb{K}[X_1, \ldots, X_n]$ un système polynomial à plusieurs variables, dont on veut trouver les solutions. Si l'idéal associé n'est pas de dimension 0, le nombre de solutions est a priori infini, auquel cas on se donne pour objectif une description simple de la variété associée à l'idéal via la théorie de l'élimination.

1.2.1 Élimination et "shape lemma"

Définition 1.2.1. *Soit I un idéal de $\mathbb{K}[X_1, \ldots, X_n]$. Pour $k \in [\![1; n]\!]$, on appelle* k-ième idéal d'élimination *l'idéal $I_k = I \cap \mathbb{K}[X_k, \ldots, X_n]$.*

Il est immédiat de vérifier que I_k est bien un idéal de $\mathbb{K}[X_k, \ldots, X_n]$. La connaissance des idéaux d'élimination permet d'obtenir une "triangularisation" du système associé à l'idéal I, donnant une sorte d'analogue de l'algorithme de Gauss pour des polynômes linéaires.

Proposition 1.2.2. *Soient I un idéal de $\mathbb{K}[X_1, \ldots, X_n]$ et G une base de Gröbner de I pour l'ordre lexicographique. Alors pour tout $k \in [\![1; n]\!]$, $G \cap \mathbb{K}[X_k, \ldots, X_n]$ est une base de Gröbner de I_k pour l'ordre lexicographique.*

Démonstration. En effet, si $f \in I_k$ alors il existe un polynôme $g \in G$ tel que $LT(g) | LT(f)$. En particulier, $LT(g) \in \mathbb{K}[X_k, \ldots, X_n]$ donc $g \in \mathbb{K}[X_k, \ldots, X_n]$, l'ordre choisi étant l'ordre lexicographique. Le résultat découle alors directement du lemme 1.1.10. □

Remarque 1.2.3. *Si l'on souhaite simplement éliminer les variables $X_1, \ldots, X_k$, il n'est pas nécessaire de calculer une base de Gröbner pour l'ordre lexicographique : on peut en effet calculer une base de Gröbner G de I pour le k-ième ordre d'élimination (moins coûteux en général). On obtient alors que $G \cap \mathbb{K}[X_k, \ldots, X_n]$ est une base de Gröbner pour l'ordre $\prec_{grevlex}$.*

Dans le cas des idéaux de dimension 0, la forme d'une base de Gröbner pour l'ordre lexicographique est souvent très simple :

Proposition 1.2.4 (Shape Lemma)**.**
Soit $I \subset \mathbb{K}[X_1, \ldots, X_n]$ un idéal radical de dimension 0 et de degré d. Alors, après presque tout changement linéaire de coordonnées (autrement dit, pour tout changement linéaire de coordonnées en dehors d'un fermé de Zariski), la base de Gröbner de I pour l'ordre lexicographique est de la forme

$$\{X_1 - g_1(X_n)\ ,\ \ldots\ ,\ X_{n-1} - g_{n-1}(X_n)\ ,\ g_n(X_n)\}$$

où g_n est un polynôme univarié de degré égal à d et $g_1, \ldots, g_{n-1}$ sont des polynômes univariés de degré inférieur strict à d.

Ce résultat se démontre assez simplement lorsqu'après changement de coordonnées, les points de la variété associée à I ont leurs dernières coordonnées distinctes. Dans le cas général, les idéaux d'élimination I_k pour $k < n$ ne sont pas forcément aussi simples, mais restent dans la plupart des cas engendrés par des polynômes de petit degré en X_k [BMMT94].

Exemple 1.2.5. *Pour l'idéal*

$$I = \langle X_1^2 + X_2^2 + X_3^2 - 25, X_1^2 - X_2^2 - (X_3 - 4)^2 + 9, (X_1 - 3)^2 + X_2^2 - 10\rangle \subset \mathbb{R}[X_1, X_2, X_3]$$

présenté dans les exemples 1.1.12 et 1.1.24, une base de Gröbner pour l'ordre $lex_{X_1 \succ X_2 \succ X_3}$ est

$$G_1 = \{6\,X_1 + X_3^2 - 24, X_2^2 + X_3^2 - 4\,X_3 - 9, X_3^4 - 48\,X_3^2 + 144\,X_3\}$$

qui n'est pas exactement de la forme attendue. Si l'on échange les coordonnées X_2 et X_3, on obtient une base de Gröbner

$$G_2 = \{816\,X_1 - X_2^6 - 71X_2^4 + 1449\,X_2 - 9825,\ 544\,X_3 + X_2^6 + 71\,X_2^4 - 1585\,X_2^2 + 7785,\\ X_2^8 + 60\,X_2^6 - 2298\,X_2^4 + 26172\,X_2^2 - 99711\}$$

pour l'ordre $lex_{X_1 \succ X_3 \succ X_2}$. Il est alors facile de déterminer les points d'intersection réels des trois quadriques : $V(I) \cap \mathbb{R}^3 = \{(4, -3, 0), (4, 3, 0)\}$, voir figure 1.2.

Connaissant une base de Gröbner pour l'ordre lexicographique d'un idéal de dimension 0, il est facile de déterminer les points de la variété associée. Le n-ième idéal d'élimination, étant principal, est engendré par un polynôme g_n dont on peut calculer les racines. En remplaçant X_n par les valeurs de ces racines dans I_{n-1}, on en déduit facilement les valeurs associées pour X_{n-1}, puis on remonte de la sorte jusqu'à X_1. La seule difficulté résulte donc dans la résolution de polynôme univarié, que l'on étudie dans la section suivante.

1.2.2 Recherche de racines de polynômes univariés sur corps finis

Le problème de déterminer les racines d'un polynôme univarié sur un corps quelconque est généralement un problème délicat. Cependant, dans le cas des corps finis, on connaît des algorithmes probabilistes de complexité polynomiale en la taille du corps et le degré du polynôme.

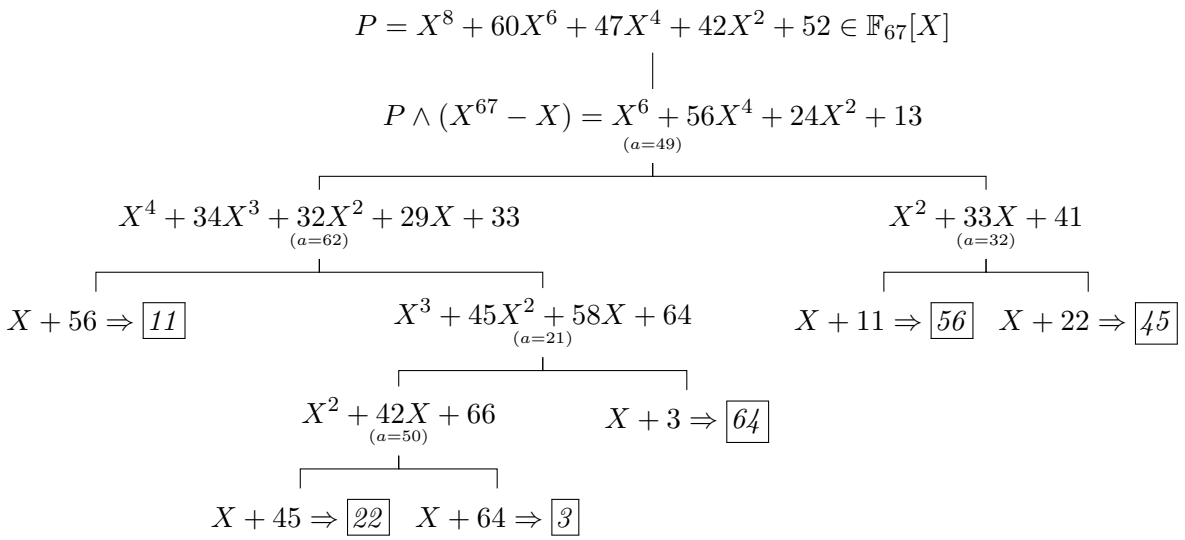
On considère dans la suite un polynôme univarié $f \in \mathbb{F}_q[X]$ unitaire dont on souhaite calculer les racines dans $\mathbb{F}_q$. Soit $f = \prod_{i=1}^{m} f_i^{\alpha_i}$ sa factorisation en facteurs irréductibles distincts de $\mathbb{F}_q[X]$. Une première étape consiste à obtenir $g = \prod_{i,\deg(f_i)=1} f_i$ par le calcul du pgcd de f et $X^q - X$, de façon à ne récupérer que les termes linéaires de f sans facteurs carrés. Pour trouver des facteurs non triviaux de g en caractéristique impaire, on calcule ensuite le pgcd de g avec des polynômes de la forme $g_a(X) = (X-a)^{(q-1)/2} - 1$ où a est un élément aléatoire de $\mathbb{F}_q$. En effet, pour un $a \in \mathbb{F}_q$ donné, le polynôme g_a a pour racines la moitié des éléments de $\mathbb{F}_q$ et celles-ci sont réparties de façon relativement aléatoire dans $\mathbb{F}_q$; la probabilité qu'une racine de g soit également racine de g_a est donc égale à $1/2$. En particulier, la probabilité que le pgcd h_a de g et g_a soit un facteur non trivial de g lorsque $\deg(g) \geq 2$ est égale $1 - (1/2)^{\deg g - 1}$. Ainsi au bout d'au plus deux essais en moyenne, on trouve un facteur non trivial h_a de g ; on peut alors itérer le procédé sur les deux polynômes h_a et g/h_a jusqu'à trouver tous les facteurs de degré 1 de g. Avec des arguments similaires, on peut également montrer (voir [G03]) que la profondeur moyenne de la récurrence est en $\log(\deg(g))$. On remarque à ce stade, qu'il est possible lorsque $\deg(g) = 2$ de calculer explicitement les racines de g, afin d'accélérer les calculs. On résume avec la fonction ExtractionRacine donnée ci-dessous, l'algorithme qui permet de calculer les racines dans $\mathbb{F}_q$ d'un polynôme de la forme $g = \prod_{i,\deg(f_i)=1} f_i \in \mathbb{F}_q[X]$ avec q puissance impaire d'un nombre premier.

Fonction ExtractionRacine(g)

Entrées : $g \in \mathbb{F}_q[X]$ produit de facteurs linéaires sans carrés, $q = p^\ell$ où p premier impair
Sortie : les racines de g dans $\mathbb{F}_q$

1. **si** $\deg(g) = 1$ **alors retourner** la racine évidente de g
2. **sinon si** $\deg(g) = 0$ **alors retourner** $\emptyset$
3. **sinon répéter**
4. $\quad a \leftarrow \text{Random}(\mathbb{F}_q)$
5. $\quad$ calculer $\tilde{g}_a \leftarrow (X-a)^{\frac{q-1}{2}} - 1 \bmod g$ par exponentiation rapide
6. $\quad h_a \leftarrow g \wedge \tilde{g}_a$
7. **jusqu'à** $h_a \neq 1$ **ou** $h_a \neq g$
8. **retourner** `ExtractionRacine(`h_a`)` $\cup$ `ExtractionRacine(`g/h_a`)`

Exemple 1.2.6. *On extrait les racines dans* $\mathbb{F}_{67}$ *du polynôme univarié de l'exemple 1.2.5 :*



Le polynôme P admet donc six racines dans $\mathbb{F}_{67}$ *:* $3, 11, 22, 45, 56, 64$.

Si l'on note $M(d)$ le nombre d'opérations nécessaires dans $\mathbb{F}_q$ pour effectuer la multiplication de deux polynômes de degré d, la complexité de la première étape de l'algorithme qui consiste à ramener le calcul des racines de f à celles de g est en $\log q \cdot M(\deg(f))$ pour le calcul de $h = X^q - X \bmod f$ avec une exponentiation modulaire, et en $\log(\deg f) \cdot M(\deg(f))$ pour le calcul du pgcd de f et h. La complexité à chaque appel récursif de la fonction ExtractionRacine est en $(\log q + \log(d))M(\deg(g))$ pour les calculs d'exponentiation modulaire et de pgcd faits en ligne 5 et 6, ce qui donne une complexité en $\tilde{O}(d^2 \log q)$ pour le calcul des racines d'un polynôme univarié de degré d qui s'écrit comme produit de d facteurs linéaires distincts dans $\mathbb{F}_q[X]$.

La complexité totale pour la recherche de racines d'un polynôme $f \in \mathbb{F}_q[X]$ est donc en $\tilde{O}(\deg(f)^2 \log q)$.

Remarque 1.2.7. *L'algorithme précédent et son analyse se transposent sans difficulté à $\mathbb{F}_{2^n}[X]$ en remplaçant les polynômes g_a par des polynômes de la forme $\sum_{k=0}^{n-1}(aX)^{2^k}$, où a est un élément aléatoire de $\mathbb{F}_{2^n}$.*

1.3 Degré de régularité

Afin d'estimer la complexité d'un calcul de base de Gröbner, on introduit la notion de degré de régularité d'un idéal.

1.3.1 Polynôme de Hilbert d'un idéal

On a vu en section 1.1.5 qu'un idéal est de dimension 0 si et seulement si le nombre de monômes sous l'escalier associé à cet idéal est fini, et ce indépendemment de l'ordre choisi. Plus généralement, la croissance du nombre de monômes sous l'escalier, comptés par la fonction de Hilbert, permet de définir la dimension d'un idéal.

Définition 1.3.1.

(i) Soit $I \subset \mathbb{K}[X_0, \ldots, X_n]$ un idéal homogène.
On note $\mathcal{H}_t = \{\overline{f} \in \mathbb{K}[X_0, \ldots, X_n]/I : f$ homogène, $\deg(f) = t\} \cup \{0\}$; c'est un sous-espace vectoriel de $\mathbb{K}[X_0, \ldots, X_n]/I$. La fonction de Hilbert de l'idéal homogène I est définie par

$$\mathrm{HF}_I(t) = \dim_{\mathbb{K}}(\mathcal{H}_t).$$

(ii) Soit $I \subset \mathbb{K}[X_1, \ldots, X_n]$ un idéal quelconque.
On note $\mathcal{A}_t = \{\overline{f} \in \mathbb{K}[X_1, \ldots, X_n]/I : \deg(f) \leq t\}$. La fonction de Hilbert de l'idéal I est définie par

$$\mathrm{HF}_I(t) = \dim_{\mathbb{K}}(\mathcal{A}_t).$$

En utilisant le fait que les monômes sous l'escalier forment une base de l'anneau quotient, on déduit facilement que dans le cas homogène $\mathrm{HF}_I(t)$ est égal au nombre de monômes sous l'escalier de degré t, et dans le cas affine $\mathrm{HF}_I(t)$ est égal au nombre de monômes de degré inférieur ou égal à t sous l'escalier, à condition que I soit muni d'un ordre monomial gradué.

Proposition 1.3.2. *Il existe un polynôme $\mathrm{HP}_I \in \mathbb{Q}[X]$, appelé* polynôme de Hilbert *de l'idéal I, tel que pour tout t suffisamment grand, $\mathrm{HF}_I(t) = \mathrm{HP}_I(t)$.*

Le polynôme de Hilbert vérifiant $\mathrm{HP}_I(t) \in \mathbb{N}$ pour t suffisamment grand, on peut montrer qu'il existe des entiers $a_0, \ldots, a_s \in \mathbb{Z}$ tels que

$$\mathrm{HP}_I(t) = \sum_{i=0}^{s} a_i \binom{t}{s-i} \tag{1.2}$$

où s est le degré de HP_I. On peut déduire de ce polynôme numérique des informations sur l'idéal, telles que sa dimension ou son *degré de régularité*.

Définition 1.3.3.

(i) La dimension *de l'idéal I est définie comme étant le degré s du polynôme de Hilbert* HP_I.

(ii) On appelle degré de régularité *de l'idéal I, le plus petit entier $d_{reg}(I)$ pour lequel* $\mathrm{HF}_I(t) = \mathrm{HP}_I(t)$ *pour tout* $t \geq d_{reg}(I)$.

Avec cette définition, on obtient qu'un idéal est de dimension 0 lorsque le nombre de monômes sous l'escalier de l'idéal est fini, ce qui coïncide avec la définition 1.1.23. En particulier, avec les notations précédentes, lorsque I est de dimension 0, a_0 correspond au *degré* de l'idéal I. D'un point de vue géométrique, le degré du polynôme HP_I correspond à la dimension de la variété $V(I)$ définie comme le lieu d'annulation des polynômes de l'idéal I.

1.3.2 Degré de régularité d'un idéal

On montre dans cette section le lien qui existe entre le degré de régularité d'un idéal et une base de Gröbner de cet idéal pour un ordre gradué par le degré.

Étant donnée une base de Gröbner d'un idéal pour un ordre gradué, il est facile de trouver une borne supérieure sur le degré de régularité de cet idéal. Comme la fonction de Hilbert diffère du polynôme de Hilbert en petit degré à cause des irrégularités de l'escalier, une fois passée le dernier "creux" (par exemple le point $(3, 2)$ sur la figure 1.3), elle devient égale au polynôme.

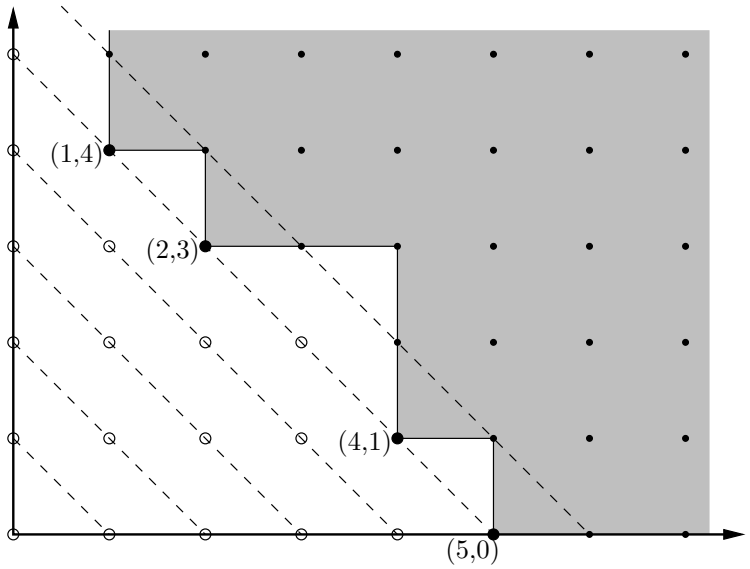


FIGURE 1.3 – Borne supérieure de $d_{reg}(I)$ pour $I = \langle X_1^5,\ X_1^4X_2,\ X_1^2X_2^3,\ X_1X_2^4 \rangle$

Proposition 1.3.4. *Soient I un idéal homogène de $\mathbb{K}[X_0, \ldots, X_n]$, resp. idéal affine de $\mathbb{K}[X_1, \ldots, X_n]$, et G une base de Gröbner minimale de I pour un ordre gradué. Si d est le degré maximal des éléments de G, alors $d_{reg}(I) \leq (n+1)(d-1)+1$, resp. $d_{reg}(I) \leq n(d-1)$.*

Cette borne supérieure est en fait optimale comme le montre l'exemple de l'idéal monomial

$$I = \langle X_1^d, \ldots, X_n^d \rangle \subset \mathbb{K}[X_1, \ldots, X_n].$$

Il est par contre en général faux de dire que le degré de régularité d'un idéal est une borne sur le degré maximal des générateurs d'une base de Gröbner pour un ordre gradué donné.

Exemple 1.3.5. *Soit $I = \langle X_1^3,\ X_0^2X_1 - X_2^3 \rangle$ un idéal de $\mathbb{K}[X_0, X_1, X_2]$. La base de Gröbner pour l'ordre $grevlex_{X_0 \succ X_1 \succ X_2}$ de cet idéal est $G = \{X_1^3,\ X_0^2X_1 - X_2^3,\ X_1^2X_2^3,\ X_1X_2^6, X_2^9\}$ de degré maximal 9. Pourtant, le degré de régularité de l'idéal homogène est seulement $d_{reg} = 4$ comme on peut le voir avec la base de Gröbner $G_2 = \{X_2^3 - X_1X_0^2,\ X_1^3\}$ pour l'ordre $grevlex_{X_0 \prec X_1 \prec X_2}$.*

Il existe pourtant des bornes valables pour tous les idéaux avec un ordre gradué mais ces bornes sont assez mauvaises, car en général exponentielles en le degré de régularité. Par exemple, à partir de la représentation de Hartshorne du polynôme de Hilbert

$$\mathrm{HP}_I(t) = \sum_{i=0}^{s} \left[\binom{t+i}{i+1} - \binom{t+i-m_i}{i+1} \right]$$

qui est équivalente à celle donnée en équation 1.2, on montre que $\max(d_{reg}(I), m_0)$ est une borne supérieure précise du degré maximal des éléments d'une base de Gröbner minimale d'un idéal homogène I. Mais, même avec cette borne, on peut trouver des exemples d'idéaux pour lesquels le degré maximum obtenu est exponentiel en le nombre de variables n et doublement exponentiel en la dimension de l'idéal s (voir [MM84]). Cependant, ces exemples restent des cas très particuliers ; on verra que dans les situations qui nous intéressent le degré de régularité est une borne souvent convenable pour le degré des éléments d'une base de Gröbner minimale pour un ordre gradué.

Chapitre 2

Algorithmes classiques de calcul de bases de Gröbner

Les bases de Gröbner sont introduites pour la première fois en 1965 par Buchberger dans sa thèse [Buc65], il y présente notamment un critère permettant de dire si un ensemble de polynômes donnés forme une base de Gröbner. De ce critère découle directement un algorithme permettant de calculer, pour un idéal de polynômes muni d'un ordre admissible, la base de Gröbner associée. Par la suite, vont se distinguer essentiellement deux familles d'algorithmes de calcul de bases de Gröbner : la première se développe autour de l'algorithme original de Buchberger [Buc85, GM88, Fau99, Fau02], alors que la deuxième fait référence à la théorie de l'élimination et des résultants et s'appuie sur la réduction de matrices de Macaulay par élimination gaussienne [Mac02, Laz83, CKPS00, MMDB08].

Dans la première section de ce chapitre, on commence par rappeler les résultats fondamentaux donnés dans la thèse de Buchberger pour le calcul des bases de Gröbner, en mettant en évidence le problème des réductions à zéro. On présente alors l'algorithme de Buchberger ainsi que les deux critères proposés pour pallier partiellement ce problème. La deuxième section est dédiée aux résultats de Lazard [Laz83] portant sur la théorie de l'élimination gaussienne des matrices de Macaulay. On montrera notamment comment déduire de cette approche et de la notion de degré de régularité, une borne supérieure sur la complexité du calcul de bases de Gröbner. On présente ensuite l'algorithme F4 introduit en 1999 par Faugère [Fau99] qui pour la première fois combine les idées des deux familles d'algorithmes. F4 est la première version réellement efficace de l'algorithme de Buchberger et son implantation en Magma [BCP97] constitue aujourd'hui encore une référence majeure pour les calculs de bases de Gröbner. Le code de cette dernière implantation n'étant cependant pas publique, on détaille dans la suite les choix faits pour notre propre implantation en langage C de l'algorithme. On donnera en chapitre 3 les résultats obtenus en temps et mémoire avec notre implantation sur différents benchmarks, montrant des performances comparables à celles de Magma. Bien que très efficace, cet algorithme F4 ne résout cependant pas le problème des réductions à zéro déjà soulevé par Buchberger en 1979 [Buc79]. On s'intéressera donc également au critère F5 proposé par Faugère en 2002 [Fau02], permettant d'éliminer a priori bien plus de réductions à zéro que ceux de Buchberger. On présente ce critère en section 2.4 en mettant en évidence les difficultés liées à l'implantation de l'algorithme incrémental F5 ; on y rappelle également les résultats connus sur la complexité de cet algorithme. La dernière section est dédiée aux algorithmes de changement d'ordre : après avoir détaillé l'intérêt de cette approche, on présente le cas particulier du changement d'ordre en dimension 0 avec l'algorithme FGLM [FGLM93] dont on fait l'analyse de la complexité.

2.1 Buchberger

On introduit ici la notion de S-polynôme, désignant littéralement un "polynôme de syzygie", qui permet de donner la principale caractérisation des bases de Gröbner. De cette caractérisation, on déduit immédiatement un algorithme simple qui permet le calcul des bases de Gröbner mais qui génère beaucoup de calculs inutiles. On raffine par la suite cet algorithme en donnant les deux critères de Buchberger permettant d'éviter certaines réductions à zéro.

Dans la suite, on supposera donné un ordre admissible $\prec$.

2.1.1 Caractérisation par les syzygies

Le calcul du S-polynôme d'un couple de polynômes donnés est l'opération consistant à prendre la combinaison monomiale de ces polynômes la plus simple permettant d'éliminer les termes de tête de ces polynômes :

Définition 2.1.1. *Soient g_1, g_2 deux polynômes de $\mathbb{K}[X_1, \ldots, X_n]$. On note $S(g_1, g_2)$ le S-polynôme de g_1, g_2 défini par*

$$S(g_1, g_2) = u_1 g_1 - u_2 g_2$$

où $lcm = LM(g_1) \vee LM(g_2)$ et $u_i = \frac{lcm}{LT(g_i)}$ pour $i = 1, 2$.

On appelle paire critique*, le quintuplet formé des données $(lcm, u_1, g_1, u_2, g_2)$; le* degré de la paire *est défini par le degré du monôme lcm. Dans la suite, on utilisera indifféremment la notion de paire critique pour désigner la paire ou le S-polynôme associé.*

On a vu en section 1.1.3 que si $G = \{g_1, \ldots, g_s\}$ est une base de Gröbner d'un idéal I alors $I = \langle g_1, \ldots, g_s \rangle$ et $LT(G) = LT(I)$. En particulier, pour vérifier si une famille de polynômes $\{f_1, \ldots, f_r\}$ est une base de Gröbner, on peut considérer les polynômes $S(f_i, f_j)$ dont les termes de tête ne sont pas trivialement dans $\langle LT(f_1), \ldots, LT(f_r) \rangle$. Comme ces polynômes sont naturellement dans l'idéal I engendré par $\{f_1, \ldots, f_r\}$, si la division de l'un des $S(f_i, f_j)$ par $\{f_1, \ldots, f_r\}$ n'est pas égale à 0, alors $\{f_1, \ldots, f_r\}$ ne peut pas être une base de Gröbner. Cette idée est à la base théorème 2.1.2 :

Théorème 2.1.2 (Buchberger)**.**
Une famille $G = \{g_1, \ldots, g_s\} \subset \mathbb{K}[X_1, \ldots, X_n]$ est une base de Gröbner de l'idéal I engendré par G si et seulement si pour tout couple $(i, j) \in [\![1; s]\!]^2$, le reste dans la division de $S(g_i, g_j)$ par G, noté $\overline{S(g_i, g_j)}^G$, est nul.

Démonstration. voir la preuve du théorème 6 p. 85 de [CLO07]. □

Exemple 2.1.3. *Soient $f_1 = X_1X_2 - X_1X_3$, $f_2 = X_1^2X_3 - X_3^3$ et $f_3 = X_2X_3^2 - X_3^3$ des polynômes de $\mathbb{K}[X_1, X_2, X_3]$ où $\mathbb{K}$ corps quelconque. La famille $G = \{f_1, f_2, f_3\}$ forme une base de Gröbner pour l'ordre $grevlex_{X_1 \succ X_2 \succ X_3}$:*
$S(f_1, f_2) = X_1X_3f_1 - X_2f_2 = -X_3f_2 + X_3f_3 \Rightarrow \overline{S(f_1, f_2)}^G = 0$,
$S(f_1, f_3) = X_3^2f_1 - X_1f_3 = 0 \Rightarrow \overline{S(f_1, f_3)}^G = 0$,
$S(f_2, f_3) = X_2X_3f_2 - X_1^2f_3 = X_3^2f_2 - X_3^2f_3 \Rightarrow \overline{S(f_2, f_3)}^G = 0$.

Il est en fait possible de donner une caractérisation plus fine des bases de Gröbner, en introduisant une notion un peu moins contraignante que celle donnée par la division en théorème 2.1.2.

Définition 2.1.4. *Soient $G = \{g_1, \ldots, g_s\} \subset \mathbb{K}[X_1, \ldots, X_n]$, $p \in \mathbb{K}[X_1, \ldots, X_n]$ et m un monôme. On note $p = o_G(m)$ s'il existe des polynômes $u_1, \ldots, u_s \in \mathbb{K}[X_1, \ldots, X_n]$ tels que $p = \sum_{i=1}^{s} u_i g_i$ et $LM(u_i g_i) \prec m$ pour tout $i \in [\![1; s]\!]$.*

En particulier, si g_1, g_2 sont deux polynômes, alors $\overline{S(g_i, g_j)}^G = 0$ implique nécessairement $S(g_1, g_2) = o_G(LM(g_1) \vee LM(g_2))$. On a en fait la caractérisation suivante :

Théorème 2.1.5. *Une famille $G = \{g_1, \ldots, g_s\} \subset \mathbb{K}[X_1, \ldots, X_n]$ est une base de Gröbner de l'idéal I engendré par G si et seulement si*

$$\forall (i, j) \in [\![1; s]\!]^2,\ S(g_i, g_j) = o_G(LM(g_i) \vee LM(g_j)).$$

Ce résultat sera utile à la mise en place du critère F5 pour un calcul de bases de Gröbner avec moins de réductions à zéro.

2.1.2 Algorithme de Buchberger, version simple

Du théorème 2.1.2, on déduit facilement l'algorithme 2 qui calcule de façon rudimentaire une base de Gröbner d'un idéal I donné.

Algorithme 2: Version basique de l'algorithme de Buchberger

Entrées : $I = \langle f_1, \ldots, f_k \rangle \subset \mathbb{K}[X_1, \ldots, X_n]$
Sortie : G base de Gröbner de I

1. $G \leftarrow \{f_1, \ldots, f_k\}$
2. CP $\leftarrow \{S(f_i, f_j), 1 \leq i < j \leq k\}$
3. **tant que** CP $\neq \emptyset$ **faire**
4. choisir $s \in$ CP et l'extraire de CP
5. $r \leftarrow \overline{s}^G$
6. **si** $r \neq 0$ **alors**
7. CP $\leftarrow$ CP $\cup \{S(g, r) : g \in G\}$
8. $G \leftarrow G \cup \{r\}$
9. **retourner** G

On vérifie sans difficulté que la propriété pour l'idéal I d'être engendré par G est un invariant de boucle. À chaque tour de boucle, soit l'idéal initial de I croît (lorsque le reste r est non nul, on ajoute un nouveau terme de tête dans $LT(I)$), soit le nombre de paires critiques diminue. En particulier, la noethérianité de $\mathbb{K}[X_1, \ldots, X_n]$ impose que l'algorithme s'arrête. À la sortie de la boucle, tout S-polynôme se réduit nécessairement à 0 dans G, et l'algorithme retourne bien une base de Gröbner de l'idéal I.

Remarque 2.1.6. *L'algorithme de Buchberger s'adapte très facilement au calcul des d-bases de Gröbner d'un idéal homogène. On peut montrer en effet que pour un idéal homogène $I \subset \mathbb{K}[X_0, \ldots, X_n]$, un ensemble $G \subset \mathbb{K}[X_0, \ldots, X_n]$ est une d-*base de Gröbner *de I si et seulement si G engendre I et $\overline{S(g_1, g_2)}^G = 0$ pour tous $g_1, g_2 \in G^2$ tels que $\deg(LT(g_1) \vee LT(g_2)) \leq d$. En particulier si l'on ne considère dans l'algorithme 2 que les paires critiques dont le degré est inférieur à d, on obtient en sortie une d-base de Gröbner de l'idéal homogène I.*

Bien que très simple, cet algorithme soulève déjà quelques difficultés d'implantation, avec notamment le problème du choix des paires critiques en ligne 4 ou encore celui de l'ordre des polynômes dans G pour la division en ligne 5. Ces choix n'ont aucune importance pour l'exactitude du résultat, mais influencent grandement les performances du calcul. Il peut être judicieux par exemple de ranger dans G les polynômes par terme de tête croissant, afin de minimiser le nombre de comparaisons dans la division : les polynômes de la base ayant les plus petits termes de tête sont en effet plus souvent susceptibles d'être diviseurs.

La stratégie optimale de sélection des paires critiques est un sujet plus fondamental. Une première idée, appelée *stratégie normale*, consiste à sélectionner les paires $(lcm, u_i, g_i, u_j, g_j)$ dont le lcm est minimal pour l'ordre donné, de façon à ajouter le plus tôt possible de nouveaux éléments dans la base afin d'augmenter les chances de réduction à zéro des paires suivantes. Cette stratégie est souvent adoptée pour les calculs avec l'ordre grevlex, pour lequel elle donne d'assez bons résultats en pratique. D'autres stratégies heuristiques ont également été proposées, voir par exemple [Cza91, GMN$^+$91], y compris pour les ordres non gradués tel que l'ordre lexicographique.

2.1.3 Le problème des réductions à zéro

Dans l'algorithme 2, la réduction des paires critiques occupe l'essentiel du temps de calcul. Pourtant en pratique, on constate que la division en ligne 5 donne extrêmement souvent un reste égal à zéro. Il est donc naturel de chercher à diminuer au maximum le nombre de paires à considérer.

Une première idée consiste à minimiser le nombre de polynômes dans la base G en éliminant au fur et à mesure les polynômes redondants. Lorsqu'un polynôme r avec un nouveau terme de tête est ajouté dans la base (ligne 8 de l'algorithme 2), un polynôme g de G devient redondant si son terme de tête est divisible par $LT(r)$; en effet, g se retrouve immédiatement à partir de $r \in G$ et de $S(r, g) = g - \frac{LT(g)}{LT(r)} r \in \text{CP}$, donc n'est pas utile pour la description de la base de l'idéal. Si l'on modifie en conséquence l'algorithme 2, l'invariant de boucle devient "I est engendré par G et les S-polynômes de CP ", et l'exactitude de l'algorithme modifié se montre alors comme précédemment. Ainsi, en "purgeant" ces polynômes redondants, on garantit un nombre minimal de polynômes dans G à chaque étape et on obtient en sortie une base de Gröbner minimale.

Pour améliorer les performances de calcul, il est également indispensable d'utiliser les critères de Buchberger [Buc79, Buc85] qui permettent d'éliminer directement certaines paires inutiles dont on peut prédire la réduction à zéro durant le calcul :

Proposition 2.1.7 (Critères de Buchberger)**.**
Soit $G \subset \mathbb{K}[X_1, \ldots, X_n]$ et f, g, h des polynômes dans G.

1. Premier critère : *Si $LM(f) \wedge LM(g) = 1$ (on dit que les polynômes f et g sont* étrangers*, alors $\overline{S(f,g)}^{\{f,g\}} = 0$.*
2. Deuxième critère : *Si $S(f, g) = o_G(LM(f) \vee LM(g))$ et $S(f, h) = o_G(LM(f) \vee LM(h))$ et si $LM(f) \mid (LM(g) \vee LM(h))$, alors $S(g, h) = o_G(LM(g) \vee LM(h))$.*

Le deuxième critère peut s'interpréter de la façon suivante : si la paire de (f, g) et la paire de (f, h) ont été traitées précédemment, alors la paire de (g, h) n'est pas utile dès lors que le terme de tête de f divise le ppcm des termes de tête de g et h. Il faut faire attention cependant que ce critère ne permet d'éliminer qu'une paire sur les trois ; en particulier, si $LT(f) \mid LT(h)$ et $LT(g) \mid LT(h)$ (ce qui équivaut à $LT(f) \mid [LT(g) \vee LT(h)]$ et $LT(g) \mid [LT(f) \vee LT(h)]$), il faut faire un choix entre l'élimination de la paire (f, h) et de la paire (g, h).

D'autres difficultés sont liées à l'implantation de ces critères : pour éliminer un maximum de paires, il peut être judicieux par exemple de ne pas éliminer immédiatement les paires qui satisfont le premier critère ; celles-ci peuvent en effet être utiles pour l'élimination d'autres paires avec le deuxième critère.

2.1.4 Algorithme de Buchberger avec critères

On donne dans cette section le pseudo-code de l'algorithme de Buchberger avec critères suivant la version proposée par Gebauer et Möller dans [GM88]. Pour la mise à jour de la base courante G et de la liste des paires critiques CP lors de l'ajout d'un nouveau polynôme, on fait appel à l'algorithme 4 qui met en place les critères donnés en proposition 2.1.7 ainsi que la purge des polynômes expliquée en section précédente. Dans cet algorithme, on note $paire(f, g)$ la paire critique (lcm, u, f, v, g) telle que $S(f, g) = uf - vg$ et $lcm = [LM(f) \vee LM(g)]$. Afin d'éliminer un maximum de paires, on introduit trois sous-ensembles CP_0, CP_1 et CP_2 contenant respectivement les paires qui peuvent être éliminées avec le premier critère, avec le deuxième et celles ne pouvant pas être éliminées. Pour éviter le problème d'élimination de "deux paires sur trois" avec le deuxième critère, on ajoute la condition de divisibilité stricte donnée en ligne 2 de l'algorithme 4.

Algorithme 3: Algorithme de Buchberger avec critères

Entrées : $I = \langle f_1, \ldots, f_k \rangle \subset \mathbb{K}[X_1, \ldots, X_n]$
Sortie : G base de Gröbner minimale de I
$G \leftarrow \emptyset$; CP $\leftarrow \emptyset$;
pour $i = 1$ *à* k **faire** `Update(`f_i`)`;
tant que CP $\neq \emptyset$ **faire**
  choisir $s \in$ CP et l'extraire de CP;
  $r \leftarrow \overline{s}^G$;
  si $r \neq 0$ **alors** `Update(`r`)`;
retourner G;

La preuve de terminaison de l'algorithme 3 se fait comme précédemment en utilisant la noethérianité de $\mathbb{K}[X_1, \ldots, X_n]$. Pour montrer l'exactitude de cet algorithme, on va supposer dans un premier temps que la purge en ligne 16 de l'algorithme 4 n'est pas faite. On a alors le même invariant de boucle $I = \langle G \rangle$, en particulier l'idéal I est bien engendré par G en fin d'algorithme. Il reste encore à vérifier que $S(g_i, g_j) = o_G(LT(g_i) \vee LT(g_j))$ pour tous $g_i, g_j \in G$. Ceci est clair pour toutes les paires dont le reste a été ajouté à G et toutes celles qui ont été éliminées par le critère 1. Pour montrer que celles éliminées par le critère 2 vérifient aussi cette propriété en fin d'algorithme, on introduit l'ensemble $E = \{(g_i, g_j) : 1 \leq i < j \leq \#G\}$ muni de la relation suivante : $(g_{i_0}, g_{j_0}) \to (g_{i_1}, g_{j_1})$ si la paire (g_{i_0}, g_{j_0}) est éliminée par la paire (g_{i_1}, g_{j_1}). En particulier, deux paires en relation ont toujours un polynôme en commun, et chaque paire a zéro ou deux successeurs. Si une paire (g_i, g_j) n'a pas de successeur, alors elle vérifie bien la propriété $S(g_i, g_j) = o_G(LT(g_i) \vee LT(g_j))$. Si les deux successeurs d'une paire vérifient cette propriété, alors le critère 2 impose que cette paire le vérifie aussi. En remontant de proche en proche dans E, on peut montrer que toutes les paires vérifient cette propriété, à condition que la relation $\to$ ne crée pas de cycle. Par l'absurde, supposons que E possède un tel cycle $(g_{i_0}, g_{j_0}) \to (g_{i_1}, g_{j_1}) \to \cdots \to (g_{i_\ell}, g_{j_\ell}) \to (g_{i_0}, g_{j_0})$. On peut déjà remarquer que $(g_{i_k}, g_{j_k}) \to (g_{i_{k+1}}, g_{j_{k+1}})$ implique que $LT(g_{i_{k+1}}) \vee LT(g_{j_{k+1}})$ divise $LT(g_{i_k}) \vee LT(g_{j_k})$; toutes les paires du cycle ont donc le même lcm. Comme les paires éliminées en ligne 2 de l'algorithme 4 doivent vérifier une divisibilité stricte des lcm, celles-ci ne peuvent pas

Algorithme 4: Algorithme Update (installation des critères de Buchberger)

Entrée : $f \in \mathbb{K}[\underline{X}]$
Résultat : Mise à jour de G et CP avec f

1. **pour tout** $paire = (lcm, t_1, g_1, t_2, g_2) \in \mathrm{CP}$ **faire**
2. | **si** $[LM(f) \vee LM(g_1)] \mid lcm$ *strictement* **et** $[LM(f) \vee LM(g_2)] \mid lcm$ *strictement* **alors**
3. | | $\mathrm{CP} \leftarrow \mathrm{CP} \setminus \{paire\}$;
4. $\mathrm{CP}_0 \leftarrow \emptyset, \mathrm{CP}_1 \leftarrow \emptyset, \mathrm{CP}_2 \leftarrow \emptyset$;
5. **pour tout** $g \in G$ **faire**
6. | **si** $[LM(f) \wedge LM(g)] = 1$ **alors**
7. | | $\mathrm{CP}_0 \leftarrow \mathrm{CP}_0 \cup \mathrm{paire}(f, g)$
8. | **sinon**
9. | | $\mathrm{CP}_1 \leftarrow \mathrm{CP}_1 \cup \mathrm{paire}(f, g)$;
10. **pour tout** $paire = (lcm, t_1, g_1, t_2, g_2) \in \mathrm{CP}_1$ **faire**
11. | $\mathrm{CP}_1 \leftarrow \mathrm{CP}_1 \setminus \{paire\}$;
12. | **si** $\nexists\ paire' = (lcm', t'_1, g'_1, t'_2, g'_2) \in \mathrm{CP}_{0,1,2}$ *tel que* $lcm' | lcm$ **alors**
13. | | $\mathrm{CP}_2 \leftarrow \mathrm{CP}_2 \cup \{paire\}$;
14. $\mathrm{CP} \leftarrow \mathrm{CP} \cup \mathrm{CP}_2$;
15. **pour tout** $g \in G$ **faire**
16. | **si** $LM(f) | LM(g)$ **alors** $G \leftarrow G \setminus \{g\}$;
17. **si** $\forall g \in G,\ LM(g) \nmid LM(f)$ **alors** $G \leftarrow G \cup \{f\}$;

faire partie du cycle. Toutes les éliminations ont donc été faites en lignes 10 à 13 de l'algorithme 4, ce qui implique que $j_k \geq j_{k+1}$ et donc que $j_k = j_0$ pour tout k. Les paires du cycle ont donc toutes été considérées lors du même appel à la fonction `Update`, qui exclut précisément la construction d'un tel cycle.

Enfin, il est facile de vérifier que les paires supplémentaires qui sont considérées lorsque la purge en ligne 16 de l'algorithme 4 n'est pas faite, sont automatiquement éliminées par le critère 2. En prenant garde de ne pas utiliser les polynômes "purgés" qui pourraient être impliqués dans les divisions successives (et en les remplaçant donc par les polynômes qui ont servi à les éliminer), on voit que ces polynômes n'interviennent pas dans le calcul d'une base de Gröbner minimale.

Exemple 2.1.8. *On considère le calcul de la base de Gröbner pour l'ordre* $grevlex_{x \succ y \succ z \succ t}$ *de l'idéal engendré par les polynômes*

$$\begin{cases} f_1 = x + y + z + t \\ f_2 = xy + yz + zt + tx \\ f_3 = xyz + yzt + ztx + txy \\ f_4 = xyzt - 1 \end{cases}$$

qui correspond au problème classique $Cyclic_4$ *[BF91].*

1. *Après la phase d'initialisation de l'algorithme 3,* $G = \{f_1\}$ *et* $\mathrm{CP} = \{p_1, p_2, p_3\}$ *où* $p_1 = (xy, y, f_1, 1, f_2)$, $p_2 = (xyz, yz, f_1, 1, f_3)$ *et* $p_3 = (xyzt, yzt, f_1, 1, f_4)$. *Aucune paire n'est éliminée par les critères de Buchberger et les trois polynômes* f_2, f_3, f_4 *ont été purgés par* f_1.
2. *Boucle de traitement des paires critiques :*

- $p_1 = (xy, y, f_1, 1, f_2)$ *: on obtient un nouveau polynôme* $f_5 = \overline{S(f_1, f_2)}^G = y^2 + 2yt + t^2$. *La paire de* f_1 *et* f_5 *est éliminée par le critère 1 et on a :* $G = \{f_1, f_5\}$, CP $= \{p_2, p_3\}$.
- $p_2 = (xyz, yz, f_1, 1, f_3)$ *: on obtient un nouveau polynôme* $f_6 = yz^2 + z^2t - yt^2 - t^3$. *La paire de* f_1 *et* f_6 *est éliminée par le critère 1, et on a :* $G = \{f_1, f_5, f_6\}$, CP $= \{p_3, p_4\}$ *où* $p_4 = (y^2z^2, z^2, f_5, y, f_6)$.
- $p_4 = (y^2z^2, z^2, f_5, y, f_6)$ *: on trouve* $\overline{S(f_5, f_6)}^G = 0$. *On a donc* $G = \{f_1, f_5, f_6\}$, CP $= \{p_3\}$.
- $p_3 = (xyzt, yzt, f_1, 1, f_4)$ *: on obtient un nouveau polynôme* $f_7 = yzt^2 + z^2t^2 - yt^3 + zt^3 - t^4 - 1$. *La paire de* f_1 *et* f_7 *est éliminée par le critère 1, et on a* $G = \{f_1, f_5, f_6, f_7\}$, CP $= \{p_5, p_6\}$ *où* $p_5 = (y^2zt^2, zt^2, f_5, y, f_7)$ *et* $p_6 = (yz^2t^2, t^2, f_6, z, f_7)$.
- $p_6 = (yz^2t^2, t^2, f_6, z, f_7)$ *: on obtient un nouveau polynôme* $f_8 = z^3t^2 + z^2t^3 - z - t$. *Les paires de* f_1 *avec* f_8 *et* f_5 *avec* f_8 *sont éliminées par le critère 1, et la paire* $p_7 = (yz^3t^2, zt^2, f_6, y, f_8)$ *est éliminée par le critère 2 via la paire* $p_8 = (yz^3t^2, z^2, f_7, y, f_8)$ *(ainsi que la paire* p_6 *déjà traitée). On a donc* $G = \{f_1, f_5, f_6, f_7, f_8\}$, CP $= \{p_5, p_8\}$.
- $p_5 = (y^2zt^2, t^2, f_5, y, f_7)$ *: on obtient un nouveau polynôme* $f_9 = yt^4 + t^5 - y - t$. *La paire de* f_1 *et* f_9 *est éliminée par le critère 1, et les paires* $p_{10} = (yz^2t^4, t^4, f_6, z^2, f_9)$ *et* $p_{12} = (yz^3t^4, yt^2, f_8, z^3, f_9)$ *sont éliminées par le critère 2 via la paire* $p_{11} = (yzt^4, t^2, f_7, z, f_9)$ *(ainsi que les paires* p_6 *– déjà traitée – et* p_8*). On a donc* $G = \{f_1, f_5, f_6, f_7, f_8, f_9\}$, CP $= \{p_8, p_9, p_{11}\}$ *où* $p_9 = (y^2t^4, t^4, f_5, y, f_9)$.
- $p_{11} = (yzt^4, t^2, f_7, z, f_9)$ *: on obtient un nouveau polynôme* $f_{10} = z^2t^4 + yz - yt + zt - 2t^2$. *Les paires de* f_1 *avec* f_{10} *et* f_5 *avec* f_{10} *sont éliminées par le critère 1. Les paires* $p_{13} = (yz^2t^4, t^4, f_6, y, f_{10})$ *et* $p_{14} = (yz^2t^4, zt^2, f_7, y, f_{10})$ *sont éliminées par le critère 2 via la paire* $p_{16} = (yz^2t^4, z^2, f_9, y, f_{10})$ *(ainsi que les paires* p_{10} *– déjà éliminée – et* p_{11} *– déjà traitée). On a donc* $G = \{f_1, f_5, f_6, f_7, f_8, f_9, f_{10}\}$, CP $= \{p_8, p_9, p_{15}, p_{16}\}$ *où* $p_{15} = (z^3t^4, t^2, f_8, z, f_{10})$.
- *Les paires restantes sont traitées par lcm croissant et se réduisent toutes à zéro par la base courante.*

Sur cet exemple, on a éliminé au total 13 *paires inutiles, mais il reste encore* 5 *réductions à zéro qui n'ont pu être évitées.*

2.2 Lazard

On introduit dans cette section une approche totalement différente de celle de Buchberger, qui se base sur l'élimination gaussienne des matrices de Macaulay pour le calcul de bases de Gröbner.

2.2.1 Matrices de Macaulay

La résolution de systèmes polynomiaux non linéaires se fait principalement avec de l'élimination successive de variables, de façon à se ramener à un système sous forme triangulaire plus facile à résoudre. La théorie des résultants est l'un des premiers outils disponibles pour faire cette élimination : étant donnés deux polynômes $f_1, f_2 \in \mathbb{A}[X]$ à coefficients dans un anneau intègre $\mathbb{A}$, elle permet de construire un élément $g \in \mathbb{A}$ dans le $\mathbb{A}[X]$-module engendré par f_1 et f_2. L'idée de Lazard [Laz83] consiste à généraliser cette construction pour plusieurs polynômes, afin d'éliminer simultanément plusieurs variables.

Avant de présenter l'approche de Lazard, on rappelle comment construire la matrice de Sylvester de deux polynômes. Cette matrice donne de nombreuses informations sur le $\mathbb{A}[X]$-module engendré par les deux polynômes, permettant notamment de calculer leur pgcd et leur résultant.

Définition 2.2.1. *Soient $f_1(X) = a_m X^m + \cdots + a_0$ et $f_2(X) = b_n X^n + \cdots + b_0$ deux polynômes de $\mathbb{A}[X]$ où $\mathbb{A}$ est un anneau intègre quelconque. On définit la* matrice de Sylvester *de f_1 et f_2 par*

$$\mathrm{Syl}(f_1, f_2) = \begin{pmatrix} a_m & a_{m-1} & \cdots & a_0 & & & \\ & a_m & a_{m-1} & \cdots & a_0 & & \\ & & \ddots & \ddots & & \ddots & \\ & & & a_m & a_{m-1} & \cdots & a_0 \\ b_n & b_{n-1} & & \cdots\cdots & & b_0 & \\ & \ddots & \ddots & & & & \ddots \\ & & b_n & b_{n-1} & \cdots\cdots & & b_0 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} n \text{ lignes} \\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} m \text{ lignes} \end{matrix}$$

Le résultant *de f_1 et f_2 est le déterminant de la matrice de Sylvester* $\mathrm{Syl}(f_1, f_2)$*, on le note :*

$$\mathrm{Res}_X(f_1, f_2) = \det(\mathrm{Syl}(f_1, f_2)).$$

La transposée de la matrice de Sylvester définit donc l'application linéaire $\varphi : (u, v) \mapsto uf_1 + vf_2$ pour tout $(u, v) \in \mathbb{A}[X]^2$ tels que $\deg u < \deg f_2$ et $\deg v < \deg f_1$ dans la base des monômes $1, X, \ldots, X^{m+n-1}$. Une propriété classique du résultant permet de caractériser quand deux polynômes ont un facteur commun non trivial dans $\mathbb{K}[X]$ où $\mathbb{K}$ est le corps de fractions de $\mathbb{A}$:

Propriété 2.2.2.
f_1 et f_2 ont un facteur commun non trivial dans $\mathbb{K}[X]$ si et seulement si $\mathrm{Res}_X(f_1, f_2) = 0$.

Démonstration.
Si g est un facteur commun de f_1 et f_2 alors $\varphi(f_2/g, -f_1/g) = 0$, donc $\mathrm{Res}_X(f_1, f_2) = 0$.
Réciproquement, si $\mathrm{Res}_X(f_1, f_2) = 0$ et $f_1 \wedge f_2 = 1$, alors il existe $u, v \in \mathbb{K}[X]$ tels que $\deg u < \deg f_2$ et $\deg v < \deg f_1$ et $uf_1 = -vf_2$; en particulier f_1 divise v, ce qui donne une contradiction. □

Dans le cas où $\mathbb{A}$ est un corps, le pgcd de f_1 et f_2 donné par la relation de Bézout s'obtient comme la combinaison linéaire non nulle minimale (commençant par le plus de zéros) des lignes de cette matrice, et se calcule explicitement à partir de la forme échelon de $\mathrm{Syl}(f_1, f_2)$.

Une propriété intéressante du résultant est d'appartenir au $\mathbb{A}[X]$-module engendré par f_1 et f_2 :

Proposition 2.2.3. *Il existe deux polynômes u et v de $\mathbb{A}[X]$ tels que* $\mathrm{Res}_X(f_1, f_2) = uf_1 + vf_2$.

Démonstration. En remplaçant la dernière colonne c_{m+n} de la matrice de Sylvester par la combinaison linéaire $X^{m+n-1}c_1 + \ldots + Xc_{n+m-1} + c_{m+n}$ des colonnes précédentes, on obtient la matrice

$$\begin{pmatrix} a_m & a_{m-1} & \cdots & a_0 & & & X^{n-1}f_1 \\ & a_m & a_{m-1} & \cdots & a_0 & & X^{n-2}f_1 \\ & & \ddots & \ddots & & \ddots & \vdots \\ & & & a_m & a_{m-1} & \cdots & f_1 \\ b_n & b_{n-1} & & \cdots\cdots & & b_0 & X^{m-1}f_2 \\ & \ddots & \ddots & & & & \vdots \\ & & b_n & b_{n-1} & \cdots\cdots & & f_2 \end{pmatrix}.$$

Le déterminant de cette matrice est égal à celui de la matrice de Sylvester, et en développant par rapport à la dernière colonne, on obtient bien $u, v \in \mathbb{A}[X]$ tels que $\deg u < \deg(f_2)$, $\deg v < \deg(f_1)$ et $\mathrm{Res}_X(f_1, f_2) = uf_1 + vf_2$. □

La matrice de Macaulay de m polynômes $f_1, \ldots, f_m \in \mathbb{K}[X_1, \ldots, X_n]$ est la généralisation naturelle de la matrice de Sylvester. Cette matrice, définie pour un certain degré d, s'obtient en indexant les colonnes par les monômes de degré au plus d en n variables, et en remplissant chaque ligne par les coefficients de tous les multiples des polynômes $f_1, \ldots, f_m$ jusqu'au degré d ; elle contient en particulier $\binom{n+d}{n}$ colonnes. On donne ci-dessous un exemple pour plus de clarté ; celui-ci permet de mettre en évidence une propriété importante des matrices de Macauclay, qui est à la base de l'approche de Lazard : connaissant la forme échelon de la matrice de Macaulay d'un système de polynômes $S = \{f_1, \ldots, f_m\}$ pour un degré assez élevé, on déduit une base de Gröbner de l'idéal I engendré par S. En effet, on sait qu'il existe une base de Gröbner $G = \{g_1, \ldots, g_s\}$ de I et que chacun de ses polynômes g_k s'obtient comme combinaison $\sum_{i=1}^{m} h_{k,i} f_i$ des polynômes de S, avec $h_{i,k} \in \mathbb{K}[X_1, \ldots, X_n]$. En particulier si l'on construit la matrice de Macaulay en prenant tous les multiples des f_i jusqu'au degré au moins $d_{max} = \max_{i=1\ldots m}(\deg(h_{k,i} f_i))$, on obtient à partir de sa forme échelon une base de Gröbner de l'idéal engendré par S. Dans le cas où le système S est formé de polynômes homogènes, le calcul de la forme échelon de la matrice de Macaulay de degré $d < d_{max}$ donne une d-base de Gröbner de l'idéal homogène engendré par S.

Exemple 2.2.4. *On considère le système polynomial défini sur* $\mathbb{F}_7[x, y, z]$ *par* $\{f_1 = x + 2y + 2z - 1;\ f_2 = xy + yz + 3y;\ f_3 = x^2 + 2y^2 + 2z^2 - x\}$*, qui intervient dans le problème classique* $Katsura_3$ *[KFI+87]. La matrice de Macaulay des multiples de* f_1, f_2, f_3 *où on a rangé les monômes en* 3 *variables jusqu'au degré* 3 *par ordre* $grevlex_{x \succ y \succ z}$ *décroissant est :*

	x^3	x^2y	xy^2	y^3	x^2z	xyz	y^2z	xz^2	yz^2	z^3	x^2	xy	y^2	xz	yz	z^2	x	y	z	1
f_1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1	2	2	6
zf_1	·	·	·	·	·	·	·	·	·	·	·	·	·	1	2	2	·	·	6	·
yf_1	·	·	·	·	·	·	·	·	·	·	·	1	2	·	2	·	·	6	·	·
xf_1	·	·	·	·	·	·	·	·	·	·	1	2	·	2	·	·	6	·	·	·
z^2f_1	·	·	·	·	·	·	·	1	2	2	·	·	·	·	·	6	·	·	·	·
yzf_1	·	·	·	·	·	1	2	·	2	·	·	·	·	·	6	·	·	·	·	·
xzf_1	·	·	·	·	1	2	·	2	·	·	·	·	·	6	·	·	·	·	·	·
y^2f_1	·	·	1	2	·	·	2	·	·	·	·	·	6	·	·	·	·	·	·	·
xyf_1	·	1	2	·	·	2	·	·	·	·	·	6	·	·	·	·	·	·	·	·
x^2f_1	1	2	·	·	2	·	·	·	·	·	6	·	·	·	·	·	·	·	·	·
f_2	·	·	·	·	·	·	·	·	·	·	·	1	·	·	1	·	·	3	·	·
zf_2	·	·	·	·	·	1	·	·	1	·	·	·	·	·	3	·	·	·	·	·
yf_2	·	·	1	·	·	·	1	·	·	·	·	·	3	·	·	·	·	·	·	·
xf_2	·	1	·	·	·	1	·	·	·	·	·	3	·	·	·	·	·	·	·	·
f_3	·	·	·	·	·	·	·	·	·	·	1	·	2	·	·	2	6	·	·	·
zf_3	·	·	·	·	1	·	2	·	·	2	·	·	·	6	·	·	·	·	·	·
yf_3	·	1	·	2	·	·	·	·	2	·	·	6	·	·	·	·	·	·	·	·
xf_3	1	·	2	·	·	·	·	2	·	·	6	·	·	·	·	·	·	·	·	·

La forme réduite de la matrice de Macaulay est :

x^3	x^2y	xy^2	y^3	x^2z	xyz	y^2z	xz^2	yz^2	z^3	x^2	xy	y^2	xz	yz	z^2	x	y	z	1
1	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	2	6
·	1	·	·	·	·	·	·	·	4	·	·	·	·	·	·	·	6	5	·
·	·	1	·	·	·	·	·	·	2	·	·	·	·	·	·	·	3	6	·
·	·	·	1	·	·	·	·	·	1	·	·	·	·	·	·	·	5	3	·
·	·	·	·	1	·	·	·	·	6	·	·	·	·	·	·	·	5	·	·
·	·	·	·	·	1	·	·	·	3	·	·	·	·	·	·	·	5	2	·
·	·	·	·	·	·	1	·	·	5	·	·	·	·	·	·	·	6	1	·
·	·	·	·	·	·	·	1	·	1	·	·	·	·	·	·	·	1	6	·
·	·	·	·	·	·	·	·	1	4	·	·	·	·	·	·	·	3	5	·
·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	1	5	6
·	·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	·	1	·	·
·	·	·	·	·	·	·	·	·	·	·	·	**1**	·	·	·	·	**4**	·	·
·	·	·	·	·	·	·	·	·	·	·	·	·	1	·	·	·	3	2	·
·	·	·	·	·	·	·	·	·	·	·	·	·	·	**1**	·	·	**2**	·	·
·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	**1**	·	·	**2**	·
·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	**1**	**2**	**2**	**6**
·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·

Les polynômes associés aux lignes de cette matrice forment une base de Gröbner pour l'ordre $grevlex_{x \succ y \succ z}$ *de l'idéal engendré par* $\{f_1, f_2, f_3\}$*, dont une base minimale est* $\{x + 2y + 2z - 1,\ z^2 + 2z,\ yz + 2y,\ y^2 + 4y\}$ *(correspondant aux polynômes associés aux lignes en gras).*

2.2.2 Degré de régularité et complexité

Pour calculer une base de Gröbner d'un idéal avec la méthode de Lazard, la principale difficulté est donc de savoir jusqu'à quel degré calculer la matrice de Macaulay correspondante. On donne dans cette section des bornes pour ce degré dans le cadre général, ainsi qu'une analyse de la complexité de la méthode.

On commence par étudier la complexité des calculs de bases de Gröbner d'idéaux homogènes, plus facile à appréhender que dans le cas affine. Soit I un idéal homogène engendré par les polynômes homogènes $f_1, \ldots, f_m \in \mathbb{K}[X_0, \ldots, X_n]$. On constate déjà que dans le calcul de la forme échelon d'une matrice de Macaulay construite à partir des multiples de $f_1, \ldots, f_m$, les polynômes de degrés différents n'interagissent pas. On peut donc travailler avec les matrices de Macaulay en chaque degré indépendamment. Par ailleurs, on sait qu'il existe une base de Gröbner $G = \{g_1, \ldots, g_s\}$ minimale de I formée de polynômes homogènes ; notons $d_1, \ldots, d_s$ les degrés respectifs de ces polynômes. Il suffit alors pour obtenir cette base, de réduire les matrices de Macaulay jusqu'au degré $d = \max_{i=1\ldots s} d_i$. Sous certaines hypothèses, ce degré est en fait borné supérieurement par le degré de régularité de l'idéal. On introduit ici les notions nécessaires pour fixer ces hypothèses.

On rappelle que pour un idéal $I \subset \mathbb{K}[X_0, \ldots, X_n]$ et un polynôme $f \in \mathbb{K}[X_0, \ldots, X_n]$, l'idéal quotient de I par f est défini par $(I : f) = \{g \in \mathbb{K}[X_0, \ldots, X_n] : gf \in I\}$.

Définition 2.2.5. *Une suite de polynômes homogènes* $(f_1, \ldots, f_m)$ *est appelée* suite régulière *si*

(i) $\langle f_1, \ldots, f_m \rangle \neq \mathbb{K}[X_0, \ldots, X_n]$,

(ii) pour tout $i \in [\![1; m]\!]$, $(\langle f_1, \ldots, f_{i-1} \rangle : f_i) = \langle f_1, \ldots, f_{i-1} \rangle$.

La deuxième condition revient à dire que l'image de f_i dans l'anneau quotient $\mathbb{K}[X_0,\ldots,X_n]/\langle f_1,\ldots,f_{i-1}\rangle$ n'est pas un diviseur de 0. D'un point de vue géométrique, cela signifie que l'idéal homogène engendré par $f_1,\ldots,f_i$ est de dimension $n-i$, ou autrement dit, à chaque fois que l'on ajoute un polynôme de la suite dans l'idéal, celui-ci "perd une dimension".

Les suites régulières sont fondamentales, puisqu'elles modélisent le comportement "standard" d'un système qui n'est pas surdéterminé (i.e. lorsque $m \leq n$). Plus précisément, l'ensemble des suites régulières de polynômes forme un ouvert dense de Zariski dans l'ensemble des suites de $m \leq n$ polynômes de degrés fixés, et tout idéal admet un système de générateurs qui forme une suite régulière. Pour ce type de systèmes, on peut borner a priori de façon assez précise, le degré des éléments de la base de Gröbner en fonction des degrés des polynômes de la suite régulière définissant l'idéal.

Proposition 2.2.6 (Lazard[Laz83], Giusti[Giu84])**.**
Soit I un idéal homogène engendré par une suite régulière $(f_1,\ldots,f_m)$. Le degré de régularité $d_{reg}(I)$ de l'idéal I est majoré par la borne de Macaulay *:*

$$d_{reg}(I) \leq \sum_{i=1}^{m}(\deg(f_i)-1)+1.$$

De plus, pour presque tout changement de coordonnées, le degré maximal des éléments d'une base de Gröbner est majoré par le degré de régularité de l'idéal.

Il est à noter que cette proposition peut également s'appliquer au cas des systèmes surdéterminés (et des suites *semi-régulières*, voir section 2.4.3), mais qu'en pratique le degré maximal atteint pour ces systèmes est nettement inférieur. Par exemple, si l'on considère un système homogène "générique" composé de $m+1$ polynômes où m est le nombre de variables, le degré maximal atteint est en $\left(\sum_{i=1}^{m+1}(\deg(f_i)-1)+1\right)/2$; on renvoie à [Bar04] pour plus de détails.

Connaissant le degré maximal atteint durant le calcul, il est maintenant possible d'estimer la complexité de l'algorithme de Lazard pour le calcul d'une base de Gröbner d'un idéal homogène engendré par $f_1,\ldots,f_m \in \mathbb{K}[X_0,\ldots,X_n]$ de degrés respectifs $d_1,\ldots,d_m$. On note ω la constante intervenant dans la complexité du produit matriciel ; plus précisément, la complexité du produit de deux matrices de taille n est en $O(n^\omega)$ opérations dans $\mathbb{K}$, où ω dépend de l'algorithme utilisé (ω est égal à 3 pour le produit matriciel classique, et peut être aussi bas que 2.36 en utilisant la méthode de Coppersmith-Winograd [CW90]). La complexité du calcul de la forme échelon d'une matrice quelconque de taille $\ell \times c$ est alors en $O(\ell^{\omega-1}c)$ opérations. Pour construire la matrice de Macaulay de degré d_{reg}, on considère tous les monômes en $n+1$ variables de degré maximal d_{reg}, soit $c = \binom{d_{reg}+n+1}{d_{reg}}$ colonnes, ainsi que tous les multiples des polynômes f_i jusqu'au degré d_{reg}, soit $\ell = \sum_{i=1}^{m}\binom{d_{reg}-d_i+n+1}{d_{reg}-d_i}$ lignes ; il est alors facile de donner une borne sur la complexité du calcul de sa forme échelon en fonction de d_{reg} en terme d'opérations dans le corps de base $\mathbb{K}$. Si l'arithmétique sur $\mathbb{K}$ ne s'effectue pas en temps constant (typiquement si $\mathbb{K} = \mathbb{Q}$ ou $\mathbb{F}_p(T)$), l'analyse complète de la complexité est plus compliquée puisqu'il faut prendre en compte la croissance, en général exponentielle, des coefficients [1]. Dans cette thèse, on s'intéresse principalement au cas des corps finis qui ne soulève pas cette difficulté.

Cependant cette borne sur la complexité est en général assez mauvaise, dans la mesure où l'on ne prend pas en compte la forme spéciale de la matrice de Macaulay. En effet, cette dernière est

1. À ce sujet, si $\mathbb{K}$ est un corps de nombres ou de fonctions, il est souvent avantageux de se ramener au cas des corps finis en calculant les bases de Gröbner modulo des nombres premiers (ou des polynômes irréductibles) distincts, puis d'utiliser un théorème type restes chinois (voir par exemple [Arn03], cf aussi Chapitre 3).

en général relativement creuse car les polynômes étant homogènes, seuls les monômes d'un certain degré apparaissent sur chacune des lignes. Elle est de plus assez structurée, vu que l'on retrouve sur chaque ligne uniquement les multiples des polynômes de départ. Enfin, la plupart de ses lignes sont redondantes (et donc vont produire des réductions à zéro), voire carrément inutiles : si l'on reprend l'exemple 2.2.4, on voit à partir de la relation triviale $f_1 f_3 = f_3 f_1$ que la dernière ligne $x f_3$ est redondante, et que par conséquent la ligne $x^2 f_1$ est inutile puisque c'est la seule ligne ayant pour terme de tête le plus grand monôme x^3. Il n'est cependant pas évident de comprendre comment exploiter ces particularités pour améliorer l'étude de la complexité. Du point de vue algorithmique par contre, on verra dans la suite de cette thèse comment il est possible de raffiner l'approche de Lazard pour donner des algorithmes plus performants. L'analyse de la complexité de ces algorithmes cependant restera *in fine* très similaire.

La situation pour les systèmes non homogènes est plus complexe. On se convainc avec l'exemple suivant que le degré de régularité de l'idéal n'est plus une borne convenable pour le degré des polynômes intervenant dans le calcul de la base de Gröbner.

Exemple 2.2.7. *Soient* $f_1 = X^2Y^2 + 2X^2Y + X^2 + XY + X + 1$ *et* $f_2 = X^3Y^3 + 3X^3Y^2 + 3X^3Y + X^3 + XY + X + 1$ *deux polynômes de* $\mathbb{Q}[X, Y]$. *On a* $\langle f_1, f_2 \rangle = \langle 1 \rangle$, *donc le degré de régularité de l'idéal* I *engendré par* $\{f_1, f_2\}$ *est* $d_{reg} = 0$. *Pourtant, le calcul de la base de Gröbner réduite de* I *pour l'ordre* $grevlex_{X>Y}$ *nécessite la construction de la matrice de Macaulay jusqu'au degré* 8.

Il n'est donc plus possible dans le cas affine d'utiliser le degré de régularité de l'idéal pour minorer le degré de la matrice de Macaulay nécessaire au calcul de la base de Gröbner. On peut toutefois donner une borne sur la complexité du calcul en affine, en se ramenant au cas homogène. On introduit à cet effet le degré de régularité d'un système affine, à ne pas confondre avec le degré de régularité de l'idéal introduit en définition 1.3.3.

Définition 2.2.8. *On appelle* degré de régularité d'un système affine $f_1, \ldots, f_m \in \mathbb{K}[X_1, \ldots, X_n]$ *le degré de régularité de l'idéal constitué des polynômes homogénéisés* $f_1^h, \ldots, f_m^h \in \mathbb{K}[X_0, \ldots, X_n]$.

On déduit alors facilement de la proposition 1.1.20 que le degré atteint durant le calcul de la base de Gröbner du système affine avec la méthode de Lazard, ne dépasse pas celui atteint pour le système homogénéisé correspondant. Sous les hypothèses données en proposition 2.2.6, il est alors possible de borner comme précédemment la complexité du calcul en affine en fonction du degré de régularité du système.

2.3 Algorithme F4

On présente dans cette section l'algorithme F4 de Faugère [Fau99], qui est une implantation efficace de l'algorithme de Buchberger reprenant les idées de Lazard pour accélérer le calcul des réductions des paires critiques. On rappelle dans une première partie les deux idées essentielles de cet algorithme, puis on détaille les choix faits pour notre implantation de F4 en langage C.

2.3.1 Principes

L'essentiel du temps de calcul de l'algorithme de Buchberger étant concentré sur la réduction des paires critiques, il devient primordial d'optimiser ces réductions. On peut à cet effet agir sur deux plans : d'une part en essayant d'éliminer a priori un maximum de paires inutiles (avec les critères

de Buchberger par exemple ou le critère F5 que l'on présentera en section suivante), d'autre part en optimisant directement les calculs de réductions. C'est cette dernière approche qui est développée dans l'algorithme F4.

La première "idée-clé" de l'algorithme consiste à utiliser – à la façon de Lazard – l'algèbre linéaire pour réduire simultanément plusieurs paires. On ne traite plus une seule paire critique à la fois, comme dans l'algorithme de Buchberger, mais un sous-ensemble de paires $\{(lcm, u_1, f_1, u_2, f_2)_i\}_{i\in I}$, obtenu par exemple en sélectionnant toutes celles de degré minimal en attente de traitement. Cette stratégie de sélection, qui correspond à une généralisation de la *stratégie normale* présentée en section 2.1.2, est en effet la plus couramment adoptée dans les calculs de bases de Gröbner pour l'ordre *grevlex*, pour lequel F4 est particulièrement efficace (on renvoie à la section 2.5 pour les calculs de bases de Gröbner avec d'autres ordres). Une fois les paires sélectionnées, on construit une matrice de type Macaulay contenant tous les produits $u_i f_i$ provenant des paires, ainsi que les multiples des polynômes de la base courante G permettant de réduire les queues de ces polynômes $u_i f_i$ (voir procédure `Preprocessing`). Le calcul de la forme échelon de cette matrice donne alors la réduction simultanée de toutes les paires critiques sélectionnées. Les nouveaux polynômes de la base s'obtiennent ensuite en sélectionnant les lignes de la matrice correspondant aux polynômes dont les termes de tête ne sont pas divisibles par les monômes de $LT(G)$. La mise à jour de la liste des paires critiques et de la base courante est identique à celle de l'algorithme original de Buchberger (voir procédure `Postprocessing`). Le deuxième point essentiel dans l'algorithme F4 consiste à mémoriser les réductions des multiples calculées à chaque étape pour accélérer les réductions suivantes : chaque produit uf d'un monôme u avec un polynôme f est remplacé par un produit de la forme $\frac{LM(uf)}{LM(f')}f'$, où f' est un polynôme qui a été obtenu dans les matrices réduites précédentes et tel que $LM(f')|\, LM(uf)$. Avec cette simplification, on ajoute donc seulement dans la matrice des polynômes dont les queues sont déjà partiellement réduites. Ceci a pour conséquence de diminuer notablement le nombre de lignes de la matrice (moins de polynômes à ajouter pour réduire les queues) ainsi que le nombre de réductions nécessaires. Cette réutilisation des calculs est assurée par la procédure `Simplify` donnée en section 2.3.2.

On illustre le fonctionnement de cet algorithme sur le système $Katsura_3$ déjà introduit en exemple 2.2.4.

Exemple 2.3.1. *Soit* $I = \langle f_1, f_2, f_3\rangle \subset \mathbb{F}_7[x, y, z]$ *où* $f_1 = x + 2y + 2z - 1$; $f_2 = xy + yz + 3y$; $f_3 = x^2 + 2y^2 + 2z^2 - x$. *On calcule une base de Gröbner de* I *pour* $grevlex_{x\succ y\succ z}$ *avec F4.*

1. *Après la phase d'initialisation (similaire à celle de l'algorithme de Buchberger),* $G = \{f_1\}$ *et* $\mathrm{CP} = \{p_1, p_2\}$ *où* $p_1 = (xy, y, f_1, 1, f_2)$ *et* $p_2 = (x^2, x, f_1, 1, f_3)$. *Aucune paire n'est éliminée par les critères de Buchberger et les polynômes* f_2 *et* f_3 *ont été purgés par* f_1.
2. *Boucle de traitement des paires critiques :*

   - *On réduit simultanément* p_1 *et* p_2. *On crée la matrice contenant les quatre multiples provenant des paires, ainsi que les polynômes* zf_1 *et* f_1 *(preprocessing), puis on calcule sa forme échelon.*

$$
\begin{array}{c}
\\ xf_1 \\ f_3 \\ yf_1 \\ f_2 \\ zf_1 \\ f_1
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} x^2 & xy & y^2 & xz & yz & z^2 & x & y & z & 1 \end{array} \\
\begin{pmatrix}
1 & 2 & \cdot & 2 & \cdot & \cdot & 6 & \cdot & \cdot & \cdot \\
1 & \cdot & 2 & \cdot & \cdot & 2 & 6 & \cdot & \cdot & \cdot \\
\cdot & 1 & 2 & \cdot & 2 & \cdot & \cdot & 6 & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot & 3 & \cdot & \cdot \\
\cdot & \cdot & \cdot & 1 & 2 & 2 & \cdot & \cdot & 6 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 2 & 2 & 6
\end{pmatrix}
\end{array}
\longrightarrow
\begin{array}{c}
\begin{array}{cccccccccc} x^2 & xy & y^2 & xz & yz & z^2 & x & y & z & 1 \end{array} \\
\begin{pmatrix}
1 & \cdot & \cdot & \cdot & \cdot & 6 & \cdot & 1 & 3 & 6 \\
\cdot & 1 & \cdot & \cdot & \cdot & 3 & \cdot & 1 & 6 & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & 5 & \cdot & 4 & 3 & \cdot \\
\cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & 3 & 4 & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & 4 & \cdot & 2 & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 2 & 2 & 6
\end{pmatrix}
\end{array}
$$

On obtient deux nouveaux polynômes (correspondant aux nouveaux termes de tête) $f_4 = y^2 + 5z^2 + 4y + 3z$ *et* $f_5 = yz + 4z^2 + 2y + z$*. Les paires de* f_1 *avec* f_4 *et* f_1 *avec* f_5 *sont éliminées par le critère 1, et on a* $G = \{f_1, f_4, f_5\}$, $\mathrm{CP} = \{p_3\}$ *où* $p_3 = (y^2z, z, f_4, y, f_5)$.

- *On réduit* p_3 *: on crée la matrice contenant les deux multiples provenant de* p_3*, ainsi que les polynômes* zf_5*,* f_4 *et* f_5 *(preprocessing) puis on calcule la forme échelon.*

$$\begin{array}{c} \\ zf_4 \\ yf_5 \\ zf_5 \\ f_4 \\ f_5 \end{array}\!\!\begin{array}{c} \begin{array}{cccccccc} y^2z & yz^2 & z^3 & y^2 & yz & z^2 & y & z \end{array} \\ \begin{pmatrix} 1 & \cdot & 5 & \cdot & 4 & 3 & \cdot & \cdot \\ 1 & 4 & \cdot & 2 & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & 4 & \cdot & 2 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & 5 & 4 & 3 \\ \cdot & \cdot & \cdot & \cdot & 1 & 4 & 2 & 1 \end{pmatrix} \end{array} \longrightarrow \begin{array}{c} \begin{array}{cccccccc} y^2z & yz^2 & z^3 & y^2 & yz & z^2 & y & z \end{array} \\ \begin{pmatrix} 1 & \cdot & 5 & \cdot & \cdot & \cdot & 6 & 1 \\ \cdot & 1 & 4 & \cdot & \cdot & \cdot & 3 & 5 \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 4 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 2 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 2 \end{pmatrix} \end{array}$$

On obtient un nouveau polynôme $f_6 = z^2 + 2z$*. Les paires de* f_1 *avec* f_6 *et* f_4 *avec* f_6 *sont éliminées par le critère 1. On a* $G = \{f_1, f_4, f_5, f_6\}$, $\mathrm{CP} = \{p_4\}$ *où* $p_4 = (yz^2, z, f_5, y, f_6)$.

- *On réduit* p_4*. Les deux multiples provenant de* p_4 *sont* zf_5 *et* yf_6*. On remarque que le multiple* zf_5*, de terme de tête* yz^2*, peut être remplacé par le polynôme correspondant à la deuxième ligne de la matrice réduite précédente* $\widetilde{zf_5} = yz^2 + 4z^3 + 3y + 5z$*. Pour le preprocessing, on remarque aussi que le polynôme* f_5*, de terme de tête* yz*, peut être remplacé par le polynôme correspondant à la quatrième ligne de la matrice réduite précédente* $\widetilde{f_5} = yz + 2y$*.*

$$\begin{array}{c} \\ \widetilde{zf_5} \\ yf_6 \\ zf_6 \\ \widetilde{f_5} \\ f_6 \end{array}\!\!\begin{array}{c} \begin{array}{cccccc} yz^2 & z^3 & yz & z^2 & y & z \end{array} \\ \begin{pmatrix} 1 & 4 & \cdot & \cdot & 3 & 5 \\ 1 & \cdot & 2 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 2 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & 2 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & 2 \end{pmatrix} \end{array} \longrightarrow \begin{array}{c} \begin{array}{cccccc} yz^2 & z^3 & yz & z^2 & y & z \end{array} \\ \begin{pmatrix} 1 & \cdot & \cdot & \cdot & 3 & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & 3 \\ \cdot & \cdot & 1 & \cdot & 2 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & 2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \end{array}$$

On obtient une réduction à zéro et aucun nouveau générateur. Il n'y a plus de paires à traiter, la base de Gröbner de I *est donc* $G = \{f_1, f_4, f_5, f_6\}$*. On remarque que* f_4 *et* f_5 *peuvent être remplacés respectivement par* $\widetilde{f_4} = y^2 + 4y$ *et* $\widetilde{f_5} = yz + 2y$ *provenant de la deuxième matrice réduite. La base obtenue* $G = \{f_1, \widetilde{f_4}, \widetilde{f_5}, f_6\}$ *est alors une base réduite.*

Par rapport à la méthode de Lazard, on constate déjà sur cet exemple que les matrices à réduire sont beaucoup plus petites. De plus, il n'est pas nécessaire de connaître a priori une borne sur le degré à atteindre durant le calcul.

La complexité de l'algorithme F4 est délicate à estimer mais peut être partiellement reliée à celle de l'algorithme de Lazard. En effet, réduire les paires paquets par paquets revient à calculer des formes échelon de sous-matrices de la matrice de Macaulay. En ce sens, l'algorithme F4 exploite en partie la structuration de la matrice de Macaulay. En particulier, la borne supérieure donnée en section précédente s'applique encore à l'algorithme F4.

2.3.2 Pseudo-code

Algorithme 5: Algorithme F4

Entrées : $I = \langle f_1, \dots, f_r \rangle \subset \mathbb{K}[X_1, \dots, X_n]$
Sortie : G base de Gröbner minimale de I

1. $G \leftarrow [\,]$, CP $\leftarrow \emptyset$, $d \leftarrow 0$;
2. **pour** $i = 1$ *à* r **faire** `Update`(f_i);
3. **tant que** CP $\neq \emptyset$ **faire**
4. $\quad \mathrm{CP}_{sel} \leftarrow$ `Sel`(CP) ; CP $\leftarrow$ CP $\setminus$ CP_{sel};
5. $\quad F_d \leftarrow [\,]$, $LM(F_d) \leftarrow \emptyset$, $T(F_d) \leftarrow \emptyset$;
6. $\quad$ **pour tout** $pair = (lcm, t_1, g_1, t_2, g_2) \in \mathrm{CP}_{sel}$ **faire**
7. $\quad\quad$ **pour** $k = 1$ *à* 2 **faire**
8. $\quad\quad\quad f \leftarrow$`Simplify`$(t_k, g_k, (F_i, F'_i)_{0 \le i < d})$;
9. $\quad\quad\quad$ Apposer(F_d, f);
10. $\quad\quad\quad LM(F_d) \leftarrow LM(F_d) \cup \{LM(f)\}$;
11. $\quad\quad\quad T(F_d) \leftarrow T(F_d) \cup \{\text{monômes de } f\}$;
12. $\quad$ `Preprocessing`$(F_d, T(F_d), LM(F_d))$;
13. $\quad M \leftarrow$ matrice dont les lignes sont les polynômes de F_d;
14. $\quad M' \leftarrow$ forme échelon réduite de M;
15. $\quad F'_d \leftarrow$ polynômes associés aux lignes de M';
16. $\quad$ `Postprocessing`$(F'_d, LM(F_d))$;
17. $\quad d \leftarrow d + 1$;
18. **retourner** G;

Algorithme 6: Postprocessing

Entrées : F_d, $LM(F_d)$

1. **pour** $i = 1$ *à* $\#F_d$ **faire**
2. $\quad f \leftarrow F_d[i]$;
3. $\quad$ **si** $f = 0$ **alors break**;
4. $\quad$ **si** $LM(f) \notin LM(F_d)$ **alors** `Update` (f);

Algorithme 7: Simplify

Entrées : t monôme, $g \in \mathbb{K}[\underline{X}]$
Sortie : tg simplifié

1. **pour** m monôme, $m|t$ **faire**
2. $\quad$ **si** $\exists i \in [\![0; d-1]\!]$, $mg \in F_i$ **alors**
3. $\quad\quad f \leftarrow$ (unique) polynôme de F'_i tel que $LM(f) = LM(mg)$;
4. $\quad\quad$ **si** $m = t$ **alors**
5. $\quad\quad\quad$ **retourner** f;
6. $\quad\quad$ **sinon**
7. $\quad\quad\quad$ **retourner** `Simplify`$(\frac{t}{m}, f)$;
8. **retourner** g;

Algorithme 8: Preprocessing

Entrées : $F_d, T(F_d), LM(F_d)$

1. $Done \leftarrow LM(F_d)$;
2. **tant que** $T(F_d) \neq Done$ **faire**
3. $\quad m \leftarrow \max(T(F_d) \setminus Done)$;
4. $\quad Done \leftarrow Done \cup \{m\}$;
5. $\quad$ **pour tous les** $g \in G$ **faire**
6. $\quad\quad$ **si** $LM(g)|m$ **alors**
7. $\quad\quad\quad g' \leftarrow$ `Simplify`$(\frac{m}{LM(g)}, g)$;
8. $\quad\quad\quad$ Apposer(F_d, g');
9. $\quad\quad\quad LM(F_d) \leftarrow LM(F_d) \cup \{m\}$;
10. $\quad\quad\quad T(F_d) \leftarrow T(F_d) \cup \{\text{monômes de } g'\}$;
11. $\quad\quad\quad$ **break**;

Dans ce pseudo-code, on suppose certaines variables globales :

- G est la liste des polynômes qui forment une base de l'idéal I ;
- CP est la liste des paires critiques en attente de traitement ;
- d numérote l'étape courante ;
- F_i est la liste des polynômes correspondant aux lignes de la matrice à échelonner à la i-ème étape, et F'_i les polynômes de la matrice échelonnée.

La fonction `Update` a déjà été décrite en section 2.1 dans l'algorithme 4. La fonction `Sel` en ligne 4 définit la stratégie de sélection des paires. À l'étape d, on crée la liste F_d des polynômes à insérer dans la matrices M, avec la liste des termes $T(F_d)$ correspondant aux colonnes de M.

La fonction `Simplify` est présentée telle que dans [Fau99] ; cette fonction cherche s'il est possible de remplacer un multiple tg par un polynôme plus réduit obtenu à une étape précédente. Clairement l'implantation de `Simplify` est problématique : la quantité d'information à stocker est importante, ce qui rend d'autant plus difficile la recherche du polynôme qui simplifie tg. On remarque d'ailleurs qu'il est inutile de mémoriser les F_i, il suffit de retenir à la place quels produits ont été effectués.

Par ailleurs, il n'y a en général pas unicité de la simplification et savoir quel polynôme conviendrait le mieux n'est pas facile. Par exemple, si l'on connaît une simplification de xf et de yf, on ne peut pas décider a priori laquelle choisir pour simplifier xyf. De même, si l'on a obtenu une simplification de x^2f au début de l'algorithme, ainsi qu'une simplification plus récente de xf, il peut être plus avantageux d'utiliser cette dernière pour recalculer x^2f. On détaillera dans la section suivante la solution adoptée pour notre implantation de la fonction `Simplify`.

2.3.3 Choix d'implantation

On ne considèrera ici que des calculs de bases de Gröbner de systèmes à coefficients dans des corps finis, les cas plus généraux n'ayant pas été implantés.

La principale difficulté liée à l'implantation de l'algorithme F4 concerne l'algèbre linéaire. Les matrices à réduire peuvent en effet avoir des tailles très différentes suivant le nombre de variables et le degré atteint durant le calcul, et être plus ou moins creuses suivant les particularités des systèmes considérés et le corps choisi.

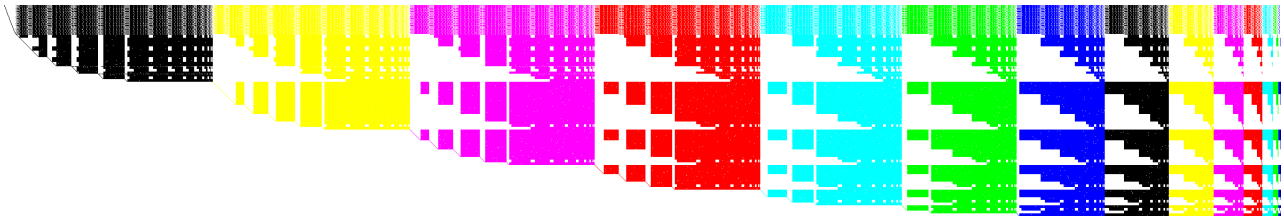


FIGURE 2.1 – Matrice de taille 393×2391 obtenue à l'étape 7 du calcul de la base de Gröbner du système de la section 3.3.1 (les couleurs permettent de distinguer les monômes de même degré modulo 7).

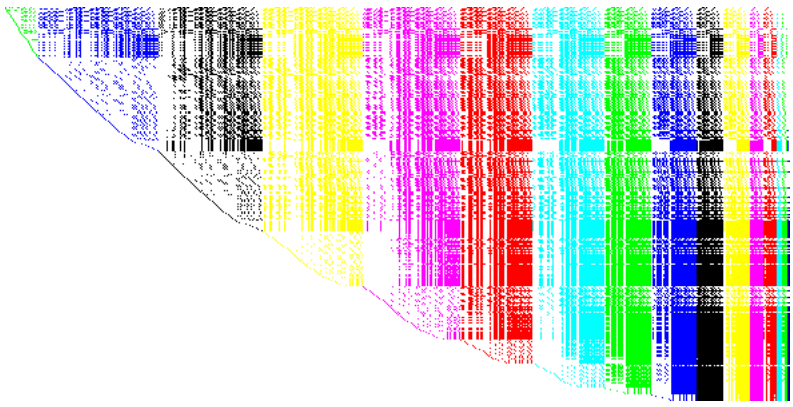
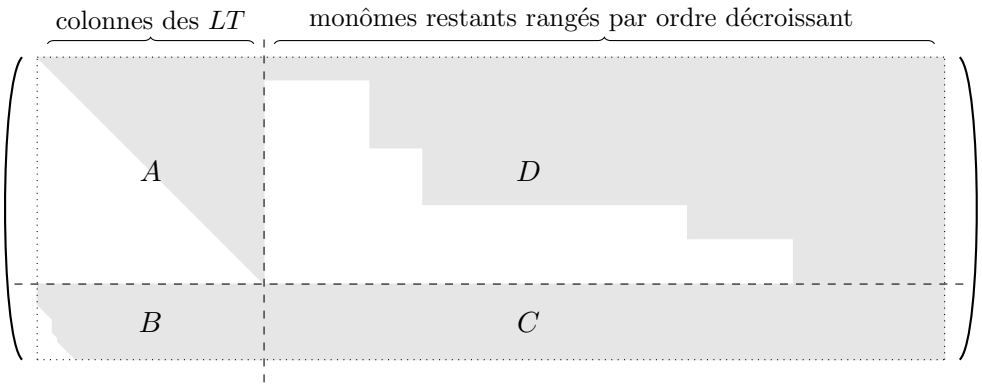


FIGURE 2.2 – Matrice de taille 295×588 obtenue à l'étape 5 du calcul de la base de Gröbner du système de la section 7.3.4 en caractéristique 2.

Néanmoins, comme on peut le voir sur les exemples ci-dessus, les matrices considérées ont la particularité commune d'être quasiment échelonnées inférieurement. Par ailleurs, si l'on effectue une élimination gaussienne pour obtenir la forme échelon, on peut facilement localiser a priori la plupart des pivots à partir des termes de tête des polynômes. Notre approche consiste donc à modifier la construction de la matrice originale (type Macaulay) de façon à obtenir une matrice de la forme :



où l'on a fait des échanges de lignes et de colonnes de telle sorte que les premières colonnes correspondent maintenant aux monômes (par ordre décroissant) qui sont des termes de tête, les colonnes restantes étant laissées dans l'ordre décroissant. Les parties non grisées dans la matrice correspondent aux zéros que l'on peut localiser facilement à la construction. La propriété d'être quasiment sous forme échelon inférieur se traduit par le fait que le nombre de lignes de B est en général petit par rapport à celui de A. De plus, on observe que la matrice A contient le plus souvent peu de coefficients non nuls dans sa partie triangulaire supérieure, ce qui est la conséquence directe de l'utilisation de la procédure `Simplify`. On ne peut cependant pas considérer la matrice A comme creuse dans le cas général.

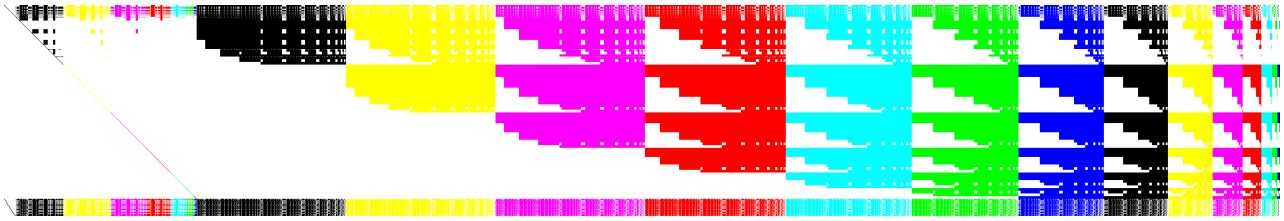


FIGURE 2.3 – Matrice de la figure 2.1 réarrangée.

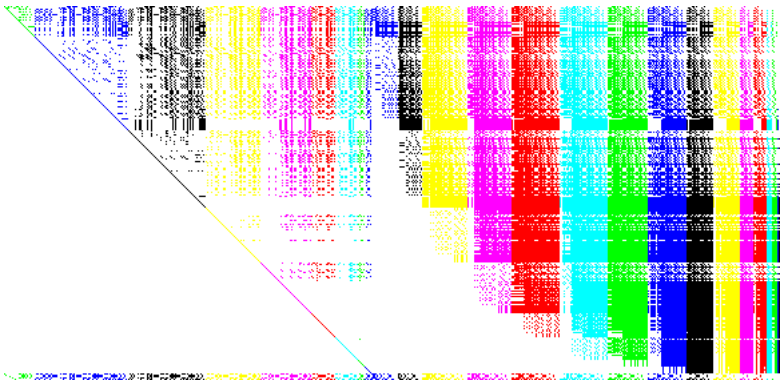


FIGURE 2.4 – Matrice de la figure 2.2 réarrangée.

L'opération fondamentale dans l'élimination gaussienne consiste à ajouter à une ligne de la matrice, un multiple de la ligne contenant le pivot. Pour améliorer les performances, il est crucial d'accélérer cette opération ; une possibilité consiste à "vectoriser" les données, ce qui peut se faire à l'aide des instructions SIMD disponibles sur la plupart des processeurs actuels. On rappelle rapidement le principe : une instruction SIMD (Single Instruction, Multiple Data) permet d'appliquer simultanément la même opération à toutes les composantes d'un vecteur de données. Dans notre

situation, cela permet de faire l'opération sur les lignes par bloc de coefficients, le nombre de coefficients par bloc dépendant de la taille du corps. On utilise dans les applications essentiellement deux types de corps.

- Sur $\mathbb{F}_p$ (p premier) : les opérations d'addition et de multiplication dans $\mathbb{Z}$ sont prises en charge par les instructions SIMD, mais pas la réduction modulaire. Par ailleurs, une telle réduction est surtout coûteuse après un produit. Pour minimiser le nombre de réductions post-produit, on représente les éléments de $\mathbb{F}_p$ par des entiers entre $[\![-p^2/4; p^2/4]\!]$ et à chaque étape de l'élimination, on ramène dans $[\![-p/2; p/2]\!]$ uniquement les coefficients intervenant dans les produits (i.e. les éléments de la ligne et de la colonne du pivot). Une réduction finale ramène tous les coefficients de la matrice dans l'intervalle $[\![-p/2; p/2]\!]$.
- Sur $\mathbb{F}_{2^n}$: on utilise une base de la forme $1, t, t^2, \ldots, t^{n-1}$, et on représente un élément par la suite de bits correspondant aux coefficients. L'addition revient alors à faire un xor bit à bit, ce qui est bien sûr compatible SIMD. La multiplication est par contre beaucoup plus complexe, et la façon la plus simple d'accélérer les calculs est de tabuler à l'avance un certain nombre de produits.

D'un autre côté, il est connu que l'élimination gaussienne, dont la complexité est proportionnelle au cube de la taille de la matrice, n'est pas l'algorithme le plus rapide pour obtenir une forme échelon. On a vu en section 2.2.2 qu'en utilisant des techniques de multiplication rapide de matrices, la complexité théorique du calcul de la forme échelon est en la taille de la matrice à la puissance ω, avec $\omega = \log_2(7)$ pour l'algorithme de Strassen [Str69]. Il est alors tentant d'utiliser cette méthode pour accélérer les calculs dans F4. Cependant, ces complexités sont optimales pour des matrices très pleines, alors que les matrices construites par F4 sont déjà sous forme presque échelonnée inférieurement et la localisation de nombreux zéros est possible. Dans ces conditions, il devient difficile d'adapter un algorithme type Strassen pour qu'il prenne en compte ces spécificités ; expérimentalement, les temps obtenus avec les choix d'implantation présentés ici sont nettement meilleurs.

Concernant la réutilisation des réductions précédentes avec `Simplify`, on a choisi de stocker dans un arbre les simplifications des produits de la forme mf, où m est un monôme et f un polynôme de la base. Chaque noeud est étiqueté par un polynôme, et chaque arête partant de ce noeud est étiquetée par une variable. Pour simplifier un produit mf, on parcourt l'arbre en partant du noeud étiqueté par f, selon les variables intervenant dans m et suivant un ordre prédéterminé. Si le chemin aboutit, on récupère le polynôme étiquette du dernier noeud, sinon on utilise le dernier polynôme rencontré sur le chemin partiel pour effectuer la simplification. Les arbres sont mis à jour après chaque calcul de forme échelon réduite : pour tout polynôme f correspondant à une ligne de la matrice réduite, et pour tout $g \in G$ tel que $LM(g) | LM(f)$, on met à jour ou on complète l'arbre des simplifications de g en insérant f au bout du chemin d'étiquette $LM(f)/LM(g)$. On renvoie à la section 3.2.1 pour le pseudo-code des nouvelles fonctions `Simplify` et `Postprocessing`.

2.4 Algorithme F5

On présente dans cette section l'algorithme F5 introduit par Faugère en 2002 [Fau02], qui a pour objectif d'éliminer *a priori* un maximum de réductions à zéro dans le calcul des bases de Gröbner. Cet algorithme a été rendu célèbre lorsqu'il a permis la cryptanalyse du Challenge HFE [FJ03]. Depuis, il a permis d'attaquer avec succès plusieurs autres cryptosystèmes (voir par exemple [BFP10, FLDVP08]), contribuant à la popularité croissante des attaques algébriques.

Après avoir expliqué le principe du critère F5, on introduit le formalisme des polynômes

étiquetés permettant d'énoncer le résultat principal sur lequel s'appuie l'algorithme F5. Dans cette présentation, on suit l'article [EP10] de Eder et Perry (variante "F5C" qui met particulièrement l'accent sur l'aspect incrémental de l'algorithme), plutôt que l'article original [Fau02].

2.4.1 Polynômes signés et critère F5

On suppose que la matrice de Macaulay est construite jusqu'au degré d, relativement à une famille de polynômes $f_1, \ldots, f_m$. En général, cette matrice n'est pas de rang plein et le calcul de sa forme échelon va faire apparaître de nombreuses réductions à zéro, correspondant à des relations de dépendance linéaire entre des lignes de la matrice. Cependant, connaissant une base de Gröbner de l'idéal $I_{i-1} = \langle f_1, \ldots, f_{i-1} \rangle$, une remarque simple permet de détecter a priori la plupart des lignes de la forme mf_i (avec m monôme de $\mathbb{K}[X_0, \ldots, X_n]$) qui ne sont pas nécessaires au calcul de la forme échelon, c'est-à-dire qui s'obtiennent comme combinaisons linéaires des lignes antérieures $m'f_j$ avec $j < i$ ou $j = i$ et $m' \prec m$. En effet, si $m \in LT(I_{i-1})$, alors $m = LT(f)$ où $f = \sum_{j=1}^{i-1} h_j f_j \in I_{i-1}$ et on a :

$$mf_i = LT(f)f_i = ff_i - (f - LT(f))f_i = \sum_{j=1}^{i-1} (h_j f_i) f_j - (f - LT(f))f_i.$$

Il est donc possible d'éviter une réduction à zéro en n'insérant pas la ligne correspondant à mf_i dans la matrice de Macaulay dès que $m \in LT(I_{i-1})$ (dans le cas homogène ; dans le cas affine, il est possible que certains des termes de l'égalité ci-dessus soient de degré plus grand que d). Cette observation est à la base du critère F5 ; bien entendu, cela nécessite de calculer successivement les bases de Gröbner des idéaux I_i pour i allant de 1 à n, autrement dit que l'algorithme de calcul soit incrémental.

Soient $I \subset \mathbb{K}[X_1, \ldots, X_n]$ un idéal dont on suppose connue une base de Gröbner, et f un polynôme ; on veut calculer une base de Gröbner de l'idéal $I + (f)$. Le critère F5 utilise la notion de signature d'un polynôme, qui joue le rôle du monôme m dans le produit mf_i de l'analyse ci-dessus. Plus généralement, on introduit la notion de polynôme étiqueté :

Définition 2.4.1. *Soit $\mathcal{T}$ l'ensemble des monômes de $\mathbb{K}[X_1, \ldots, X_n]$ auquel on ajoute* 0.

Un polynôme étiqueté *est la donnée d'un couple $r = [s, p] \in \mathcal{T} \times \mathbb{K}[X_1, \ldots, X_n]$. Le monôme s est appelé* signature *de r, on note $Sgn(r) = s$ et $Poly(r) = p$. Par abus de notation, on note $LM(r) = LM(Poly(r))$ et $LC(r) = LC(Poly(r))$.*
Si $m \in \mathcal{T} \setminus \{0\}$, on pose $mr = [m\,Sgn(r), m\,Poly(r)]$.
Si $Sgn(r) \neq Sgn(r')$, on pose $r + r' = [\max_{\prec}(Sgn(r), Sgn(r')), Poly(r) + Poly(r')]$ (où $0 \prec m$ pour tout $m \in \mathcal{T}$ non nul).

Un polynôme étiqueté $r = [s, p]$ est dit admissible *(relativement à I et f), si*

$$\begin{cases} p \in I \text{ si } s = 0, \\ p = hf + g \text{ avec } LM(h) = s \text{ et } g \in I, \text{ sinon.} \end{cases}$$

On note que la définition de signature donnée ici diffère légèrement de celle de l'article [EP10], puisque la notion d'indice est supprimée et que la signature 0 est attribuée à tous les polynômes de I (le Corollaire 33 de [EP10] montre que ceci est licite).

Désormais tous les polynômes étiquetés considérés seront admissibles relativement à I et f.

On vérifie aisément que la notion d'admissibilité est préservée par la multiplication par un monôme et par la somme. La discussion introductive montre que les polynômes dont la signature est dans $LT(I)$ ne sont pas pertinents, ce qui motive les définitions suivantes :

Définition 2.4.2. *Un polynôme étiqueté r est dit* normalisé *si $Sgn(r) \notin LT(I)$.*
On appelle paire critique de deux polynômes étiquetés r_1 et r_2 le quintuplet, noté $CP(r_1, r_2)$, formé des données $(lcm, u_1, r_1, u_2, r_2)$ où $lcm = LM(r_1) \vee LM(r_2)$ et $u_i = lcm / LM(r_i)$ pour $i = 1, 2$. Une telle paire critique est dite normalisée *si $u_1 r_1$ et $u_2 r_2$ sont normalisés et si $Sgn(u_1 r_1) \succ Sgn(u_2 r_2)$.*

On remarque qu'avec cette notion de paires normalisées, on retrouve en partie le premier critère de Buchberger (proposition 2.1.7) : si $LM(g_1) \wedge LM(g_2) = 1$ et $g_2 \in I$, alors $u_1 = LM(g_2) \in LT(I)$, et la paire critique de r_1 et r_2 n'est pas normalisée.

On veut donner une caractérisation des bases de Gröbner similaire à celle du théorème 2.1.5 faisant intervenir des polynômes étiquetés. On adapte donc la notation o (voir définition 2.1.4) dans ce cadre :

Définition 2.4.3. *Soient $G = \{[s_1, p_1], \ldots, [s_k, p_k]\}$ et $r = [s_r, p_r]$, $t = [s_t, p_t]$ des polynômes étiquetés. Alors $r = o_G(t)$ s'il existe un polynôme étiqueté $t' = [s_{t'}, p_{t'}]$ tel que*

$$\begin{cases} s_{t'} \preccurlyeq s_t \text{ et } LM(p_{t'}) \prec LM(p_t), \\ \exists h_1, \ldots, h_k \text{ polynômes tels que } p_r = \sum_{i=1}^{k} h_i p_i \text{ avec } LM(h_i p_i) \preccurlyeq LM(p_{t'}) \text{ et } LM(h_i) s_i \preccurlyeq s_r. \end{cases}$$

La deuxième condition montre que si $r = o_G(t)$, alors $p_r = o_{Poly(G)}(LM(p_t))$. La principale différence concerne les signatures : on demande que dans l'écriture de p_r dans la base G, on n'introduise pas de signatures plus grandes que celle de r.

Théorème 2.4.4 ([Fau02, Th 1])**.** *Soient $I \subset \mathbb{K}[X_1, \ldots, X_n]$ idéal, $f \in \mathbb{K}[X_1, \ldots, X_n]$, et $G = \{r_1, \ldots, r_k\}$ une famille de polynômes étiquetés admissibles ($r_i = [s_i, g_i]$) telle que*

1. *$[1, f] \in G$,*
2. *il existe $s < k$ tel que $\{g_1, \ldots, g_s\}$ est une base de Gröbner de I et $s_1 = \ldots = s_s = 0$,*
3. *$\forall i, j,\ 1 \leq i, j \leq k$ tels que $CP(r_i, r_j)$ normalisée, on a*

$$S(r_i, r_j) := [u_i s_i, LC(g_j) u_i g_i - LC(g_i) u_j g_j] = o_G(u_i r_i) \text{ où } u_{i,j} = \frac{LM(g_i) \vee LM(g_j)}{LM(g_{i,j})}.$$

Alors $\{g_1, \ldots, g_k\}$ est une base de Gröbner de $I + (f)$.

On remarque que la condition donnée en *3.* implique que $S(g_i, g_j) = o_{Poly(G)}(LM(g_i) \vee LM(g_j))$, qui est précisément celle donnée dans le théorème 2.1.5. On doit donc vérifier ici une propriété plus forte sur les S-polynômes, cependant cette propriété doit être vérifiée sur moins de paires (toutes les paires non normalisées étant automatiquement rejetées).

2.4.2 Description de l'algorithme

Comme pour Buchberger, on peut déduire du théorème 2.4.4 un algorithme permettant de calculer une base de Gröbner d'un idéal $I = \langle f_1, \ldots, f_k \rangle$ donné. L'idée est de calculer récursivement une base de Gröbner de $I_i = \langle f_1, \ldots, f_i \rangle = I_{i-1} + (f_i)$ pour $i = 2, \ldots, k$, en ne considérant que

des paires normalisées. La principale différence avec l'algorithme de Buchberger est que lors de la réduction d'une paire critique $CP(r_1, r_2)$ par la base courante G, on veut que le reste p_r ait une signature admissible s et que l'écriture donnée par la réduction $Poly(S(r_1, r_2)) = \sum h_i g_i + p_r$ garantisse que $S(r_1, r_2) = o_{G\cup\{[s,p_r]\}}(u_1 r_1)$. Pour cela, on n'autorise que les réductions par des polynômes étiquetés ayant des signatures strictement inférieures à $u_1 Sgn(r_1)$; la signature s du reste p_r est alors naturellement $u_1 Sgn(r_1)$ et le polynôme étiqueté t' intervenant dans la notation $o_{G\cup\{[s,p_r]\}}(u_1 r_1)$ précédente est exactement $S(r_1, r_2)$.

L'algorithme présenté dans cette section comporte un certain nombre d'optimisations par rapport au squelette ci-dessus. Premièrement, comme dans l'algorithme F4, on traite plusieurs paires critiques à chaque étape selon une stratégie de sélection `Sel`. Deuxièmement, lors de la phase de réduction des têtes des S-polynômes (top-réductions), il se peut qu'un polynôme étiqueté r' dans la base courante ait un terme de tête qui divise le terme de tête d'un S-polynôme r, mais que la réduction de r par r' ne soit pas autorisée parce que $uSgn(r') \succ Sgn(r)$ (où $u = LM(r)/\,LM(r')$). Dans ce cas, on introduit sans attendre la paire critique de r et de r', dont le S-polynôme a pour signature $uSgn(r')$. Ainsi, la procédure `Reduction` peut retourner plus de polynômes qu'elle n'en avait en entrée. Enfin, on introduit un deuxième critère basé sur des "règles de réécriture", dont l'idée est d'éviter de considérer plusieurs polynômes ayant les mêmes signatures ; si l'on reprend l'analogie avec la matrice de Macaulay, cela reviendrait à insérer plusieurs fois la ligne mf_i de signature m. Pour cela, on maintient un tableau de "règles" indiquant, pour chaque signature déjà rencontrée, le polynôme étiqueté correspondant ; à chaque fois que l'on doit faire un produit ur, on teste s'il n'existe pas un polynôme plus récent (donc plus réduit) r' et un monôme u' tel que $uSgn(r) = u'Sgn(r')$. Par souci de clarté, certaines parties du pseudo-code ont été fusionnées par rapport à celui donné dans [EP10] et [Fau02] : ainsi, `Reduction` et `SeReecrit` regroupent toutes les sous-procédures ayant trait à la réduction des polynômes et au critère de réécriture respectivement.

Le cœur de l'algorithme est la procédure `F5Increment`, qui calcule une base de Gröbner réduite de l'idéal $\langle f_1, \ldots, f_{i+1}\rangle$ connaissant une base de Gröbner de $\langle f_1, \ldots, f_i\rangle$. Tous les polynômes étiquetés intervenants sont stockés dans une liste globale *Liste*, et la liste G contient les numéros dans *Liste* des éléments de la base courante. À l'initialisation, la signature 0 est attribuée à tous les polynômes de l'ancienne base de Gröbner G_{prec}, tandis que le nouveau générateur reçoit la signature 1. La procédure `PaireCrit` utilise le critère F5 pour vérifier que la paire critique de deux polynômes étiquetés est bien normalisée ; le critère de réécriture ajouté en ligne 5 n'est pas strictement nécessaire puisqu'il sera retesté dans `CalculSpol` ensuite (avec éventuellement plus de règles), mais permet de diminuer la liste CP des paires en attente de traitement. Le rôle de `CalculSpol` est principalement de tester ce critère de réécriture et de créer les nouveaux polynômes étiquetés, en mettant à jour au passage la liste des règles. Enfin, la procédure `Reduction` réduit les S-polynômes par rapport à la base courante tout en respectant les critères de signatures.

Algorithme 9: Algorithme F5

Entrées : $I = \langle f_1, \ldots, f_k\rangle \subset \mathbb{K}[X_1, \ldots, X_n]$
Sortie : GB base de Gröbner minimale de I

1. $GB \leftarrow \{f_1\}$;
2. **pour** $i = 2$ *à* k **faire**
3. $\quad f_{nouv} \leftarrow \overline{f_i}^{GB}$;
4. $\quad$ **si** $f_{nouv} \neq 0$ **alors** $GB \leftarrow$ `F5Increment` $(GB, f_{nouv}/\,LC(f_{nouv}))$;
5. $\quad$ **si** $GB = \{1\}$ **alors retourner** GB;
6. **retourner** GB;

Algorithme 10: F5Increment

Entrées : G_{prec} une base de Gröbner, $f \in \mathbb{K}[X_1, \ldots, X_n]$
Sortie : GB base de Gröbner minimale de $G_{prec} + (f)$

1. $Liste \leftarrow [\,]$; CP $\leftarrow \emptyset$; $Regles \leftarrow [\,]$;
2. **pour tout** $g \in G_{prec}$ **faire**
3. CP $\leftarrow$ CP $\cup$ `PaireCrit` $([0, g], [1, f], G_{prec}, Regles)$;
4. Apposer$(Liste, [0, g])$;
5. Apposer$(Liste, [1, f])$; Apposer$(Regles, (1, \#G_{prec} + 1))$; $G \leftarrow [1, \ldots, \#G_{prec} + 1]$;
   `// Boucle principale`
6. **tant que** CP $\neq \emptyset$ **faire**
7. CP$_{sel}$ $\leftarrow$`Sel` (CP) ; CP $\leftarrow$ CP $\setminus$ CP$_{sel}$;
8. $F \leftarrow$`CalculSpol` (CP$_{sel}$, $Regles$);
9. $Nouveaux \leftarrow$`Reduction` $(F, G, G_{prec}, Regles)$;
10. **pour tout** $r \in Nouveaux$ **faire**
11. **pour** $i = 1$ *à* $\#G$ **faire** CP $\leftarrow$ CP $\cup$ `PaireCrit` $(Liste[G[i]], r, G_{prec}, Regles)$;
12. $G \leftarrow G \cup \{\text{Indice}_{Liste}(r)\}$;
13. $GB \leftarrow \{Poly(Liste[k]) : k \in G\}$;
14. $GB \leftarrow$ Minimalisation(GB);
15. $GB \leftarrow$ InterReduction(GB);
16. **retourner** GB;

Algorithme 11: PaireCrit

Entrées : r_1, r_2 polynômes étiquetés, G_{prec} une base de Gröbner,
Sortie : ensemble de paire soit vide soit réduit à un singleton

1. $lcm \leftarrow LM(r_1) \vee LM(r_2)$;
2. $u_1 \leftarrow lcm / LM(r_1)$; $u_2 \leftarrow lcm / LM(r_2)$;
3. **si** $u_1 Sgn(r_1) \neq 0$ *et* $\exists g \in G_{prec}$ *tq* $LM(g) | u_1 Sgn(r_1)$ **alors retourner** $\emptyset$; `// critère F5`
4. **si** $u_2 Sgn(r_2) \neq 0$ *et* $\exists g \in G_{prec}$ *tq* $LM(g) | u_2 Sgn(r_2)$ **alors retourner** $\emptyset$; `// critère F5`
5. **si** `SeReecrit` $(u_1, r_1, Regles)$ *ou* `SeReecrit` $(u_2, r_2, Regles)$ **alors retourner** $\emptyset$;
6. **si** $u_1 Sgn(r_1) \prec u_2 Sgn(r_2)$ **alors retourner** $\{(lcm, u_2, r_2, u_1, r_1)\}$;
7. **sinon retourner** $\{(lcm, u_1, r_1, u_2, r_2)\}$;

Algorithme 12: CalculSpol

Entrées : CP$_{sel}$ liste de paire critiques, $Regles$
Sortie : F liste de polynômes étiquetés

1. $F \leftarrow \emptyset$;
2. **pour tout** $paire = (lcm, u_1, r_1, u_2, r_2) \in$ CP$_{sel}$ **faire**
3. **si** $\neg$`SeReecrit` $(u_1, r_1, Regles)$ *et* $\neg$`SeReecrit` $(u_2, r_2, Regles)$ **alors**
4. $r \leftarrow [u_1 Sgn(r_1), u_1 Poly(r_1) - u_2 Poly(r_2)]$;
5. Apposer$(Liste, r)$;
6. Apposer$(Regles, (Sgn(r), \text{Indice}_{Liste}(r)))$;
7. $F \leftarrow F \cup r$;
8. Trier F par signature croissante;
9. **retourner** F;

Algorithme 13: Reduction

Entrées : F liste de polynômes étiquetés triés par signature croissante, G les numéros dans liste des éléments de la base courante, G_{prec}, *Regles*
Sortie : *Done* une liste de polynômes étiquetés (triés par signature croissante)

1. $Done \leftarrow [\,]$; $Red \leftarrow G$ (réducteurs potentiels);
2. **tant que** $F \neq [\,]$ **faire**
3. $\quad r \leftarrow$ élément minimal de F ; $F \leftarrow F \setminus \{r\}$;
4. $\quad Poly(r) \leftarrow \overline{Poly(r)}^{G_{prec}}$;
5. $\quad$ **pour** $i = 1$ *à* $\#Red$ **faire**
6. $\quad\quad$ **si** $Poly(r) = 0$ **alors** $Liste[\text{Indice}_{Liste}(r)] \leftarrow r$; **break**; `// réduction à zéro`
7. $\quad\quad r' \leftarrow Liste[Red[i]]$;
8. $\quad\quad$ **si** $LM(r') \nmid LM(r)$ **alors continue**;
9. $\quad\quad u \leftarrow LM(r)/\,LM(r')$;
10. $\quad\quad$ **si** $Sgn(r') \neq 0$ *et* $\exists g \in G_{prec}$ *tq* $LM(g) | uSgn(r')$ **alors continue**; `// critère F5`
11. $\quad\quad$ **si** `SeReecrit` $(u, r', Regles)$ **alors continue**;
$\quad\quad$ `// Top-Réduction`
12. $\quad\quad$ **si** $uSgn(r') \prec Sgn(r)$ **alors**
13. $\quad\quad\quad Poly(r) \leftarrow Poly(r) - LC(r)uPoly(r')$;
14. $\quad\quad\quad i \leftarrow 0$;
15. $\quad\quad$ **sinon**
16. $\quad\quad\quad r'' \leftarrow [uSgn(r'), Poly(r) - LC(r)uPoly(r')]$;
17. $\quad\quad\quad$ Apposer$(Liste, r'')$;
18. $\quad\quad\quad$ Apposer$(Regles, (Sgn(r''), \text{Indice}_{Liste}(r'')))$;
19. $\quad\quad\quad$ Apposer(F, r'') ; Trier F;
20. $\quad$ **si** $Poly(r) \neq 0$ **alors**
21. $\quad\quad Poly(r) \leftarrow Poly(r)/\,LC(r)$; $Liste[\text{Indice}_{Liste}(r)] \leftarrow r$;
22. $\quad\quad Done \leftarrow Done \cup \{r\}$;
23. $\quad\quad Red \leftarrow Red \cup \{\text{Indice}_{Liste}(r)\}$;
24. **retourner** *Done*;

Algorithme 14: SeReecrit

Entrées : u monôme, r polynôme étiqueté, *Regles* une liste de couples (monôme, numéro)
Dessert : vrai ou faux

1. **si** $Sgn(r) = 0$ **alors retourner** faux;
2. **pour** $(m, num) \in Regles$ *en partant de la fin* **faire**
3. $\quad$ **si** $m | uSgn(r)$ **alors**
4. $\quad\quad$ **si** $num = \text{Indice}_{Liste}(r)$ **alors retourner** faux;
5. $\quad\quad$ **sinon**
6. $\quad\quad\quad$ **retourner** vrai;
7. **retourner** faux;

Exemple 2.4.5. *On reprend le calcul de la base de Gröbner pour l'ordre* $grevlex_{x\succ y\succ z\succ t}$ *de l'idéal engendré par les polynômes*

$$Cyclic_4 \ : \ \begin{cases} f_1 = x+y+z+t \\ f_2 = xy+yz+zt+tx \\ f_3 = xyz+yzt+ztx+txy \\ f_4 = xyzt-1 \end{cases}$$

déjà présenté en exemple 2.1.8.

1. *On commence par calculer la base de Gröbner de* $I_2 = \langle f_1, f_2\rangle$. *On appelle* `F5Increment` *avec* $G_{prec} = \{f_1\}$ *et* $f_{nouv} = f_5 = \overline{f_2}^{\{f_1\}} = y^2+2yt+t^2$, *et on pose* $r_1 = [0, f_1]$ *et* $r_2 = [1, f_5]$. *La paire critique de* r_2 *et* r_1 *est* $(xy^2, x, [1, f_5], y^2, [0, f_1])$ *qui n'est pas normalisée car* $x.1 \in LT(G_{prec})$. *On a donc* $G = \{r_1, r_2\}$ *et* $\mathrm{CP} = \emptyset$ *;* `F5Increment` *termine et retourne* $GB = \{f_1, f_5\}$.
2. *On calcule ensuite la base de Gröbner de* $I_3 = \langle f_1, f_2, f_3\rangle$, *en appellant* `F5Increment` *avec* $G_{prec} = \{f_1, f_5\}$ *et* $f_{nouv} = f_6 = \overline{f_3}^{G_{prec}} = yz^2+z^2t-yt^2-t^3$. *On pose* $r_1 = [0, f_1]$, $r_2 = [0, f_5]$ *et* $r_3 = [1, f_6]$. *Comme précédemment, la paire critique de* r_3 *et* r_1 *est rejetée par le critère F5 (non normalisée). La paire critique de* r_3 *et* r_2 *est conservée :* $p_1 = (y^2z^2, y, [1, f_6], z^2, [0, f_5])$. *On a donc* $G = \{r_1, r_2, r_3\}$, $\mathrm{CP} = \{p_1\}$, *Regles* $= [(1, r_3)]$.
   - $\mathrm{CP}_{sel} = \{p_1\}$. *La paire* p_1 *passe le critère de réécriture (car* $y.r_3$ *ne se réécrit qu'avec* r_3*), on calcule donc son S-polynôme* $r_4 = [y, -yz^2t-y^2t^2-z^2t^2-yt^3]$ *et on rajoute la règle* (y, r_4). *La réduction de* r_4 *par* G *donne* 0, *aucun élément n'est ajouté à* G. *On a donc* $G = \{r_1, r_2, r_3\}$, $\mathrm{CP} = \emptyset$, *Regles* $= [(1, r_3), (y, r_4)]$. *Comme il n'y a plus de paires en attente,* `F5Increment` *termine et retourne* $GB = \{f_1, f_5, f_6\}$.
3. *On calcule ensuite la base de Gröbner de* $I_4 = \langle f_1, f_2, f_3, f_4\rangle$, *en appellant* `F5Increment` *avec* $G_{prec} = \{f_1, f_5, f_6\}$ *et* $f_{nouv} = f_7 = \overline{f_4}^{G_{prec}} = yzt^2+z^2t^2-yt^3+zt^3-t^4-1$. *On pose* $r_1 = [0, f_1]$, $r_2 = [0, f_5]$, $r_3 = [0, f_6]$ *et* $r_4 = [1, f_7]$. *La paire critique de* r_4 *et* r_1 *est rejetée par le critère F5, et on a* $G = \{r_1, r_2, r_3, r_4\}$, $\mathrm{CP} = \{p_1, p_2\}$, *Regles* $= [(1, r_4)]$ *où* $p_1 = (y^2zt^2, y, [1, f_7], zt^2, [0, f_5])$ *et* $p_6 = (yz^2t^2, z, [1, f_7], t^2, [0, f_6])$.
   - $\mathrm{CP}_{sel} = \{p_1, p_2\}$. *Les deux paires passent le critère de réécriture, on calcule leurs S-polynômes* $r_5 = [z, z^3t^2-yzt^3+yt^4-zt^4+t^5-z]$ *et* $r_6 = [y, yz^2t^2-y^2t^3-yzt^3-yt^4-zt^4-y]$, *et on rajoute deux nouvelles règles* (z, r_5) *et* (y, r_6). *La procédure de réduction ne modifie pas* $r_5 = [z, f_8]$ *(il serait en fait possible de réduire la queue de* r_5, *voir section 2.4.4) et donne comme nouvelle valeur à* r_6 $[y, f_9]$ *où* $f_9 = yt^4+t^5-y-t]$. *Les paires de* r_5 *avec* r_1 *et* r_2 *sont rejetées par le critère F5. La paire de* r_5 *avec* r_3 *est* $(yz^2t^2, y, r_5, zt^2, r_3)$, *et le produit* $y \cdot r_5$ *peut se réécrire avec la règle* (y, r_6) *: cette paire est donc rejetée par le critère de réécriture, de même que celle de* r_5 *avec* r_4. *Les paires de* r_6 *avec* r_1, r_2, r_3 *et* r_5 *sont rejetées par le critère F5, et on a* $G = \{r_1, r_2, r_3, r_4, r_5, r_6\}$, $\mathrm{CP} = \{p_3\}$, *Regles* $= [(1, r_4), (z, r_5), (y, r_6)]$ *où* $p_3 = (yzt^4, z, [y, f_9], t^2, [1, f_7])$.
   - $\mathrm{CP}_{sel} = \{p_3\}$. *La paire passe le critère de réécriture, on calcule son S-polynôme* $r_7 = [yz, z^2t^4-yt^5-t^6+yz+zt-t^2]$ *et on rajoute la règle* (yz, r_7). *La procédure de réduction ne modifie pas* $r_7 = [z, f_{10}]$, *même s'il serait encore possible de réduire sa queue. Les paires de* r_7 *avec tous les éléments de* G *sont rejetées par le critère F5, et on a* $G = \{r_1, r_2, r_3, r_4, r_5, r_6\}$, $\mathrm{CP} = \emptyset$, *Regles* $= [(1, r_4), (z, r_5), (y, r_6), (yz, r_7)]$. *Comme il n'y a plus de paires en attente,* `F5Increment` *termine et retourne la réduction de la base de Gröbner* $\{f_1, f_5, f_6, f_7, f_8, f_9, f_{10}\}$, *obtenue en remplaçant* f_8 *par* $z^3t^2+z^2t^3-z-t$ *et* f_{10} *par* $z^2t^4+yz-yt+zt-2t^2$.

Sur cet exemple, on a éliminé au total 17 *paires inutiles, et il ne reste plus qu'une seule réduction à zéro (à la place de* 5 *pour l'algorithme de Buchberger avec critères). À ce propos, on remarque que* $f_5 = (y+t)^2$ *et que* $f_6 = (y+t)(z^2-t^2)$*, en particulier* $(y+t)f_6 \in I_2$ *; la suite* $\{f_1, f_2, f_3, f_4\}$ *n'est donc pas régulière (ni semi-régulière, voir ci-dessous).*

2.4.3 Analyse et complexité

Il est facile de vérifier que tous les polynômes étiquetés qui interviennent dans l'algorithme présenté sont bien admissibles. De même, si l'on supprime les tests de réécriture, la base G en sortie de la boucle principale de l'algorithme `F5Increment` satisfait bien les hypothèses du théorème 2.4.4, ce qui prouve que F5 retourne bien une base de Gröbner de I. Justifier la correction de l'algorithme lorsque l'on prend en compte le critère de réécriture est plus délicat : on renvoie à l'article [EP10] pour une démonstration exhaustive.

La terminaison de l'algorithme est plus problématique et reste non démontrée [2], bien que l'on ne connaisse aucun contre-exemple à ce jour. En effet, une des principales différences entre cet algorithme et les algorithmes de Buchberger et F4 réside dans la construction de polynômes "redondants" : ce sont des polynômes dont le terme de tête est divisible par un élément de la base courante mais qui ne peuvent être réduits à cause du critère sur les signatures. Ces polynômes ne font donc pas partie d'une base de Gröbner minimale, mais ne peuvent pas être écartés lors de l'exécution de F5 dans la mesure où leur signature apporte une information nécessaire pour son déroulement correct (un des intérêts de la variante F5C [EP10] est précisément de diminuer le nombre de ces polynômes). On peut alors imaginer une situation dans laquelle on ajouterait indéfiniment à la base courante des polynômes redondants qui créeraient de nouvelles paires critiques, dont la réduction donnerait à nouveau des polynômes redondants, etc. Il est cependant possible de modifier légèrement l'algorithme et de rajouter quelques tests qui permettent de garantir la terminaison dans tous les cas [EGP11].

La grande force de l'algorithme F5 est d'éviter la plupart des réductions à zéro. De fait, il est possible de démontrer que pour une large classe de systèmes, F5 ne calcule aucune réduction à zéro ; c'est le cas en particulier des systèmes réguliers présentés en section 2.2.2 [Fau02, Corollaire 3]. Un autre cas intéressant est celui des suites *semi-régulières* :

Définition 2.4.6 ([Bar04])**.** *Une suite* $f_1, \ldots, f_m$ *de polynômes homogènes est dite* semi-régulière *si*

(i) $I = \langle f_1, \ldots, f_m \rangle \neq \mathbb{K}[X_0, \ldots, X_n]$,

(ii) pour tout $i \in [\![1; m]\!]$*, si* $g_i \in (\langle f_1, \ldots, f_{i-1} \rangle : f_i)$ *est tel que* $\deg(g_i f_i) < d_{reg}(I)$*, alors* $g_i \in \langle f_1, \ldots, f_{i-1} \rangle$.

Il est conjecturé que la plupart des systèmes surdéterminés sont de type semi-régulier ; la généricité est démontrée dans de nombreux cas. Les suites semi-régulières sont essentiellement les suites de polynômes qui se comportent bien pour l'algorithme F5 : pour un ordre gradué par le degré et une stratégie de sélection par degré croissant, on peut montrer qu'il n'y a pas de réduction à zéro avant le degré de régularité (voir [Bar04] pour plus de détails). Comme le degré de régularité est aussi le degré maximal atteint durant le calcul (à changement linéaire de variables près), ceci signifie qu'on peut calculer les bases de Gröbner de systèmes semi-réguliers sans réductions à zéro.

2. Comme déjà remarqué dans [EGP11], le Théorème 2 de [Fau02] qui établit la terminaison dans le cas où il n'y a pas de réduction à zéro est en fait faux, ainsi que le montre l'exemple 8 (au degré $d = 8$) du même article [Fau02].

Comme pour tous les algorithmes qui descendent de celui de Buchberger, il est très difficile de donner une borne précise de la complexité de l'algorithme F5. Dans le cas des systèmes réguliers et semi-réguliers, on peut la majorer par la complexité de la réduction de la matrice de Macaulay en degré d_{reg}, dont on a supprimé toutes les lignes donnant des réductions à zéro. Cette matrice contient au plus $\binom{d_{reg}+n+1}{d_{reg}}$ colonnes et nécessairement moins de lignes puisqu'il n'y a pas de réductions à zéro. En utilisant les algorithmes de calcul de forme échelon basés sur les méthodes de multiplication matricielle rapide (comme par exemple Strassen [Str69]), on obtient une complexité en

$$O\left(\binom{d_{reg}+n+1}{d_{reg}}^{\omega}\right) \tag{2.1}$$

opérations dans $\mathbb{K}$, où ω est la constante intervenant dans la complexité du produit matriciel. Cette borne reste valable dans le cas affine s'il n'y a pas de réduction à zéro, pour un ordre gradué par le degré et une stratégie de sélection adaptée.

2.4.4 Vers une implantation efficace de F5

Dans l'algorithme tel qu'il est présenté ici ou dans [Fau02], la réduction des polynômes est incomplète : à part pour le calcul initial de la forme normale en ligne 4 de `Reduction`, on se contente de "top-réduire" les polynômes, i.e. la réduction s'arrête quand le terme de tête n'est plus divisible par les éléments de la base courante. Pour accélérer les réductions ultérieures, il serait possible d'essayer de réduire plus complètement les S-polynômes. Cependant, le surcoût lié à la recherche de réducteurs vérifiant les conditions de signature peut être plus important que le gain attendu. Il reste néanmoins possible de recalculer la réduction complète par G_{prec}, qui ne pose pas de problème de signature, après la top-réduction.

Le choix de la stratégie de sélection `Sel` a aussi une influence non négligeable sur les calculs. Dans [Fau02], Faugère suggère d'utiliser la stratégie "normale", qui consiste à sélectionner à chaque étape toutes les paires critiques de *lcm* de plus petit degré : c'est la stratégie qui donne usuellement les meilleurs résultats pour l'algorithme F4 pour un ordre gradué par le degré. Si cela semble être encore vrai pour F5 avec des polynômes homogènes, dans le cas affine, on a constaté qu'une stratégie de sélection par signature de S-polynômes croissante était plus efficace et générait moins de polynômes redondants (elle correspond en quelque sorte à la stratégie du sucre donnée dans le contexte de l'algorithme de Buchberger ; voir [GMN+91]). On donne en figure 2.5, voir aussi section 3.3.1, une comparaison entre ces deux stratégies sur un exemple où il y a de nombreuses chutes de degré : le nombre d'itérations dans la stratégie par signature est plus de sept fois inférieur à celui de la stratégie normale.

Mais pour vraiment améliorer les performances, il est crucial d'agir au niveau de la réduction des S-polynômes en utilisant des outils d'algèbre linéaire ; comme remarqué par Faugère dans [Fau02], *"from the efficiency point of view, it is recommended to translate the algorithm in a F4 fashion"*. On note d'ailleurs que les performances de l'implantation FGb de l'algorithme F5 (dont le code source n'est pas public) telles qu'elles apparaissent dans les articles [BFP10, Fau02, FLDVP08] sont de très loin supérieures à celles des implantations publiques (voir par exemple [EP10, GGV10]), ce qui laisse place à de nombreuses optimisations. Cependant, combiner F5 avec des réductions matricielles est beaucoup plus difficile qu'il n'y ne paraît. Une première tentative en ce sens est l'algorithme F5-Matriciel introduit dans [Bar04] ; mais cette version n'utilise pas le concept de paires critiques et est en fait une amélioration de l'algorithme de Lazard, où le critère F5 sert à éliminer certaines lignes des matrices de Macaulay. En conséquence, de nombreuses lignes restent inutiles dans cette matrice car elles n'interviennent dans aucune top-réduction.

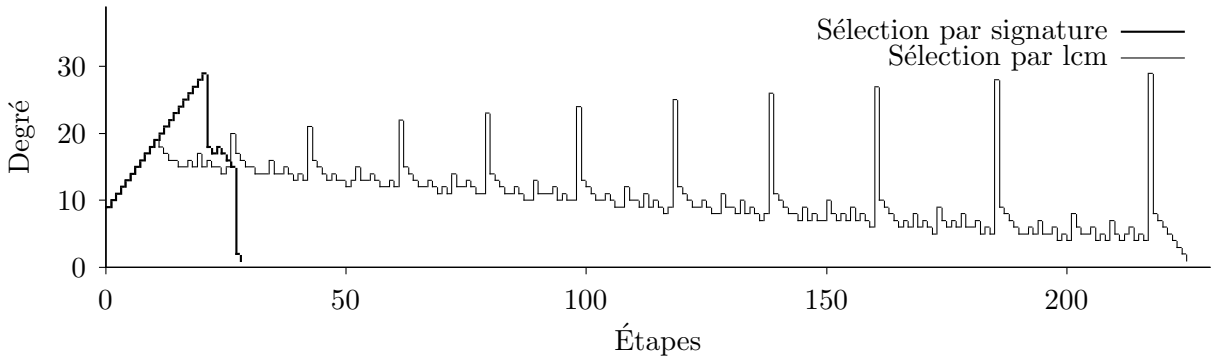


FIGURE 2.5 – Comparaison, selon la stratégie de sélection, des degrés maximaux des polynômes des paires critiques à chaque étape de `F5Increment` pour le système présenté en section 3.3.1

Une autre approche, directement inspirée de F4, est présentée dans [AP10, Ars05]. On range dans une matrice de type Macaulay les polynômes provenant des paires critiques, que l'on complète lors d'un preprocessing par les multiples normalisés des éléments de la base courante permettant de réduire les termes de ces polynômes. On peut alors procéder à une pseudo-réduction gaussienne, en faisant attention à respecter les signatures : si l'on trie les lignes de la matrice par signature croissante, chaque pivot ne peut réduire que les lignes qui lui sont inférieures (donc de signature plus grande). En particulier, la forme de la matrice réduite n'est en général absolument pas triangulaire supérieure. Le plus délicat est la gestion des règles de réécriture, puisque chaque nouveau générateur, apparaissant dès que le terme de tête initial d'une ligne est réduit, correspond à une nouvelle règle de réécriture qui peut rendre invalides certaines des lignes inférieures.

On donne en figures 2.6 et 2.7 des exemples de matrices "peudo-réduites" obtenues avec cette version matricielle. La partie supérieure de la matrice (en noir) correspond aux éléments de signature 0, pour lesquelles on a effectué une réduction complète et utilisé une procédure type `Simplify` ; dans la partie inférieure, les polynômes sont triés par signature croissante et les tranches de couleur correspondent à des degrés totaux de signature différents. Les grandes divisions verticales correspondent aux différents degrés des monômes. La structure de ces matrices est assez représentative. Tant que les chutes de degré ne sont pas très nombreuses, le terme de tête d'un polynôme étiqueté croît globalement avec la signature : ceci explique l'aspect triangulaire inférieur droit (sauf pour la partie de signature 0). Par contre, la structure locale est celle de blocs échelonnés inférieurement, et dont la partie triangulaire supérieure est pleine puisqu'une réduction complète n'est pas permise par le critère de signatures. L'importante partie vide inférieure gauche sur la figure 2.7 provient d'une part de la sélection des paires critiques par signature croissante et d'autre part de la présence de nombreuses chutes de degré jusqu'à cette étape pour le système considéré.

Cette approche matricielle de l'algorithme F5 présente néanmoins de nombreux inconvénients. Premièrement, la pseudo-réduction paraît incompatible avec les méthodes de réduction rapide, ou avec les optimisations présentées en section 2.3.3. De façon plus gênante, le preprocessing risque d'insérer de nombreuses lignes inutiles dans la matrice. En effet, si la signature d'une ligne uf de terme de tête m est plus grande que la signature des autres lignes où intervient le monôme m, alors cette ligne ne va jouer un rôle dans la réduction que dans le cas où m devient le terme de tête d'un nouveau générateur, ce qu'il est évidemment impossible de prévoir au moment du preprocessing. Enfin, une part importante des performances de l'algorithme F4 provient de la réutilisation des calculs d'une matrice à l'autre (procédure `Simplify`). Dans la mesure où les réductions sont

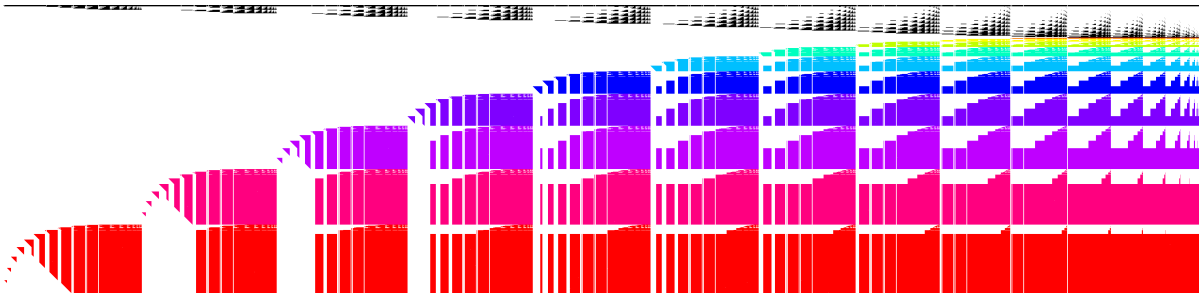


FIGURE 2.6 – Matrice de taille $1\,078 \times 4\,479$ obtenue à l'étape 10 du calcul de la base de Gröbner du système (composé des 4 premiers polynômes) issu du problème présenté en section 3.3.1.

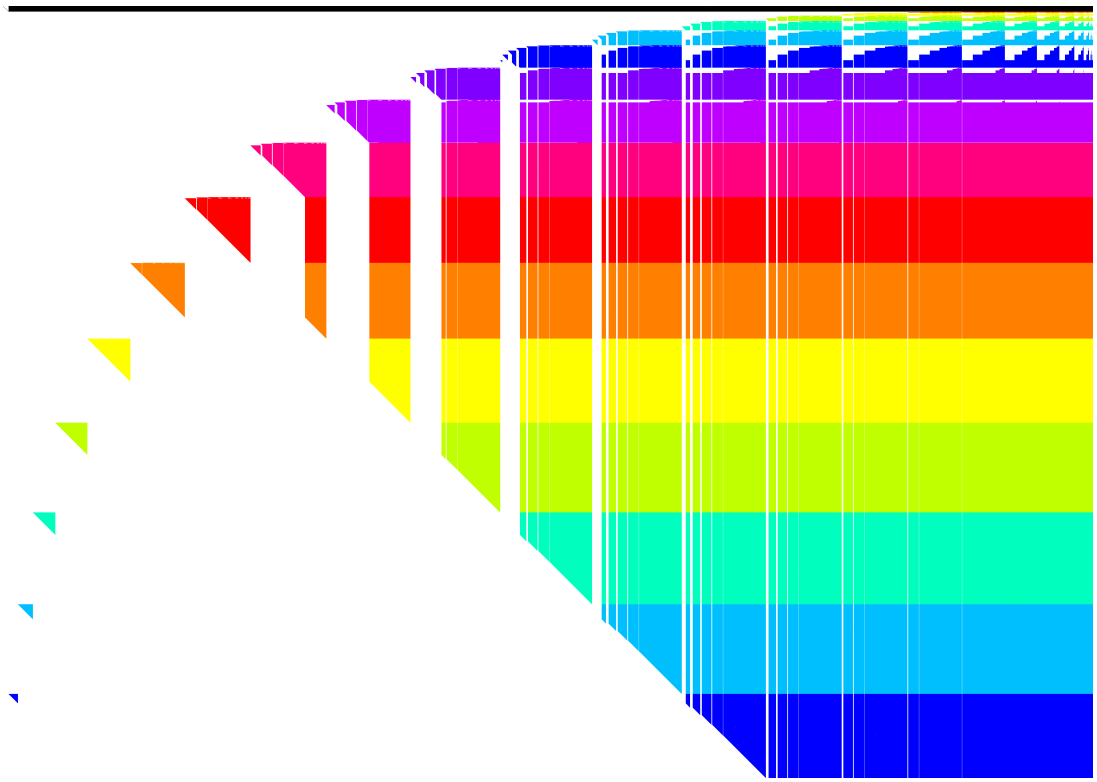


FIGURE 2.7 – Matrice de taille $2\,892 \times 4\,082$ obtenue à l'étape 16 du calcul de la base de Gröbner du système (composé de 5 polynômes) issu du même problème.

beaucoup moins complètes avec F5 (à cause des critères de signatures), le gain attendu d'une telle réutilisation est relativement faible.

Le caractère incrémental de l'algorithme F5 est aussi sujet à amélioration. Par exemple, avec la version présentée ici, il est clair qu'il est plus difficile de calculer une base de Gröbner de $\langle f_1, \ldots, f_m \rangle$ que de $\langle f_1, \ldots, f_{m-1} \rangle$, alors que pour des systèmes surdéterminés cela ne devrait pas être le cas. Pour remédier à ce problème, on peut inverser ou fusionner les deux boucles principales de l'algorithme (celle sur les f_i et celle sur les paires critiques). La signature d'un polynôme p devient alors un couple (u, ind), où u est un monôme et $ind \in [\![1; m]\!]$, qui est admissible s'il existe une écriture $p = \sum_{i=1}^{ind} h_i f_i$ avec $LM(h_{ind}) = u$, et qui est normalisée si $u \notin LT(\langle f_1, \ldots, f_{ind-1} \rangle)$. La difficulté est que dans ce cas on ne connaît pas lors du déroulement de l'algorithme les bases de Gröbner des $\mathcal{I}_i = \langle f_1, \ldots, f_i \rangle$ pour pouvoir appliquer le critère F5. Cependant, si l'on travaille par degré

croissant avec des polynômes homogènes, à l'étape d on va obtenir les bases de Gröbner tronquées à l'ordre d des idéaux $\mathcal{I}_i$, ce qui est suffisant pour déterminer les paires critiques normalisées à l'étape $d+1$.

2.5 Changement d'ordres

La connaissance d'une base de Gröbner d'un idéal pour l'ordre lexicographique est bien utile pour exprimer les solutions du système correspondant (cf section 1.2.1), mais comme on peut s'y attendre, son calcul reste en général très difficile (la résolution de systèmes polynomiaux à plusieurs variables étant elle-même un problème difficile). En pratique, le calcul direct d'une base de Gröbner est beaucoup plus dur pour certains ordres que pour d'autres, l'ordre lexicographique étant généralement le plus complexe et l'ordre grevlex le plus simple (voir [Laz83]).

Lorsque l'on connaît déjà une base de Gröbner G_1 d'un idéal I pour un ordre monomial $\prec_1$ donné, il est possible d'utiliser cette information pour accélérer le calcul d'une base G_2 du même idéal pour un autre ordre $\prec_2$. On dispose pour cela essentiellement de deux algorithmes de changement d'ordre : le premier est l'algorithme FGLM (du nom de ses auteurs Faugère, Gianni, Lazard et Mora [FGLM93]) qui s'applique au cas 0-dimensionnel, le deuxième est l'algorithme *Gröbner walk* (ou marche de Gröbner) applicable dans tous les cas [CKM97]. Dans cette thèse, on s'intéresse principalement à la résolution de systèmes polynomiaux ayant un nombre fini de solutions (donc 0-dimensionnel), et c'est l'algorithme FGLM qui est le plus pertinent dans ce contexte. On utilisera cependant à quelques occasions des bases de Gröbner pour l'ordre lex d'idéaux de dimension positive, pour lesquelles on emploiera des marches de Gröbner (voir section 7.3.5).

Expérimentalement, on constate qu'il est souvent plus rapide pour obtenir une base de Gröbner lexicographique de calculer d'abord une base pour un ordre gradué par le degré tel que grevlex, puis d'appliquer un changement d'ordre, plutôt que de faire un calcul direct.

2.5.1 Le cas de la dimension 0 : FGLM

Soit $I \subset \mathbb{K}[X_1, \ldots, X_n]$ un idéal de dimension 0 et de degré D, et soit G_1 une base de Gröbner de I pour un ordre $\prec_1$. On a vu en section 1.1.5 qu'une base (en tant qu'espace vectoriel) du quotient $A = \mathbb{K}[X_1, \ldots, X_n]/I$ est donnée par les classes des monômes sous l'escalier de G_1 ; on note $\mathcal{B}_1 = \{m_1, \ldots, m_D\}$ l'ensemble de ces monômes et $\overline{\mathcal{B}}_1$ leurs classes dans A. Le calcul de la forme normale modulo G_1 donne alors une procédure pour exprimer dans $\overline{\mathcal{B}}_1$ l'image d'un polynôme dans A. L'idée de l'algorithme FGLM est de parcourir l'ensemble des monômes dans l'ordre croissant pour $\prec_2$ et d'exprimer de proche en proche leur classe dans A, jusqu'à trouver une relation de dépendance linéaire qui correspond à un premier polynôme de G_2. On recommence ensuite avec les monômes qui ne sont pas divisibles par le terme de tête de ce premier polynôme. Une description plus complète de l'algorithme est la suivante :

1. Pour tout $l \in [\![1; n]\!]$, l'application $mult_l : A \to A, \quad \overline{f} \mapsto \overline{X_l.f}$ est $\mathbb{K}$-linéaire. La première étape de l'algorithme consiste à construire pour tout l la matrice $M_l = (M_{l,ij})$ de cette application dans la base $\overline{\mathcal{B}}_1$; cette construction s'effectue de proche en proche en parcourant les monômes sous l'escalier dans l'ordre. On introduit pour cela le *bord* de l'escalier de G_1 : $\mathcal{M}_1 = (\bigcup_l X_l.\mathcal{B}_1) \setminus \mathcal{B}_1$.
   Pour $m_j \in \mathcal{B}_1$, si $m = X_l.m_j$ est dans $\mathcal{B}_1$ alors la j-ème colonne de M_l est simple à remplir, ne contenant que des 0 sauf un 1 sur la ligne correspondant à m. Sinon, m est dans le bord

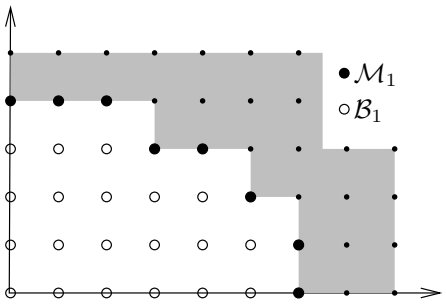
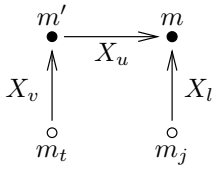


FIGURE 2.8 – L'escalier et son bord.

$\mathcal{M}_1$, et deux cas sont possibles.

(a) Soit m est le monôme de tête d'un élément $f = m + \sum_i c_i m_i$ de G_1. Dans ce cas, $\overline{m} = -\sum_i c_i \overline{m_i}$, et on a $M_{l,ij} = -c_i$.

(b) Sinon, plutôt que de calculer la forme normale de m, on utilise le fait qu'il existe $m' \in \mathcal{M}_1$ et $u \in [\![1; n]\!]$ tel que $m = X_u.m'$. Comme m' est dans le bord, il est lui-même de la forme $X_v.m_t$ pour un certain monôme $m_t \in \mathcal{B}_1$. La forme normale de m' est déjà stockée dans M_v : on a $\overline{m'} = \sum_k M_{v,kt}\overline{m_k}$. Par conséquent, $\overline{m} = mult_u(\overline{m'}) = \sum_i \sum_k M_{u,ik}M_{v,kt}\overline{m_i}$, et $M_{l,ij} = \sum_k M_{u,ik}M_{v,kt}$, i.e. la j-ème colonne de M_l est obtenue en faisant le produit de la t-ième colonne de M_v par la matrice M_u ; si l'on parcourt les monômes de la forme $X_l.m_j$ dans l'ordre croissant pour $\prec_1$ toutes ces valeurs ont déjà été calculées.



2. On construit ensuite les ensembles $\mathcal{B}_2$ et G_2 pour l'ordre $\prec_2$, en partant de $\mathcal{B}_2 = \{1\}$ et $G_2 = \emptyset$. Pour chaque monôme $b_j \in \mathcal{B}_2$, on stocke les coefficients de sa classe dans A : $\overline{b}_j = \sum_i c_{ij}\overline{m_i}$. À chaque étape, on sélectionne le plus petit monôme qui n'est ni dans $\mathcal{B}_2$ ni le multiple d'un terme de tête d'un élément de G_2 ; il est forcément de la forme $b = X_l.b_j$ pour un certain $b_j \in \mathcal{B}_2$. On a donc $\overline{b} = mult_l(\overline{b}_j) = \sum_i \sum_k M_{l,ik}c_{kj}\overline{m_i}$. Deux cas sont alors possibles.

   (a) Soit la famille $(\overline{b}_1, \ldots, \overline{b}_{\#\mathcal{B}_2}, \overline{b})$ est libre, ce qui se teste en calculant la réduction de Gauss de la matrice de leurs coefficients dans $\overline{\mathcal{B}}_1$. Dans ce cas on rajoute b à $\mathcal{B}_2$.

   (b) Soit on obtient une relation de dépendance linéaire $\overline{b} + \sum_i \lambda_i \overline{b}_i = 0$. Dans ce cas $b + \sum_i \lambda_i b_i \in I$, et on rajoute ce polynôme à G_2.

   L'algorithme se termine quand tous les monômes n'étant pas dans $\mathcal{B}_2$ sont divisibles par le terme de tête d'un élément de G_2.

On renvoie à l'article [FGLM93] pour une description plus détaillée et un pseudo-code.

Exemple 2.5.1. *Soient $f_1 = x^2 - x - y^2 + 3y - 2$, $f_2 = xy + x + y^2 - y - 2$ et $f_3 = y^3 - 2y^2 - y + 2$; on vérifie que $G_1 = \{f_1, f_2, f_3\}$ est une base de Gröbner pour l'ordre $\prec_1$ égal à $lex_{y \prec x}$. On cherche à obtenir une base de l'idéal correspondant pour l'ordre $\prec_2$ égal à $grevlex_{x \prec y}$.*

On a $\mathcal{B}_1 = \{1, y, y^2, x\}$ et $\mathcal{M}_1 = \{y^3, xy, xy^2, x^2\}$. On construit simultanément les matrices de multiplications par x et par y dans la base $\overline{\mathcal{B}}_1$ en parcourant les produits variable-monôme dans l'ordre $\prec_1$. Les premiers produits $y.1$ et $y.y$ donnent des monômes dans $\mathcal{B}_1$ et ne posent donc pas

de problème. Pour le produit $y.y^2$, on obtient un élément de $LT(G_1)$, donc on insère les coefficients du polynôme dans M_y. Le produit $x.1$ ne pose pas de problème, ensuite $x.y = y.x$ est dans $LT(G_1)$. À ce stade, on a déjà

$$M_x = \begin{pmatrix} 0 & 2 & * & * \\ 0 & 1 & * & * \\ 0 & -1 & * & * \\ 1 & -1 & * & * \end{pmatrix}, \qquad M_y = \begin{pmatrix} 0 & 0 & -2 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

Le produit suivant est $x.y^2$ qui est dans $\mathcal{M}_1$ mais pas $LT(G_1)$; on a $x.y^2 = y.(xy)$, où $xy \in \mathcal{M}_1$ a déjà été obtenu comme produit $x.y$. On a alors $\overline{xy^2} = M_y(\overline{xy}) = M_y\,{}^t\begin{pmatrix}2 & 1 & -1 & -1\end{pmatrix} = {}^t\begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix}$. Le dernier produit est $x.x$ qui est dans $LT(G_1)$. Finalement,

$$M_x = \begin{pmatrix} 0 & 2 & 0 & 2 \\ 0 & 1 & 0 & -3 \\ 0 & -1 & 0 & 1 \\ 1 & -1 & 1 & 1 \end{pmatrix}.$$

Dans la deuxième phase, on parcourt les monômes pour l'ordre $\prec_2$ et on calcule les coefficients de leurs classes dans $\overline{\mathcal{B}}_1$. On obtient $\overline{1} = {}^t\begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}$, $\overline{x} = M_x.\overline{1} = {}^t\begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix}$, $\overline{y} = M_y.\overline{1} = {}^t\begin{pmatrix}0 & 1 & 0 & 0\end{pmatrix}$, $\overline{x^2} = M_x.\overline{x} = {}^t\begin{pmatrix}2 & -3 & 1 & 1\end{pmatrix}$, et on vérifie que ces 4 vecteurs forment une famille libre ; à ce stade, on a $\mathcal{B}_2 = \{1, x, y, x^2\}$ et $G_2 = \emptyset$. Le monôme suivant est xy et on a $\overline{xy} = M_x.\overline{y} = {}^t\begin{pmatrix}2 & 1 & -1 & -1\end{pmatrix}$; on obtient une relation de dépendance $\overline{xy} = 4.\overline{1} - 2.\overline{y} - \overline{x^2}$, et on ajoute à G_2 le polynôme $xy + x^2 + 2y - 4$. Le monôme suivant est y^2 et on a $\overline{y^2} = M_y.\overline{y} = {}^t\begin{pmatrix}0 & 0 & 1 & 0\end{pmatrix}$; on obtient une relation de dépendance $\overline{y^2} = -2.\overline{1} - \overline{x} + 3.\overline{y} + \overline{x^2}$, et on ajoute à G_2 le polynôme $y^2 - x^2 - 3y + x + 2$. Le dernier monôme est x^3 et on a $\overline{x^3} = M_x.\overline{x^2} = {}^t\begin{pmatrix}-4 & -6 & 4 & 7\end{pmatrix}$; on obtient une relation de dépendance $\overline{x^3} = -12.\overline{1} + 3.\overline{x} + 6.\overline{y} + 4.\overline{x^2}$, et on ajoute à G_2 le polynôme $x^3 - 4x^2 - 6y - 3x + 12$. Finalement la base pour l'ordre $\prec_2$ est

$$G_2 = \{xy + x^2 + 2y - 4, y^2 - x^2 - 3y + x + 2, x^3 - 4x^2 - 6y - 3x + 12\}.$$

2.5.2 Analyse et complexité

Durant la première phase, on construit n matrices de taille $D \times D$ correspondant à la multiplication par $X_1, \ldots, X_n$ dans $A = Vect(\overline{m}_1, \ldots, \overline{m}_D)$. On a vu que le calcul de chaque coefficient coûte au plus D produits et additions, donc la complexité de cette phase est en $O(nD^3)$ opérations dans $\mathbb{K}$.

Pour tester les dépendances linéaires dans la deuxième phase, il est évident qu'il ne faut pas recalculer la forme échelon de la matrice des coefficients $(c_{ij})_{\substack{1 \le i \le D \\ 1 \le j \le \#\mathcal{B}_2}}$ à chaque fois, mais conserver sa forme échelon d'une fois sur l'autre (ainsi que la matrice de transformation associée). À chaque étape, on doit calculer le produit d'une des matrices de multiplication par un vecteur, ce qui se fait en $O(D^2)$ opérations arithmétiques, puis faire une réduction gaussienne de la matrice réduite de l'étape précédente à laquelle on a rajouté une ligne, ce qui se fait en $O(D^2)$ opérations. Comme il y a au plus nD étapes, la complexité de cette deuxième phase est encore en $O(nD^3)$ opérations dans $\mathbb{K}$.

Chapitre 3

Un algorithme adapté à la résolution de nombreux systèmes similaires

Dans ce chapitre, on s'intéresse à la résolution de familles de systèmes polynomiaux instances d'un même système paramétré générique. Ces familles apparaissent naturellement en cryptanalyse algébrique, lorsque l'on tente de modéliser une primitive cryptographique sous forme de systèmes polynomiaux, de telle sorte que leurs solutions fournissent l'information secrète du cryptosystème. Les systèmes considérés sont alors très souvent de la même forme (mêmes degrés, nombre de variables, symétries...) et leurs coefficients sont soit aléatoires soit dépendants d'un petit nombre de paramètres. Comme le succès de l'attaque dépend de la faisabilité du calcul en temps raisonnable des racines de chacun de ces systèmes "similaires", il apparaît plus intéressant dans ce contexte d'utiliser des algorithmes permettant d'exploiter l'information commune pour accélérer la résolution, plutôt que des algorithmes généraux tels que F4 ou F5 qui sont conçus pour ne résoudre qu'un seul système à la fois.

Après avoir précisé la notion de systèmes similaires, on donne quelques rappels dans la première section sur les techniques existantes pour la résolution de tels systèmes dans le cas rationnel. Parmi celles-ci, on retiendra l'idée de Traverso [Tra89] consistant à utiliser les "traces" d'un premier calcul de base de Gröbner avec l'algorithme de Buchberger pour accélérer les calculs suivants, idée reprise dans [ABF09] pour le décodage de certains codes binaires. On explique alors dans la section suivante comment adapter cette technique au cas de l'algorithme F4, en proposant une variante dont l'intérêt est double : on évite a priori toutes les réductions à zéro à la manière de F5, tout en restant conceptuellement aussi simple et efficace que l'algorithme original F4. Ce nouvel algorithme étant par nature probabiliste, on en donne une analyse détaillée en section 3.2 qui permet de déduire des bornes sur la probabilité de succès du calcul d'une base de Gröbner ; des tests de correction pour vérifier la validité du calcul sont également explicités dans cette section. On donne enfin une comparaison des complexités de la variante de F4 et de F5 ainsi que de nombreux exemples d'applications en dernière section.

3.1 Systèmes paramétrés

On rappelle tout d'abord le cadre mathématique utilisé pour définir précisément ce que l'on entend par systèmes polynomiaux similaires, ainsi que les méthodes existantes pour résoudre ce type de systèmes.

Définition 3.1.1. *Soit $V \subset \mathbb{K}^\ell$ une variété algébrique.*

(i) On appelle système paramétré générique *un système $\{F_1, \ldots, F_r\} \in K(V)[X_1, \ldots, X_n]$ de polynômes dont les coefficients sont des polynômes en les paramètres $y = (y_1, \ldots, y_\ell) \in V$.*

(ii) Une instance aléatoire $\{f_1, \ldots, f_r\}$ *du sytème générique $\{F_1, \ldots, F_r\}$ est un système obtenu en évaluant (ou "spécialisant") $y \in V$.*

Deux systèmes sont dits similaires *s'ils sont instances d'un même système paramétré générique.*

Pour modéliser le comportement des polynômes à coefficients aléatoires dans un corps donné, on introduit également la notion de polynôme générique :

Définition 3.1.2. *Un* polynôme générique *F de degré d en n variables $X_1, \ldots, X_n$ est un polynôme à coefficients dans $\mathbb{K}[\{Y_{i_1,\ldots,i_n}\}_{i_1+\ldots+i_n \leq d}]$ de la forme*

$$F = \sum Y_{i_1,\ldots,i_n} X_1^{i_1} \ldots X_n^{i_n}.$$

Un polynôme générique est donc un cas particulier de polynôme paramétré générique, dans la mesure où ses coefficients jouent le rôle des paramètres. Il est assez naturel d'introduire ces polynômes, puisque l'on s'attend en pratique à ce que les calculs de bases de Gröbner de systèmes polynomiaux à coefficients aléatoires soient très proches de ceux que l'on obtiendrait pour le système de polynômes génériques correspondants.

D'un point de vue théorique, il est possible de calculer une base de Gröbner de la famille paramétrée qui a la propriété de rester une base de Gröbner pour toutes les spécialisations des paramètres $y = (y_1, \ldots, y_\ell)$; ce type de bases est appelé "comprehensive Gröbner basis" [Wei92]. Malheureusement en pratique, le coût du calcul d'une telle base, ou même simplement de l'évaluation des paramètres, est en général prohibitif, particulièrement pour les systèmes issus de la cryptanalyse algébrique. Une autre possibilité pour accélérer les calculs de bases de Gröbner consiste à exploiter ceux obtenus pour une première instance. Cette approche a été proposée et analysée pour la première fois par Traverso [Tra89] en 1988, dans le contexte des calculs de bases de Gröbner avec l'algorithme de Buchberger pour des systèmes polynomiaux à coefficients rationnels. Pour ce type de systèmes, une technique assez courante consiste en effet à effectuer le calcul de la base de Gröbner du système réduit modulo un nombre premier, afin d'éviter le "phénomène d'explosion" de la taille des coefficients observé lors du calcul dans le cas rationnel. L'idée de Traverso est alors de conserver une "trace" des opérations utiles (celles qui ne produisent pas des réductions à zéro) faites durant un premier calcul de la base de Gröbner pour le système réduit modulo un nombre premier p. On peut alors tenter de calculer une base de Gröbner du système rationnel en appliquant ces opérations soit directement au système initial ("Gröbner trace lifting"), soit à différents systèmes polynomiaux obtenus en réduisant le système initial modulo plusieurs nombres premiers distincts, pour déduire ensuite le résultat avec un théorème type restes chinois ("Gröbner trace modular algorithm"). Bien entendu, l'algorithme proposé par Traverso est de nature probabiliste et ne retournera pas nécessairement un calcul correct. Son analyse reste ciblée sur le cas de l'algorithme de Buchberger et les probabilités d'échec données dans [Tra89] sont estimées uniquement dans le cas rationnel. On va voir qu'en appliquant ces idées à l'algorithme F4, on peut faire une analyse plus poussée de la probabilité de réussite du calcul grâce à la structure matricielle utilisée.

3.2 La variante de F4

On donne dans cette section une description complète de la variante F4 proposée dans [JV11b]. Soient $V \subset \mathbb{K}^l$ une variété et $F_1, \ldots, F_r \in \mathbb{K}[V][X_1, \ldots, X_n]$ un système paramétré générique de

polynômes ; on s'intéresse aux calculs des bases de Gröbner d'instances aléatoires de ce système.

3.2.1 Description de l'algorithme

L'objectif de la variante de F4 proposée est d'éliminer les réductions à zéro qui s'opèrent malgré l'utilisation des critères de Buchberger. Ces réductions correspondent à des relations de dépendance linéaire entre certaines lignes de la matrice type Macaulay qui est construite à chaque étape de l'algorithme à partir d'un sous-ensemble de paires sélectionnées à réduire simultanément. Si l'on détermine, lors d'un premier calcul sur une instance aléatoire, les lignes "redondantes" qui produisent ces réductions à zéro, on peut espérer que ces mêmes lignes soient également redondantes dans les calculs suivants.

L'approche consiste donc à faire tout d'abord un précalcul sur une première instance aléatoire en appliquant l'algorithme F4 avec les modifications suivantes :

- à chaque étape (autrement dit à chaque fois que l'on sélectionne un sous-ensemble de paires critiques à traiter simultanément), on enregistre dans une liste L tous les multiples (u_i, f_i) qui proviennent des paires critiques ;
- durant la phase de calcul de la forme échelon, pour chaque réduction à zéro qui s'opère, on enlève de la liste L un multiple bien choisi.

Il faut faire attention à la compatibilité des choix des multiples que l'on supprime dans L pour les réductions à zéro. Plus précisément, si M est la matrice construite à une étape du calcul et M' sa forme échelon réduite, on note A la matrice contenant les opérations faites sur les lignes de M pour obtenir M', autrement dit telle que $AM = M'$. Si la matrice M' contient d lignes nulles, alors les lignes correspondantes dans A donnent une base de l'espace des relations de dépendance linéaire des lignes de M. On note A' la matrice composée de ces d lignes de A et $\tilde{A}$ sa réduction de Gauss (où l'on a fait attention à ce que les pivots ne soient choisis que dans les colonnes correspondant à des lignes de M provenant des paires critiques et non du preprocessing). Les colonnes des pivots de $\tilde{A}$ correspondent alors aux lignes de M qu'il est possible de supprimer. Après cette opération, il est encore possible de supprimer les lignes de M "célibataires", i.e. dont le terme de tête est le seul coefficient non nul de sa colonne. Ces lignes correspondent en fait à des polynômes provenant soit de paires dont on a supprimé le deuxième membre, soit du preprocessing préalablement effectué pour les lignes supprimées. Ce précalcul est assuré par la sous-routine `F4Precomp` dont le pseudo-code est donné en algorithme 15 ; l'élimination des multiples inutiles est assurée par les lignes de code 20 à 24 de cet algorithme. Cet algorithme de précalcul est très proche de F4 dont le pseudo-code est donné en algorithme 5. On a simplement introduit la liste de couples L telle que $L[i]$ contienne tous les multiples utiles au calcul à l'étape i de l'algorithme. Chaque multiple de polynôme est représenté par un couple (m, n) où m est un monôme et n est l'indice d'un polynôme dans une liste globale G, qui est progressivement reconstruite par `F4Remake`. La fonction `Index`(g, G) retourne l'indice i tel que $G[i] = g$. Afin d'éliminer les multiples inutiles à chaque étape, on introduit la liste L_{tmp} qui stocke temporairement les couples intervenant dans une étape du calcul, ainsi que les matrices M', A, A' et $\tilde{A}$ décrites précédemment. Enfin, la liste globale G_{min} est la base minimale associée à G. Dans ce pseudo-code, on détaille également l'implantation faite de la fonction `Simplify` déjà présentée dans sa version originale en algorithme 7 ; *TabSimplify* est le tableau d'arbres dont les noeuds sont étiquetés par des polynômes, et les arêtes par des variables, il est mis à jour à la fin de chaque calcul de forme échelon de la matrice par la nouvelle fonction `Postprocessing` présentée en algorithme 17.

Une fois le précalcul fait, les bases de Gröbner des autres instances aléatoires du système paramétrique s'obtiennent en appliquant l'algorithme F4 avec les modifications suivantes :

Algorithme 15: F4Precomp

Entrées : $I = \langle f_1, \ldots, f_r \rangle \subset \mathbb{K}[X_1, \ldots, X_n]$
Sortie : Une liste de listes de couples monôme/indice $(m, n) \in T \times \mathbb{N}$

1. $G \leftarrow [\,]$; $G_{min} \leftarrow [\,]$; CP $\leftarrow \emptyset$; $TabSimplify \leftarrow [\,]$; $L \leftarrow [\,]$;
2. **pour** $i = 1$ *à* r **faire** $G[i] \leftarrow f_i$; $TabSimplify[i] \leftarrow$ Noeud(f_i); `Update(`f_i`)`;
3. $step \leftarrow 1$;
4. **tant que** CP $\neq \emptyset$ **faire**
5. CP$_{sel}$ $\leftarrow$ `Sel(`CP`)`;
6. $F \leftarrow [\,]$; $LM(F) \leftarrow \emptyset$; $T(F) \leftarrow \emptyset$; $L[step] \leftarrow [\,]$; $L_{tmp} \leftarrow [\,]$;
7. **pour tout** $pair = (lcm, t_1, g_1, t_2, g_2) \in$ CP$_{sel}$ **faire**
8. **pour** $k = 1$ *à* 2 **faire**
9. $ind \leftarrow$ `Index(`g_k, G`)`;
10. **si** $(t, ind) \notin L_{tmp}$ **alors**
11. Apposer$(L_{tmp}, (t_k, ind))$;
12. $f \leftarrow$`Simplify(`t_k, ind`)`;
13. Apposer(F, f);
14. $LM(F) \leftarrow LM(F) \cup \{LM(f)\}$;
15. $T(F) \leftarrow T(F) \cup \{$monômes de $f\}$;
16. `Preprocessing(`$F, T(F), LM(F)$`)`;
17. $M \leftarrow$ matrice dont les lignes sont les polynômes de F;
18. $(M'|A) \leftarrow$ forme échelon réduite de $(M|I_{\#F})$; `//` en particulier $AM = M'$
19. $rang \leftarrow$`Postprocessing(`$M', LM(F)$`)`;
20. **si** $rang < \#F$ **alors**
21. $A' \leftarrow A[rang + 1..\#F][1..\#L_{tmp}]$; $\tilde{A} \leftarrow$ forme échelon réduite de A';
22. $C \leftarrow \{c \in [\![1, \#L_{tmp}]\!] : c$ n'est pas une colonne pivot de $\tilde{A}\}$;
23. **pour** $j \in C$ **faire**
24. **si** $\exists k \in C, k \neq j$ *et* $LM(F[k]) = LM(F[j])$ **alors** Apposer$(L[step], L_{tmp}[j])$;
25. **sinon** $L[step] \leftarrow L_{tmp}$;
26. $step \leftarrow step + 1$;
27. **retourner** G;

- on ne maintient plus de liste CP des paires critiques en attente de traitement ;
- à chaque itération de l'algorithme, au lieu de sélectionner un sous-ensemble de paires de CP à réduire simultanément, on remplit directement la matrice avec les multiples utiles (u_i, f_i) stockés dans la liste L produite par le précalcul.

Ces calculs de bases de Gröbner sont assurés par la deuxième sous-routine `F4Remake` dont on donne le pseudo-code en algorithme 18. On note que cet algorithme est probabiliste de type *Monte-Carlo* : il est possible, lorsque l'instance aléatoire choisie n'a pas le "comportement attendu", que l'algorithme retourne une base de l'idéal correspondant qui ne soit pas une base de Gröbner. On précisera, dans la prochaine section, la probabilité de réussite de `F4Remake` ainsi que les tests de correction à ajouter pour s'assurer de la validité du calcul. Le cas échéant, on expliquera comment il est possible lorsque `F4Remake` échoue, de poursuivre les calculs avec l'algorithme F4 classique. L'algorithme `F4Remake` fait appel aux fonctions `Simplify`, `Preprocessing` et `Postprocessing` décrites en algorithmes 16, 8 et 17. Puisque l'on n'utilise plus de paires critiques, la fonction `Update` se simplifie considérablement, et peut être remplacée par la fonction `Update2` décrite en algorithme 19.

Algorithme 16: Simplify

Entrées : $t \in T, i_0 \in \mathbb{N}$
Sortie : $p \in \mathbb{K}[X_1, \ldots, X_n]$

1. $noeud \leftarrow TabSimplify[i_0]$; $p \leftarrow$ Etiquette($noeud$); $m \leftarrow t$;
2. **pour** $i = n$ *à* 1 **faire**
3. **tant que** $X_i | t$ **faire**
4. **si** *il existe une arête* $(noeud \xrightarrow{X_i} noeud')$ **alors** $g \leftarrow$ Etiquette($noeud'$);
5. **si** $g \neq 0$ **alors** $p \leftarrow g$; $m \leftarrow t/X_i$;
6. **sinon**
7. $noeud' \leftarrow$ nouveau noeud d'étiquette 0;
8. AjouterArete($noeud \xrightarrow{X_i} noeud'$);
9. $noeud \leftarrow noeud'$; $t \leftarrow t/X_i$;
10. **si** Etiquette($noeud$) $= 0$ **alors** Etiquette($noeud$) $\leftarrow m \cdot p$;
11. **retourner** Etiquette($noeud$);

Algorithme 17: Postprocessing

Entrées :
- M matrice sous forme échelon réduite contenant $\#F$ lignes
- $LM(F)$ un ensemble ordonné de monômes

Sortie : le rang de la matrice M

1. **pour** $i = 1$ *à* $\#F$ **faire**
2. $f \leftarrow M[i]$;
3. **si** $f = 0$ **alors break**;
4. **si** $LM(f) \notin LM(F)$ **alors**
5. Apposer(G, f);
6. `Update` (f);
7. $TabSimplify[\#G] \leftarrow$ Noeud(f);
8. **sinon**
9. **pour tous les** $g \in G_{min}$ *tel que* $LM(g) | LM(f)$ **faire**
10. $m \leftarrow LM(f) / LM(g)$;
11. $noeud \leftarrow TabSimplify[$`Index` $(g, G)]$;
12. **pour** $j = n$ *à* 1 **faire**
13. **tant que** $X_j | m$ **faire**
14. **si** *il n'existe pas d'arête* $(noeud \xrightarrow{X_j} noeud')$ **alors**
15. $noeud' \leftarrow$ nouveau noeud d'étiquette 0;
16. AjouterArete($noeud \xrightarrow{X_j} noeud'$);
17. $noeud \leftarrow noeud'$; $m \leftarrow m/X_j$;
18. Etiquette($noeud$) $\leftarrow f$;
19. **retourner** $i - 1$;

On remarque qu'il serait également possible d'enregistrer durant le précalcul une liste de tous les multiples de polynômes utiles apparaissant dans la matrice M à chaque étape du calcul, plutôt que simplement ceux qui proviennent des paires critiques. Ceci permettrait notamment de ne pas faire

Algorithme 18: F4Remake

Entrées : $f_1, \ldots, f_r \in \mathbb{K}[X_1, \ldots, X_n]$, une liste L de listes de couples $(m, n) \in T \times \mathbb{N}$
Sortie : G_{min}, la base de Gröbner minimale réduite de $f_1, \ldots, f_r$

1. $G \leftarrow [\,]$, $G_{min} \leftarrow \emptyset$; $TabSimplify \leftarrow [\,]$;
2. **pour** $i = 1$ *à* r **faire**
3. $\quad G[i] \leftarrow f_i$;
4. $\quad TabSimplify[i] \leftarrow \text{Noeud}(f_i)$;
5. $\quad$ `Update2(`f_i`)`;
6. **pour** $step = 1$ *à* $\#L$ **faire**
7. $\quad F \leftarrow [\,]$; $LM(F) \leftarrow \emptyset$; $T(F) \leftarrow \emptyset$;
8. $\quad$ **pour tous les** $(m, n) \in L[step]$ **faire**
9. $\quad\quad$ **si** $n > \#G$ **alors** échec ! **exit**;
10. $\quad\quad f \leftarrow$`Simplify(`m, n`)`; Apposer(F, f);
11. $\quad\quad LM(F) \leftarrow LM(F) \cup \{LM(f)\}$;
12. $\quad\quad T(F) \leftarrow T(F) \cup \{\text{monômes de } f\}$;
13. $\quad$ `Preprocessing(`$F, T(F), LM(F)$`)`;
14. $\quad M \leftarrow$ matrice dont les lignes sont les polynômes de F;
15. $\quad M' \leftarrow$ forme échelon réduite de M;
16. $\quad$ `Postprocessing(`$M', LM(F)$`)`;
17. **retourner** `InterReduce(`G_{min}`)`;

Algorithme 19: Update2

Entrées : $f \in \mathbb{K}[X_1, \ldots, X_n]$

1. **si** $\nexists g \in G_{min}$ *tel que* $LM(g) | LM(f)$ **alors**
2. $\quad$ **pour tous les** $g \in G_{min}$ **faire**
3. $\quad\quad$ **si** $LM(f) | LM(g)$ **alors** $G_{min} \leftarrow G_{min} \setminus \{g\}$;
4. $\quad G_{min} \leftarrow G_{min} \cup \{f\}$;

le preprocessing dans les calculs suivants, mais augmenterait toutefois considérablement la taille de la sortie de `F4Precomp`. En pratique, la phase de preprocessing étant nettement plus courte que la phase de calcul de la forme échelon, le gain apporté par cette modification n'est pas manifeste ; celle-ci n'est donc pas proposée dans le pseudo-code. À ce sujet, il est à noter qu'une autre approche pour le calcul de bases de Gröbner utilisant des "traces" est décrite dans [ABF09], dans le contexte de décodage de codes cycliques binaires : plutôt que d'enregistrer dans un fichier l'information sur les polynômes utiles durant le calcul de la base de Gröbner, le précalcul génère directement un programme (en langage C) qui contient les instructions à exécuter pour obtenir les bases de Gröbner des systèmes suivants. Cette technique de génération de code est plus complexe que notre approche, mais devrait être plus rapide même lorsque l'on prend en compte le temps de compilation du programme de sortie.

3.2.2 Comportement générique et probabilité de réussite

Si $\{f_1^0, \ldots, f_r^0\}$ et $\{f_1, \ldots, f_r\}$ sont deux systèmes similaires instances du système paramétrique générique $\{F_1, \ldots, F_r\}$, on souhaite estimer la probabilité de réussite de l'algorithme `F4Remake` pour le deuxième système, le précalcul ayant été fait sur le premier. À cet effet, on définit la notion

de *comportement générique* pour une instance aléatoire d'un système paramétré.

D'un point de vue théorique, on peut toujours calculer une base de Gröbner de l'idéal engendré par $\{F_1, \ldots, F_r\}$ dans $\mathbb{K}(V)[X_1, \ldots, X_n]$ avec l'algorithme F4 (même si en pratique, on s'attend à une "explosion" de la taille des coefficients sous forme de fractions rationnelles). On dit que $\{f_1, \ldots, f_r\}$ se comporte *génériquement* si durant le calcul de la base de Gröbner de l'idéal associé avec l'algorithme F4 :

- le même nombre d'étapes que pour le calcul du système générique est considéré ;
- à chaque étape, le même nombre de nouveaux générateurs apparaît avec les mêmes monômes de tête.

Avec cette définition, on peut donner une condition algébrique pour le comportement générique d'un système. On suppose que le système $\{f_1, \ldots, f_r\}$ s'est comporté génériquement jusqu'à la $(i-1)$-ième étape du calcul. En particulier, ceci implique que toutes les paires critiques impliquées à l'étape i du calcul pour le système générique et l'instance aléatoire sont *similaires* au sens suivant :

$$(lcm, u_1, p_1, u_2, p_2) \text{ est } \textit{similaire} \text{ à } (lcm', u_1', p_1', u_2', p_2') \text{ si } LM(p_i) = LM(p_i') \text{ pour } i = 1, 2$$

(en particulier, $u_i = u_i'$ et $lcm = lcm'$). On considère maintenant les matrices M_g et M construites par F4 à l'étape i pour le système générique et l'instance aléatoire respectivement. Il est possible qu'après le préprocessing M soit plus petite que M_g, mais on supposera pour la démonstration que les multiples des polynômes manquants ont été ajoutés à M ; bien entendu, ces lignes supplémentaires n'influenceront en rien la suite du calcul de la base de Gröbner pour l'instance. Par hypothèse, toutes les lignes de M et M_g vues comme des polynômes ont donc exactement les mêmes monômes de tête ; on note s le nombre de monômes de tête distincts pour chacune des matrices. Comme déjà expliqué dans la section 2.3, on remarque que l'algorithme F4 construit des matrices quasiment triangulaires supérieures, et que donc s est proche du nombre de lignes des matrices M_g ou M. Quitte à faire des échanges de lignes et de colonnes, on peut ramener M_g et M sous la forme présentée en section 2.3.3 :

$$M_g = \overset{LT(M)}{\overbrace{}}\ \overset{s}{\left\{\right.}\left(\begin{array}{c|c} \begin{array}{cc} A_{g,0} & \\ & 0 \end{array} & A_{g,1} \\ \hline A_{g,3} & A_{g,2} \end{array}\right) \qquad M = \left(\begin{array}{c|c} \begin{array}{cc} A_0 & \\ & 0 \end{array} & A_1 \\ \hline A_3 & A_2 \end{array}\right)$$

Les deux matrices ayant les mêmes termes de tête, on peut échelonner les s premières colonnes similairement pour les deux ; soient M_g' et M' les matrices correspondantes.

$$M_g' = \left(\begin{array}{c|c} I_s & B_{g,1} \\ \hline 0 & B_{g,2} \end{array}\right) \qquad M' = \left(\begin{array}{c|c} I_s & B_1' \\ \hline 0 & B_2' \end{array}\right)$$

On note $\tilde{M}_g$ la forme échelon réduite de M_g' et $r = s + \ell$ le rang de cette matrice qui contient éventuellement d lignes nulles. À échange de lignes et de colonnes près dans sa moitié droite, on a :

$$\tilde{M}_g = \left(\begin{array}{c|c|c} I_s & 0 & C_{g,1} \\ \hline 0 & I_\ell & C_{g,2} \\ \hline 0 & 0 & 0 \end{array}\right)$$

où l'on a représenté en vert les colonnes des nouveaux pivots. Si l'on fait les mêmes échanges de colonnes sur M', on obtient la matrice $\tilde{M}$ suivante :

$$\tilde{M} = \left(\begin{array}{c|c|c} I_s & & B_1 \\ \hline 0 & B & B_2 \end{array}\right)$$

où la matrice B contient ℓ colonnes. Si jamais l'un des pivots considérés dans M_g s'annule après spécialisation des paramètres dans la matrice B, les monômes de tête des polynômes générés à cette étape pour l'instance seront distincts de ceux obtenus pour la matrice génératrice ; en effet les monômes correspondant aux colonnes de la matrices B_2 sont par construction tous plus petits que ceux associés aux colonnes de B, on ne peut donc pas espérer faire un échange de colonnes qui compense la perte de ce pivot. Il est donc clair que le système $\{f_1, \ldots, f_r\}$ se comporte génériquement à l'étape i du calcul si et seulement si la matrice B obtenue à cette étape est de rang plein.

Afin de donner des estimées sur la probabilité que cette matrice soit de rang plein à chaque étape, on supposera désormais que l'on travaille sur le corps $\mathbb{K} = \mathbb{F}_q$. On fera de plus les deux hypothèses heuristiques suivantes :

1. les coefficients des matrices B intervenant à chaque étape de l'algorithme sont uniformément distribués dans $\mathbb{F}_q$;
2. les probabilités pour chaque matrice B d'être de rang plein à une étape sont indépendantes.

La première hypothèse semble naturelle pour les systèmes à coefficients aléatoires : à la première étape, les coefficients sont par construction choisis aléatoirement dans $\mathbb{F}_q$, et lors des étapes suivantes, on se convainc que les opérations de mise sous forme échelon sont suffisamment mélangeantes pour uniformiser les coefficients. Elle est plus critiquable pour les instances aléatoires de systèmes paramétrés génériques ; il est par exemple tout à fait possible que certains coefficients des polynômes génériques soient indépendants des paramètres, donc à valeurs dans $\mathbb{F}_q$ (et non $\mathbb{F}_q(V)$) et constants pour chaque spécialisation. Un cas particulier important est celui des systèmes paramétrés génériques dont la partie homogène de plus haut degré a ses coefficients constants : dans ce cas, toutes les instances aléatoires vont se comporter génériquement jusqu'à la première chute de degré. En particulier, la probabilité de succès de notre algorithme peut dans ce cas être meilleure que pour les systèmes aléatoires, même lorsque le corps de base est relativement petit. On renvoie aux exemples de la section 3.3 pour plus de détails. La deuxième hypothèse doit être perçue comme le "pire cas" ; en effet, on peut s'attendre en pratique à ce que si une matrice est de rang plein à une étape, les suivantes tendent à l'être également, c'est-à-dire que les corrélations sont positives. Néanmoins, les estimées qui seront obtenues sous ces hypothèses seront satisfaisantes comme on le verra avec l'exemple de calcul exhaustif donné en section 3.3.2.

Sous les hypothèses 1 et 2, le lemme et le théorème suivants donnent alors la probabilité de succès de notre variante F4 :

Lemme 3.2.1. *Soit $M = (m_{ij}) \in \mathcal{M}_{n,\ell}(\mathbb{F}_q)$, où $n \geq \ell$, une matrice aléatoire, i.e. dont les coefficients m_{ij} sont choisis aléatoirement de façon indépendante et uniforme dans $\mathbb{F}_q$. Alors M est de rang plein avec la probabilité $\prod_{i=n-\ell+1}^{n}(1-q^{-i})$. Une borne inférieure pour cette probabilité est donnée par :*

$$c(q) = \prod_{i=1}^{\infty}(1-q^{-i}) \geq \left(\frac{q-1}{q}\right)^{\frac{q}{q-1}}.$$

En particulier, lorsque q est grand, $c(q) = 1 - 1/q + O(1/q^2)$ et la probabilité pour M d'être de rang plein est très proche de 1.

Pour garantir le comportement générique d'un système, les matrices B intervenant à chaque étape de l'algorithme doivent être de rang plein.

Théorème 3.2.2. *Si l'on suppose que le précalcul a été fait avec* `F4Precomp` *en n_{step} étapes pour un système $f_1^0, \ldots, f_r^0 \in \mathbb{F}_q[X_1, \ldots, X_n]$ qui se comporte génériquement, alors l'algorithme* `F4Remake` *calcule une base de Gröbner d'un système aléatoire $f_1, \ldots, f_r \in \mathbb{F}_q[X_1, \ldots, X_n]$ instance de la même famille paramétrée avec une probabilité qui est heuristiquement plus grande que $c(q)^{n_{step}}$.*

D'après les résultats donnés en section 2.2, le nombre d'étapes n_{step} dans le calcul de la base de Gröbner d'un système générique de polynômes avec l'algorithme F4 pour un ordre monomial gradué, est majoré par le degré de régularité d_{reg} du système homogénéisé, qui est plus petit que la borne de Macaulay $\sum_{i=1}^{r}(\deg(F_i)-1)+1$. Plus la taille du corps $\mathbb{F}_q$ est grande, plus $c(q)$ est proche de 1, donc à degré de régularité fixé, la probabilité de réussite de notre algorithme sera très proche de 1 lorsque le corps de base $\mathbb{F}_q$ est suffisamment grand.

3.2.3 Changement de caractéristique

Une autre application possible de notre algorithme est le calcul de bases de Gröbner de systèmes polynomiaux aléatoires définis sur un grand corps fini, utilisant un précalcul sur un corps fini de petite taille. En effet, même lorsque l'on souhaite calculer la base de Gröbner d'un seul système $f_1, \ldots, f_r \in \mathbb{F}_p[X_1, \ldots, X_n]$, il peut parfois être plus avantageux de faire un précalcul d'un système $f'_1, \ldots, f'_r \in \mathbb{F}_{p'}[X_1, \ldots, X_n]$ tel que $\deg(f_i) = \deg(f'_i)$ pour un petit nombre premier p', et d'utiliser `F4Remake` sur le système initial, plutôt que de calculer directement la base de Gröbner avec l'algorithme F4. Dans ce cas, on ne peut pas appliquer directement les estimées données précédemment pour la probabilité de succès, mais on peut faire une analyse similaire.

On rappelle que pour tout nombre premier p, il existe une application de réduction $\mathbb{Q}[X_1, \ldots, X_n] \to \mathbb{F}_p[X_1, \ldots, X_n]$ bien définie, qui envoie un polynôme P sur sa réduction $\overline{P} = cP \bmod p$, où $c \in \mathbb{Q}$ est tel que $cP \in \mathbb{Z}[X_1, \ldots, X_n]$ soit primitif. Soit $I = \langle f_1, \ldots, f_r \rangle$ un idéal de $\mathbb{Q}[X_1, \ldots, X_n]$ et $\overline{I} = \langle \bar{f}_1, \ldots, \bar{f}_r \rangle$ l'idéal correspondant dans $\mathbb{F}_p[X_1, \ldots, X_n]$; on note $\{g_1, \ldots, g_s\}$ la base de Gröbner minimale réduite de I. Suivant la terminologie de [Ebe83], on dira que

- p est un "lucky prime" si $\{\bar{g}_1, \ldots, \bar{g}_s\}$ est la base de Gröbner minimale réduite de $\overline{I}$,
- p est un "unlucky prime" sinon.

On peut donner une notion plus faible (adaptée de [ST89]) mais plus utile de "F4-unlucky prime" : un nombre premier p est appelé ainsi lorsque le calcul des bases de Gröbner de I et $\overline{I}$ avec F4 diffèrent. En faisant la même analyse qu'en section 3.2.2, on peut montrer que p est "F4-unlucky" si et seulement si l'une des matrices B apparaissant à chacune des étapes de l'algorithme n'est pas de rang plein modulo p, autrement dit si p divise le pgcd des déterminants des sous-matrices

carrées de taille maximale de B. Il n'existe donc qu'un nombre fini de valeurs de p qui sont "F4-unlucky". Avec les mêmes hypothèses que précédemment, on obtient qu'un nombre premier p tiré aléatoirement dans une plage de taille donnée n'est pas "F4-unlucky" avec une probabilité qui devrait heuristiquement être bornée inférieurement par $c(p)^{n_{step}}$.

Ainsi, si l'on veut calculer une base de Gröbner d'un système $f_1, \ldots, f_r \in \mathbb{F}_p[X_1, \ldots, X_n]$ où p est un grand nombre premier, on peut relever ce système à $\mathbb{Q}[X_1, \ldots, X_n]$ et le réduire à $f'_1, \ldots, f'_r \in \mathbb{F}_{p'}[X_1, \ldots, X_n]$ où p' est un petit nombre premier. On exécute alors `F4Precomp` sur ce dernier et on utilise ce précalcul pour obtenir la base de Gröbner du premier système avec `F4Remake`. Ce procédé permet d'acquérir un calcul correct dès que p et p' ne sont pas "F4-unlucky" ; en particulier, p' (bien que devant être le plus petit possible pour que le précalcul soit accéléré au mieux) devra être suffisamment grand pour que la probabilité $c(p')^{n_{step}}$ soit assez proche de 1. En pratique, on doit adopter cette démarche dès que possible : si l'on doit calculer par exemple plusieurs bases de Gröbner d'instances d'une même famille paramétrée sur $\mathbb{F}_q$ avec q grand, le précalcul ne doit pas être fait sur $\mathbb{F}_q$ mais sur un corps plus petit. Cette stratégie sera adoptée dans la plupart des applications données en section 3.3.

3.2.4 Tests de correction

Par soucis de clarté et de concision, le pseudo-code de `F4Remake` donné en algorithme 18 ne présente pas (mis à part la vérification très basique faite en ligne 9) de tests vérifiant si la base retournée est bien une base de Gröbner. On décrit ici quelques tests de correction que l'on pourrait aisément ajouter à ce pseudo-code.

Une première idée consiste à enregistrer, durant le précalcul, la liste des monômes de tête des polynômes créés à chaque étape de l'algorithme, et à vérifier si chacun des nouveaux polynômes apparaissant dans `F4Remake` a bien le monôme de tête attendu. Lorsqu'un terme de tête n'est pas correct ou que le nombre de nouveaux générateurs n'est pas le bon, deux cas peuvent se produire :

- si le terme de tête obtenu ou le nombre de générateurs est plus petit que celui attendu, on sait qu'au moins l'un des pivots dans les matrices calculées par `F4Remake` pour l'instance aléatoire s'est annulé après spécialisation des paramètres ; cette instance aléatoire n'a donc pas un comportement générique ;
- sinon, c'est le précalcul qui est incorrect ; on explique dans la section suivante comment gérer ce cas.

Lorsque l'on détecte un cas de comportement non générique avec `F4Remake`, on peut reprendre les calculs avec l'algorithme F4 en partant de l'étape précédant celle où l'on détecte un comportement non générique ; par contre, il faut lancer l'algorithme F4 sur la base G de l'idéal, et non G_{min} qui n'engendre pas forcément tout l'idéal (voir section 2.1.3).

Un deuxième test pourrait éventuellement être ajouté à la fin de l'exécution de l'algorithme, afin de vérifier si le résultat obtenu (qui est toujours une base de l'idéal) est bien une base de Gröbner. En général, cette vérification potentiellement coûteuse n'est pas nécessaire, tant que l'ensemble des monômes de tête des bases retournés par `F4Precomp` et `F4Remake` sont les mêmes et que l'on suppose que le précalcul s'est comporté génériquement (voir la section suivante pour plus de détails dans le cas contraire). Par ailleurs, si l'idéal considéré est de dimension 0 et de petit degré (comme c'est souvent le cas dans les attaques algébriques), la vérification devient pratiquement immédiate.

3.2.5 Cas d'un précalcul incorrect

Si le premier système considéré a un comportement générique, alors la sortie de `F4Precomp` est correcte ; on a vu que ceci se produit en pratique avec une assez forte probabilité en $c(q)^{n_{step}}$. On considère dans cette section, ce qu'il peut se passer lorsque le précalcul n'est pas correct et comment détecter ce cas défavorable. À cet effet, on analyse ce qu'il se passerait si on lançait l'algorithme `F4Remake` sur le système générique (du moins théoriquement) ; suivant l'analyse de Traverso donnée dans [Tra89], on distingue alors deux cas possibles :

1. L'algorithme `F4Remake` génère une erreur. Dans ce cas, les calculs avec `F4Remake` vont également échouer avec grande probabilité pour la plupart des systèmes suivants. Cette situation est facilement détectable après seulement quelques exécutions ; on peut se référer à [Tra89] pour des estimations larges de la probabilité que l'on ne détecte aucune erreur pour plusieurs systèmes consécutifs. En pratique, lorsqu'une erreur se produit avec `F4Remake`, on a vu précédemment comment distinguer si celle-ci est due à un précalcul incorrect ou au comportement non générique de l'instance aléatoire considérée.
2. L'algorithme `F4Remake` ne génère pas d'erreur, mais le système retourné n'est pas une base de Gröbner de l'idéal considéré. Cette situation, bien qu'improbable, est plus difficile à détecter : on doit tester si les sorties des premières exécutions de l'algorithme `F4Remake` sont bien des bases de Gröbner. Si ce n'est pas le cas pour l'un des systèmes, alors le précalcul est incorrect.

Pour s'assurer de la validité du précalcul, une alternative consiste à lancer `F4Precomp` sur plusieurs systèmes, puis à tester si les listes en sortie coïncident. Si ce n'est pas le cas, on a intérêt à choisir la liste créée la plus courante, dans la mesure où la probabilité qu'une majorité de précalculs soient incorrects est très faible. Bien entendu, lorsque la valeur de $c(q)^{n_{step}}$ est très proche de 1, la probabilité d'un précalcul incorrect est suffisamment faible pour qu'il n'y ait pas lieu de s'inquiéter de cette situation.

3.2.6 Complexité

On a vu qu'il est assez difficile d'obtenir de bonnes estimées pour la complexité des calculs de bases de Gröbner, surtout lorsque l'on choisit l'approche de Buchberger. Dans le cas de l'algorithme `F4Remake`, il est quand même possible de donner une borne supérieure de la complexité :

Proposition 3.2.3. *Le nombre d'opérations effectuées par l'algorithme* `F4Remake` *pour le calcul de la base de Gröbner d'un système de polynômes semi-réguliers sur* $\mathbb{K}[X_1, \ldots, X_n]$ *est borné par*

$$O\left(\binom{d_{reg}+n}{n}^{\omega}\right),$$

où d_{reg} *est le degré de régularité du système homogénéisé correspondant et* ω *est la constante intervenant dans la complexité du produit matriciel.*

Cette borne supérieure s'obtient avec une analyse similaire à celle donnée pour l'algorithme F5 présenté en section 2.4.3 : le calcul fait par `F4Remake` peut être ramené à celui de la forme échelon d'une matrice de Macaulay du système homogénéisé dont on aurait supprimé les lignes inutiles. Dans le cas d'un système générique, le degré de la matrice de Macaulay à considérer est égal au degré de régularité du système homogénéisé correspondant. On note cependant que cette borne est loin d'être optimale, et ne reflète pas les différences de performances entre `F4Remake` et F5. Le premier a en effet deux principaux avantages :

- les polynômes engendrés sont complètement réduits à chaque étape, contrairement à ceux apparaissant dans F5 qui doivent respecter des conditions de compatibilité de signatures ;
- on s'affranchit de la nature incrémentale de F5, qui ne permet pas d'utiliser l'information portée par les derniers polynômes décrivant l'idéal en entrée alors qu'elle pourrait accélérer les premières étapes du calcul de la base de Gröbner.

Par conséquent, il est plus intéressant d'utiliser notre variante de l'algorithme F4 dès lors que l'on doit faire plusieurs calculs de bases de Gröbner, sur un corps de taille suffisamment grande, d'instances issues d'une même famille de systèmes.

3.3 Applications

On donne dans cette section plusieurs exemples de calculs de bases de Gröbner de systèmes issus d'attaques algébriques ou de bancs d'essai classiques. Pour chacun des exemples choisis, on compare les performances de `F4Remake` avec celles de notre implantation de F4 (voir section 2.3) utilisant les mêmes structures et primitives, ainsi que celles du logiciel propriétaire Magma (V2.15-15) [BCP97] qui est probablement la meilleure implantation disponible à l'heure actuelle pour les corps finis considérés. Sauf mention contraire, tous les tests ont été réalisés sur un cœur d'un processeur Intel Core 2 Duo à 2.6 GHz ; les temps sont donnés en secondes.

3.3.1 Calcul d'indices

On s'intéresse à la résolution de systèmes polynomiaux issus de l'attaque du logarithme discret d'une courbe elliptique $E(\mathbb{F}_{p^5})$ définie sur une extension d'un corps premier de degré 5, selon la méthode présentée en section 7.3.1. Cette attaque repose sur une méthode de calcul d'indices, pour laquelle la recherche d'une relation se transcrit en la résolution de systèmes polynomiaux à plusieurs variables surdéterminés. Plus précisément, pour chaque tentative de décomposition d'un point $R \in E(\mathbb{F}_{p^5})$ aléatoire dans la base de factorisation, on crée un système de polynômes dont les coefficients dépendent polynomialement de l'abscisse (aléatoire) de R et de l'équation de la courbe elliptique. On peut donc voir chacun de ces systèmes de polynômes comme une instance aléatoire d'une famille paramétrée générique. Notre algorithme est particulièrement adapté à cette attaque, puisque l'on doit résoudre de l'ordre de $4!p^2$ systèmes de cette forme pour obtenir suffisamment de relations pour attaquer le logarithme discret. De plus, comme dans les applications cryptographiques on doit prendre p suffisamment grand, la probabilité de succès de notre variante de F4 est très proche de 1.

Les systèmes à résoudre sont composés de 5 équations définies sur $\mathbb{F}_p$, en 4 variables de degré total 8. On donne les temps de calculs des bases de Gröbner des idéaux correspondants sur différents corps premiers de tailles respectives 8, 16, 25 et 32 bits. La probabilité d'échec du calcul avec l'algorithme `F4Remake` est estimée dans chaque cas, en supposant que les systèmes sont aléatoires et sachant que le calcul prend 29 étapes. On note que ceci correspond bien au nombre d'étapes attendu : le système surdéterminé de départ étant composé de 5 équations de degré total 8, on trouve avec la borne de Macaulay que le calcul de la base de Gröbner du système homogénéisé correspondant ne dépasse pas le degré 36, ce qui fait bien 29 étapes au total.

Dans cette application, on effectue un seul précalcul sur un corps de caractéristique petite ($|p|_2 = 8$ bits) permettant d'obtenir la liste des multiples des polynômes utiles pour tous les autres cas. Les temps d'exécution obtenus pour `F4Precomp` pour les corps de caractéristique 16, 25 et 32 bits

taille de p	proba. échec estimée	temps exécution (implantation en C)			rapport temps F4/F4Remake	temps exécution F4 (magma)
		F4Precomp	F4Remake	F4		
8 bits	0.11	8.963	2.844	5.903	2.1	9.660
16 bits	4.4×10^{-4}	(19.07)	3.990	9.758	2.4	9.870
25 bits	2.4×10^{-6}	(32.98)	4.942	16.77	3.4	118.8
32 bits	5.8×10^{-9}	(44.33)	8.444	24.56	2.9	1046

Étape	degré	tailles des matrices		rapport de tailles des matrices
		F4Remake	F4	
14	17	1062×3072	1597×3207	1.6
15	16	1048×2798	1853×2999	1.9
16	15	992×2462	2001×2711	2.2
17	14	903×2093	2019×2369	2.5
18	13	794×1720	1930×2000	2.8

TABLE 3.1 – Résultats expérimentaux sur $E(\mathbb{F}_{p^5})$

sont donc juste donnés à titre indicatifs. On constate que le temps de précalcul est immédiatement amorti lorsque l'on doit résoudre ne serait-ce qu'un seul système en grande caractéristique. Il aurait été hasardeux par contre de tenter d'accélérer le précalcul en l'exécutant sur un corps de caractéristique plus petite, dans la mesure où la probabilité d'échec augmente lorsque la taille du corps diminue.

On a également utilisé notre propre implantation de l'algorithme F5 [1] (utilisant les mêmes primitives en C) pour résoudre ce système. De façon surprenante, la taille de la base de Gröbner avant la dernière étape (avant la minimisation) est très importante : elle contient 17 249 polynômes étiquetés alors que les deux versions de F4 ne construisent jamais plus de 2 789 polynômes à la fois et produisent des bases contenant au plus 329 générateurs. La situation est meilleure, mais reste toutefois moins bonne qu'avec F4, si l'on adopte la stratégie par signature croissante décrite en section 2.4.4 : seulement 4 238 polynômes étiquetés sont créés par F5 dans ce cas. On note que ces chiffres ne dépendent pas des détails d'implantations. Ce nombre important de polynômes construits par F5 a des conséquences évidentes sur les performances ; en particulier, les temps obtenus sur ces systèmes sont largement moins bons que ceux obtenus avec F4 ou la variante. Ceci montre que l'algorithme F5 tel qu'il est décrit dans [Fau02] n'est pas le mieux adapté à la résolution de ce type de systèmes.

3.3.2 Approche hybride

L'approche hybride proposée dans [BFP10] est une méthode de résolution de systèmes à plusieurs variables sur corps finis reposant sur un compromis entre la recherche exhaustive et le calcul de bases de Gröbner. L'idée de départ est que lorsque l'on cherche à calculer une solution d'un

1. Les implantations de F5 actuellement disponibles en ligne (comme par exemple [AP10] implantée dans Sage [Ste08]) n'ont pas pu terminer le calcul.

système $f_1, \ldots, f_r \in \mathbb{K}[X_1, \ldots, X_n]$, il peut être parfois plus rapide de deviner la valeur d'un petit nombre de variables $X_1, \ldots, X_k$: pour tous les k-uplets possibles $(x_1, \ldots, x_k)$, on calcule la base de Gröbner du système spécialisé correspondant $f_1(x_1, \ldots, x_k), \ldots, f_r(x_1, \ldots, x_k) \in \mathbb{K}[X_{k+1}, \ldots, X_n]$ jusqu'à ce qu'une solution soit trouvée. L'intérêt de cette approche est que le calcul de la base de Gröbner du système spécialisé est plus simple que celui pour le système initial.

Les familles paramétrées introduites en section 3.1 permettent donc de modéliser parfaitement les systèmes issus de cette approche. Pour que la recherche exhaustive sur un certain nombre de variables reste raisonnable, il est nécessaire que le corps de définition soit relativement petit ; il doit être néanmoins suffisamment grand pour que la probabilité de succès de `F4Remake` soit bonne. On note par ailleurs que, lorsque le bon choix de valeurs pour le k-uplet $(x_1, \ldots, x_k)$ est fait, le système spécialisé correspondant n'a pas un comportement générique. Dès que `F4Remake` détecte ce comportement inattendu (cf. section 3.2.2), on a vu que le calcul peut être poursuivi avec par exemple l'algorithme standard F4.

À titre d'illustration, on considère la cryptanalyse du système UOV (Unbalanced Oil and Vinegar [KPG99]) décrit dans [BFP10]. De façon succincte, l'attaque proposée se ramène à la résolution d'un système quadratique en n variables et n équations défini sur un corps $\mathbb{K}$; pour le jeu de paramètres recommandés, $n = 16$ et $\mathbb{K} = \mathbb{F}_{16}$. Bien que la taille du corps de base soit petite, la variante F4 donne pour cette cryptanalyse des résultats très satisfaisants. En effet, la partie quadratique des polynômes évalués $f_i(x_1, \ldots, x_k) \in \mathbb{K}[X_{k+1}, \ldots, X_n]$ ne dépend pas des valeurs des variables spécialisées $X_1, \ldots, X_k$; par conséquent, tous les systèmes considérés se comportent génériquement jusqu'à la première chute de degré.

Par exemple pour $k = 3$, le calcul avec F4 se fait en 6 étapes, et il n'y a pas de chute de degré avant l'avant-dernière étape. Ainsi, la probabilité de succès de `F4Remake` est heurisitiquement proche de $c(16)^2 \simeq 0.87$. Pour vérifier cette estimée, on a lancé `F4Remake` en faisant une exploration exhaustive de tout l'espace $\mathbb{F}_{16}^3$. La probabilité de succès obtenue (qui dépend bien sûr du système spécialisé considéré) est toujours de l'ordre de 90%, ce qui montre que l'estimée donnée est satisfaisante.

Les temps de calcul obtenus durant cette expérience confirment que la variante de F4 proposée offre un gain de performance non négligeable. En particulier, on note une amélioration (après un précalcul de 32.3 s) par rapport aux 9.41 s de F5 mentionnés dans [BFP10] ; bien entendu, cette comparaison est juste indicative, dans la mesure où les machines utilisées et les implantations de l'arithmétique et de l'algèbre linéaire ne sont pas les mêmes.

	F4Remake	F4	F4 Magma	F4/F4Remake
Temps (sec)	5.04	16.77	120.6	3.3
Matrice la plus large	5913×7005	10022×8329	10245×8552	2.0

TABLE 3.2 – Résultats expérimentaux sur UOV avec 3 variables spécialisées

3.3.3 Problème MinRank

On rappelle brièvement le problème MinRank : étant données $m+1$ matrices $M_0, \ldots, M_m \in \mathcal{M}_n(\mathbb{K})$ et un entier $r \in \mathbb{N}$, trouver un m-uplet $(\alpha_1, \ldots, \alpha_m) \in \mathbb{K}^m$ tel que

$$\text{Rang}\left(\sum_{i=1}^{m} \alpha_i M_i - M_0\right) \leq r.$$

On s'intéresse ici essentiellement au challenge A proposé dans [Cou01] pour lequel $\mathbb{K} = \mathbb{F}_{65521}$, $m = 10$, $n = 6$ et $r = 3$. Avec l'attaque de Kipnis-Shamir proposée dans [KS99], il est possible de convertir des instances du problème MinRank en des système quadratiques à plusieurs variables. Par exemple, pour le jeu de paramètres proposé dans le challenge A, on doit résoudre un système composé de 18 équations quadratiques en 20 variables. Une solution de ce système peut être trouvée en évaluant d'abord deux variables pour se ramener à un système "plein" (i.e. ayant le même nombre d'équations et de variables), ou avec l'approche hybride en spécialisant davantage de variables. L'utilisation de la variante de F4 se justifie naturellement dans le deuxième cas ; elle reste intéressante dans le premier cas, dans la mesure où plusieurs spécialisations sont souvent nécessaires pour obtenir des solutions rationnelles (elle est bien sûr aussi utile lorsque l'on souhaite trouver plusieurs instances du problème MinRank).

Que ce soit avec F4 ou avec la variante, avec les systèmes pleins ou les systèmes avec une variable spécialisée, les expériences montrent que les matrices obtenues aux différentes étapes de l'algorithme sont de tailles importantes (jusqu'à $39\,138 \times 22\,968$) et relativement creuses (moins de 5% d'entrées non nulles). Pour les deux types de systèmes, on obtient beaucoup de réductions à zéro avec F4 ; par exemple, on a observé que pour le système plein, à la huitième étape du calcul avec F4, parmi 17 739 paires critiques, 17 442 sont réduites à zéro. Ceci justifie clairement que l'algorithme F4 n'est pas adapté à la résolution de ce type de systèmes.

Il reste difficile de comparer les temps obtenus avec ceux donnés par [FLDVP08] utilisant F5 : comme on l'a déjà fait remarquer, les calculs ont été exécutés sur des machines différentes ; de plus il semble que l'algèbre linéaire utilisée dans l'implantation par Faugère de F5 dans FGb (dont le code source n'est pas public) soit hautement optimisée, bien plus que pour l'implantation de F4 faite dans Magma. Sur ce point, notre implantation n'est clairement pas compétitive : par exemple, à la septième étape de l'algorithme pour le système plein, le F4 de Magma réduit une matrice $26\,723 \times 20\,223$ en 28.95 s, alors qu'à la même étape notre implantation réduit une matrice légèrement plus petite de taille $25\,918 \times 19\,392$ en 81.52 s. Malgré ces limitations, on a obtenu des temps comparables à ceux de [FLDVP08], qu'on liste dans la table 3.3. Ces expériences montrent qu'avec une implantation plus élaborée de l'algèbre linéaire, la variante F4 serait probablement la plus efficace sur ces systèmes.

	F5	F4Remake	F4	F4 Magma
système plein	30.0	27.87	320.2	116.6
1 variable spécialisée	1.85	2.605	9.654	3.560

TABLE 3.3 – Résultats expérimentaux sur MinRank

Les calculs avec F5 ont été faits sur un bi-processor Xeon à 3.2 GHz. Les résultats avec `F4Remake` ont été obtenus après un précalcul de 4682 s sur $\mathbb{F}_{257}$ pour le système complet et 113 s pour le système avec une variable spécialisée.

3.3.4 Systèmes Katsura

Afin d'illustrer l'approche présentée en section 3.2.3, on applique notre algorithme de calcul de bases de Gröbner aux systèmes Katsura11 et Katsura12 [KFI+87], sur deux corps premiers de taille 16 et 32 bits. L'idée consiste alors à lancer le précalcul sur un corps de petite taille avant d'exécuter `F4Remake` sur un corps de taille plus grande. Les temps obtenus pour les deux systèmes montrent que le temps de précalcul est largement compensé par le gain de rapidité apporté par `F4Remake` sur le corps de taille 32 bits ; ce n'est par contre pas le cas pour le corps de taille 16 bits.

	8 bits	16 bits			32 bits		
	Précalcul	F4Remake	F4	F4 Magma	F4Remake	F4	F4 Magma
Katsura11	27.83	9.050	31.83	19.00	15.50	60.93	84.1
Katsura12	202.5	52.66	215.4	143.3	111.4	578.8	$> 5\,h$

TABLE 3.4 – Résultats expérimentaux sur Katsura11 et Katsura12

À titre de remarque, on observe que de façon surprenante, les matrices créées par F4 sont beaucoup plus petites dans notre version que dans celle de Magma (par exemple, à l'étape 12 de Katsura12, on obtient une matrice de taille 15393×19368 contre 20162×24137 pour l'implantation de Magma) ; bien entendu, on trouve avec les deux versions les mêmes nouveaux polynômes à chaque étape. Ce phénomène avait déjà été remarqué sur les systèmes précédents, mais jamais dans une telle proportion. Ceci semble indiquer que notre implantation de la fonction `Simplify` est bien plus efficace.

Étape	degré	tailles matrices dans F4Remake	tailles matrices dans F4	rapport de taille
9	10	14846×18928	18913×20124	1.4
10	11	15141×19235	17469×19923	1.2
11	12	8249×12344	16044×19556	3.1
12	13	2225×6320	15393×19368	21.2
13	14	–	15229×19313	–

(À la treizième étape, F4 ne trouve pas de nouveau générateur, donc cette étape est éliminée dans `F4Remake`)

TABLE 3.5 – Tailles des matrices apparaissant dans les dernières étapes de Katsura12

Deuxième partie

Problème du logarithme discret sur courbes algébriques définies sur des extensions

Chapitre 4

La problématique du logarithme discret

Le développement de la cryptographie à clef publique, qui a débuté assez tard dans l'histoire de la cryptologie, repose principalement sur le concept de *fonction à sens unique*. De façon informelle, une fonction de ce type correspond à une opération facilement calculable, mais très difficile à inverser connaissant son résultat. Les deux exemples fondamentaux sur lesquels s'appuie l'essentiel de la cryptographie moderne sont basés sur le problème de la factorisation d'entiers et le calcul de logarithmes discrets. Le premier de ces problèmes – n'importe quel calculateur est capable de trouver deux grands nombres premiers et de calculer leur produit, tandis que factoriser un grand nombre semi-premier est beaucoup plus difficile – a été fortement popularisé par le cryptosystème RSA [RSA78], qui reste le plus important schéma de chiffrement à clef publique. Ainsi, le problème de la factorisation a été largement étudié durant les dernières décennies et les algorithmes de résolution du problème ont enregistré des progrès significatifs, notamment avec l'introduction du crible algébrique [LL93]. En particulier, les tailles de clefs recommandées pour ces cryptosystèmes sont importantes : comme on sait factoriser des entiers de taille jusqu'à 768 bits [KAF+10], il est actuellement conseillé d'utiliser des clefs d'au moins 2048 bits.

Le problème du logarithme discret, dans sa formulation initiale sur les groupes multiplicatifs des corps finis, a connu le même sort que son homologue (les mêmes algorithmes de crible s'appliquant dans ce contexte). Néanmoins, et contrairement au problème de la factorisation, cette primitive présente l'énorme avantage de pouvoir être définie sur n'importe quel type de groupe fini. Des résultats théoriques [Sho97] laissent même à penser qu'il existe des groupes pour lesquels on ne peut pas trouver d'algorithmes performants pour attaquer le logarithme discret. Jusqu'à présent, les groupes associés aux courbes elliptiques et hyperelliptiques, proposés pour la première fois par Koblitz et Miller [Kob87, Mil86a], sont ceux sur lesquels le logarithme discret est le plus résistant et seront les principaux objets d'étude de la deuxième partie de cette thèse. À titre de comparaison, on donne en table 4.1 les différentes tailles de clefs à niveau de sécurité équivalent pour les cryptosystèmes basés sur la factorisation et le logarithme discret, ainsi que pour le standard AES de chiffrement symétrique, tels que présentés dans le rapport [Sma10].

Dans ce chapitre introductif, on commence par présenter les principaux protocoles cryptographiques basés sur la difficulté du problème du logarithme discret. On rappelle ensuite les attaques génériques, applicables sur n'importe quel groupe, et qui serviront de point de comparaison pour toutes les autres attaques proposées par la suite. La section suivante détaille le cadre général du calcul d'indices ; cette méthode est à la base des algorithmes de cribles algébriques qui sont ac-

Symétrique (AES)	Factorisation (RSA)	Log. discret sur courbes elliptiques
80	1 248	160
96	1 776	192
112	2 432	224
128	3 248	256
256	15 424	512

TABLE 4.1 – Comparaison des tailles de clefs par niveau de sécurité.

tuellement les plus performants sur les corps finis. On verra dans les chapitres suivants qu'elle s'applique aussi aux jacobiennes de courbes de grand genre [ADH94], de petit degré [Die06], ou définies sur des extensions de corps. Enfin, on présente quelques problèmes dits *non-standards* qui sont des variations sur le problème du logarithme discret, intervenant naturellement dans certains protocoles ou types d'attaques.

4.1 Importance du logarithme discret en cryptographie

4.1.1 Un problème difficile ?

De façon très générale, lorsque l'on dispose d'un groupe, on peut définir une fonction logarithme discret dans une base donnée par un élément de ce groupe :

Définition 4.1.1 (Logarithme discret)**.** *Soit G un groupe contenant un élément g d'ordre n. On appelle* logarithme discret en base g *d'un élément h appartenant au groupe engendré par g, l'entier $x \in \mathbb{Z}/n\mathbb{Z}$ tel que $h = g^x$.*

C'est donc la fonction réciproque de la fonction exponentielle $\exp_g : \mathbb{Z}/n\mathbb{Z} \to G, x \mapsto g^x$ qui se calcule aisément (en $O(\log x)$ opérations dans G) si l'on dispose d'une loi de groupe facilement calculable. En revanche, le problème du calcul du logarithme discret (DLP) défini par

Étant donnés un groupe G et $g, h \in G$, calculer – s'il existe – le logarithme de h en base g.

est de difficulté très variable selon le groupe G considéré. Par exemple, si l'on prend pour G un sous-groupe de $(\mathbb{Z}/n\mathbb{Z}, +)$, on peut résoudre très facilement le DLP avec l'algorithme d'Euclide étendu dont la complexité est polynomiale en la taille de n. Par contre, si l'on considère pour G un sous-groupe du groupe multiplicatif d'un corps fini $\mathbb{F}_q$, les meilleurs attaques connues du DLP s'appuient sur des méthodes de cribles [Kra26, Adl94, LL93], aussi appelées algorithmes de calcul d'indices, permettant d'obtenir des complexités sous-exponentielles en $L_q(1/3)$ lorsque q tend vers l'infini (voir [JL07] pour un état de l'art sur le sujet). On rappelle que la fonction L est une fonction d'estimation de complexité, définie pour tous réels positifs N, c et $\alpha \in [0, 1]$ par

$$L_N(\alpha, c) = \exp\left(c(\log N)^\alpha (\log\log N)^{1-\alpha}\right), \tag{4.1}$$

et on désigne par $L_N(\alpha)$ toute fonction de la forme $L_N(\alpha, c + o(1))$ pour une certaine constante c. Ainsi, une complexité sous cette forme est polynomiale en la taille de N lorsque $\alpha = 0$, et exponentielle lorsque $\alpha = 1$.

D'autres exemples très intéressants de groupes sont donnés par l'ensemble des points rationnels d'une courbe elliptique ou encore ceux d'une jacobienne de courbe de genre 2 définie sur $\mathbb{F}_p$ (p

premier) : dans ce cas, on ne connaît en général que des attaques génériques du DLP de complexité exponentielle, voir la section suivante et le chapitre 5 pour plus de détails.

4.1.2 Échange de clef de Diffie-Hellman

Pour s'échanger un secret via un canal de communication non sûr, i.e. sur lequel l'information peut être interceptée, Alice et Bob peuvent utiliser le protocole d'échange de clef de Diffie-Hellman [DH76] : ils s'entendent d'abord publiquement sur un groupe G cyclique engendré par un élément g pour lequel le DLP est supposé difficile, puis chacun de leur côté, choisissent un entier aléatoire, noté a pour Alice et b pour Bob, qu'ils gardent secret. Pour construire la clef commune K_{ab}, ils s'échangent alors via le canal de communication les éléments g^a et g^b, puis calculent respectivement $K_{ab} = (g^b)^a$ pour Alice et $K_{ab} = (g^a)^b$ pour Bob. L'espion Charlie qui observe les échanges

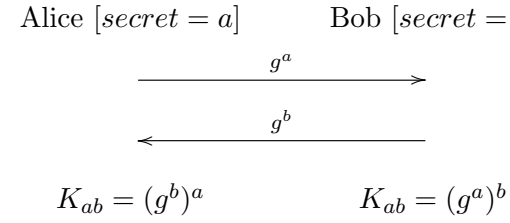


FIGURE 4.1 – Échange de clef de Diffie-Hellman

ne peut généralement pas en déduire la clef commune : son objectif serait en effet de résoudre le problème Diffie-Hellman calculatoire (appelé CDHP) suivant

Étant donnés un groupe G et trois éléments g, g^a et g^b de G, calculer g^{ab}.

Il est clair que si l'on sait résoudre le DLP, alors on peut résoudre le CDHP ; cette réduction est notée CDHP $\propto$ DLP. L'équivalence entre ces deux problèmes est moins claire, mais il y a de fortes présomptions en ce sens (voir par exemple [MW00]). Un problème similaire est appelé Diffie-Hellman décisionnel (DDHP) :

Étant donnés g, g^a, g^b et h, déterminer si $h = g^{ab}$.

L'introduction des couplages, i.e. d'applications bilinéaires, a montré qu'il existe des groupes – appelés groupes *Gap Diffie-Hellman* – sur lesquels ce problème est facile sans que le CDHP le soit [JN03] ; cette propriété est à la base de nombreux protocoles cryptographiques.

4.1.3 Chiffrement et signature ElGamal

Le premier schéma de chiffrement à clef publique basé sur le problème DLP est proposé par ElGamal dans [ElG85].

Pour que Bob puisse envoyer un message chiffré à Alice, il faut (comme dans tout schéma de chiffrement à clef publique) qu'Alice génère un couple clef privée/clef publique. Le protocole se déroule ainsi en trois étapes :

1. *Génération de clefs* : les deux protagonistes s'entendent préalablement sur un groupe cyclique G engendré par g dans lequel le DLP est difficile. Alice choisit alors sa clef privée en tirant aléatoirement un entier a et en déduit sa clef publique $K_a = g^a$.
2. *Chiffrement du message* : pour chiffrer son message m (que l'on suppose dans G pour simplifier), Bob génère une clef temporaire $K_t = g^t$ où t est un entier qu'il tire aléatoirement et calcule $c = mK_a^t$; le chiffré de m est alors la paire (K_t, c).
3. *Déchiffrement du message* : pour déchiffrer la paire (K_t, c), Alice calcule $K_t^{-a}c$ et retrouve le message clair m.

Encore une fois, pour que l'espion Charlie soit en mesure de retrouver le texte clair en observant les échanges entre Alice et Bob, il doit être capable étant donnés g, g^a et g^t, de calculer g^{at}, autrement dit de résoudre le CDHP.

Sur le même principe, un schéma de signature est également proposé dans l'article [ElG85]. On suppose que c'est Alice qui signe un message ; le protocole se décrit alors en trois étapes :

1. *Génération de clefs* : Alice et Bob s'entendent préalablement sur un groupe cyclique G engendré par g dans lequel le DLP est difficile et pour lequel on dispose d'une fonction de réduction (bijective) $\phi : G \to \mathbb{Z}/\#G\mathbb{Z}$ et d'une fonction de hachage $H : \{0;1\}^* \to \mathbb{Z}/\#G\mathbb{Z}$. Alice choisit alors sa clef privée en tirant aléatoirement un entier a et en déduit sa clef publique $K_a = g^a$.
2. *Signature du message* : pour signer $m \in \{0;1\}^*$, Alice génère un couple $(h, s) \in G \times \mathbb{Z}$ tel que $g^{H(m)} = K_a^{\phi(h)}h^s$. Pour cela, elle commence par choisir un entier aléatoire t premier avec $\#G$ et calcule $h = g^t$; elle détermine ensuite s en résolvant $a\phi(h) + ts = H(m) \bmod \#G$.
3. *Vérification de la signature* : pour vérifier que la signature (h, s) est valide, Bob doit vérifier si $g^{H(m)} = K_a^{\phi(h)}h^s$.

De nombreuses variantes de ce schéma existent, notamment les standards DSA dans le contexte des corps finis (FIPS 183) et ECDSA dans le contexte des courbes elliptiques (IEEE P1363/D3) [BSS05, chapitre I], ou encore la signature de Schnorr [Sch91].

4.1.4 Autres protocoles basés sur le logarithme discret

Si Alice dispose d'une clef publique $(g, K_a) = (g, g^a)$ où $g \in G$ et a est un entier secret, ainsi que d'une fonction de hachage (publique) $H : \{0;1\}^* \to G$, la façon la plus simple de signer un message est de calculer $\sigma = H(m)^a$. Pour vérifier si la signature est valide, il faut résoudre le DDHP pour $g, K_a, H(m)$ et σ. En revanche, pour forger une signature c'est le CHDP qu'il faut résoudre. Si le groupe G est "Gap Diffie-Hellman", on obtient ainsi le schéma de signature courte de Boneh-Lynn-Shacham [BLS01].

Si le DDHP est difficile sur G, on obtient une signature "undeniable" (non répudiable) au sens de Chaum-van Antwerpen [CvA90]. L'idée est que seule Alice peut autoriser Bob à vérifier sa signature, de la façon suivante : Bob choisit deux entiers aléatoires b et c, et envoie à Alice $K_a^b\sigma^c$; celle-ci calcule alors $\tau = (K_a^b\sigma^c)^{1/a}$ et le renvoie à Bob qui vérifie que $\tau = g^bH(m)^c$. Si $\tau \neq g^bH(m)^c$, il y a deux possibilités :
- soit la signature n'est pas celle d'Alice,
- soit Alice cherche intentionnellement à induire Bob en erreur.

Pour empêcher cette deuxième situation (non répudiabilité), Bob envoie un deuxième challenge $K_a^{b'}\sigma^{c'}$ à Alice qui doit répondre $\tau' = (K_a^{b'}\sigma^{c'})^{1/a}$; pour tester la sincérité d'Alice, Bob vérifie que $(\tau g^{-b})^{c'} = (\tau' g^{-b'})^c$.

Un schéma d'authentification reposant sur le même principe a été proposé dans [Fre05], mais l'idée en était probablement antérieure : pour que Bob puisse authentifier Alice, il lui envoie un défi $D = g^b$ pour un entier b choisi aléatoirement ; celle-ci répond $R = D^a$, et Bob vérifie si $R = K_a^b$. La sécurité de ce protocole repose une fois encore sur le CDHP.

4.2 Attaques génériques

Un certain nombre d'attaques connues du logarithme discret s'appliquent quelque soit le groupe fini G considéré : de telles attaques sont appelées *génériques*. De façon plus formelle, un algorithme est générique si les seules opérations qu'il utilise sont le calcul de produits, le calcul d'inverses et le test d'égalité dans G (on suppose néanmoins connus le cardinal de G et l'ordre de l'élément $g \in G$ choisi comme base pour le logarithme) ; autrement dit, le groupe G est vu comme une "boîte noire" ou comme un oracle fournissant le résultat de ces opérations. On note qu'en pratique, aucun des groupes que l'on considère n'est vraiment qu'une "boîte noire" : sa représentation donne toujours des informations supplémentaires.

Sans perte de généralité, on suppose désormais que G est le groupe cyclique engendré par g ; on cherche à calculer le logarithme d'un élément $h \in G$. L'attaque la plus simple est la recherche par force brute : on teste tous les entiers $x \in [\![0; \#G - 1]\!]$ jusqu'à ce que $g^x = h$. Bien entendu, la complexité est en $O(\#G)$ opérations, ce qui n'a d'intérêt que pour les groupes G de très petites tailles. On présente ici des attaques un peu plus élaborées basées principalement sur l'utilisation du théorème des restes chinois et sur le paradoxe des anniversaires.

4.2.1 Réduction de Pohlig-Hellman

Soit $n = \prod_{i=1}^{N} p_i^{\alpha_i}$ la décomposition en facteurs premiers de $\#G$. Si l'on connaît le logarithme de h modulo chacun des $p_i^{\alpha_i}$, on peut retrouver son logarithme en base g avec le théorème des restes chinois. Autrement dit, on va exploiter l'isomorphisme $G \simeq \prod \mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}$ pour réduire le DLP dans G à plusieurs DLP dans les sous-groupes de G d'ordre $p_i^{\alpha_i}$; une seconde réduction "à la Hensel" permet alors de ne travailler que dans des sous-groupes d'ordre p_i [PH78].

On commence par réduire le calcul du logarithme x de h en base g défini modulo n à celui du logarithme $x_i = x \bmod p_i^{\alpha_i}$ de $h_i = h^{n/p_i^{\alpha_i}}$ en base $g_i = g^{n/p_i^{\alpha_i}}$ pour $i = 1, \ldots, N$. En effet, le sous-groupe G_i engendré par g_i est de cardinal $p_i^{\alpha_i}$, et on a $h_i = g_i^x = g_i^{x_i}$. On se ramène ainsi au problème du logarithme discret dans un groupe de la forme $\mathbb{Z}/p^{\alpha}\mathbb{Z}$.

Dans une deuxième phase, on cherche x_i sous la forme $x_i = z_{i1} + z_{i2}p_i + \cdots + z_{i\alpha_i}p_i^{\alpha_i - 1}$ avec $z_{ij} \in [\![0; p_i - 1]\!]$ (pour simplifier les écritures, on omet dans la suite l'indice i). Le calcul peut se faire de proche en proche en travaillant dans les sous-groupes de cardinal p^k engendrés par $y_k = g^{p^{\alpha-k}}$. En effet, on a $h^{p^{\alpha-k}} = y_k^{z_1 + z_2 p + \cdots + z_k p^{k-1}}$ et si l'on a déjà calculé $z_1, \ldots, z_{k-1}$, alors

$$h^{p^{\alpha-k}} (y_k^{-1})^{\sum_{j=1}^{k-1} z_j p^{j-1}} = y_1^{z_k} \ ;$$

on obtient ainsi z_k par un calcul de logarithme discret dans le sous-groupe engendré par $y_1 = g^{p^{\alpha-1}}$ d'ordre p.

Si l'on note $c(p_i)$ le coût du calcul par une méthode générique d'un logarithme discret dans un groupe de cardinal p_i, on voit qu'avec la réduction de Pohlig-Hellman, le calcul d'un logarithme

discret dans G d'ordre $n = \prod_{i=1}^{N} p_i^{\alpha_i}$ coûte $\sum_{i=1}^{N} \alpha_i\, c(p_i)$. Cette réduction est particulièrement efficace lorsque chacun des facteurs p_i est petit, et fait qu'en pratique, on estime que le DLP dans un groupe G est essentiellement aussi difficile que le DLP dans son plus grand sous-groupe d'ordre premier. Dans la suite, on considère donc presque exclusivement des logarithmes en base g où $g \in G$ est d'ordre un grand nombre premier.

4.2.2 "Pas-de-bébé pas-de-géant"

L'idée de la méthode "pas-de-bébé pas-de-géant" [Sha71] est d'utiliser un compromis temps-mémoire pour accélérer considérablement la recherche par force brute. Soit $d < \#G$ un entier ; le logarithme $x \in [\![0; \#G - 1]\!]$ de h en base g s'écrit de façon unique $x = ad + r$ avec $0 \leq r < d$ et $0 \leq a \leq \lfloor \#G/d \rfloor$. En particulier, a est l'unique entier entre 0 et $\lfloor \#G/d \rfloor$ tel que $h \cdot (g^{-d})^a$ appartienne à l'ensemble $\{g^i : 0 \leq i < d\}$.

Le principe de l'algorithme "pas-de-bébé pas-de-géant" est le suivant. Dans un premier temps, on construit la liste L_d des couples (i, g^i) pour $0 \leq i < d$. Puis on calcule de proche en proche $h \cdot (g^{-d})^k$ en partant de $k = 0$ jusqu'à obtenir un élément de $\{g^i : 0 \leq i < d\}$, qui correspond donc à un unique couple (s, g^s) de L_d. On a alors $h \cdot (g^{-d})^k = g^s$, et on obtient $x = kd + s$.

La difficulté est de tester rapidement l'appartenance à la liste L_d. Si cette liste est préalablement triée suivant la représentation des valeurs de g^i (ce qui se fait en $O(d \log d)$), le test d'appartenance se fait par dichotomie en temps $O(\log d)$. La complexité totale de l'algorithme est alors en $O(d \log d + \lfloor \#G/d \rfloor \log d)$. Une façon plus efficace de procéder est d'utiliser une table de hachage (suivant les valeurs de g^i) pour ranger les éléments de L_d. Le test d'appartenance est alors en temps constant et la complexité totale $O(d + \lfloor \#G/d \rfloor)$. Dans tous les cas, la complexité en mémoire est en $O(d)$. Le compromis donnant la meilleur complexité temporelle est de prendre d de l'ordre de $\sqrt{\#G}$, donnant une complexité finale (en temps et en mémoire) de $O(\sqrt{\#G})$.

4.2.3 La méthode rho de Pollard

Pour gagner sur la complexité en mémoire, Pollard introduit en 1978 [Pol78] un algorithme probabiliste de calcul du logarithme discret dont la complexité temporelle reste en $O(\sqrt{\#G})$. L'idée est d'itérer une fonction $F : G \to G$ vérifiant les propriétés suivantes :

1. F doit être simple à calculer,
2. étant donnés $\alpha, \beta \in \mathbb{Z}/\#G\mathbb{Z}$, on doit pouvoir trouver facilement $\alpha', \beta' \in \mathbb{Z}/\#G\mathbb{Z}$ tels que $F(g^\alpha h^\beta) = g^{\alpha'} h^{\beta'}$,
3. le comportement de F doit être "suffisamment proche" de celui d'une fonction aléatoire.

Les fonctions simples vérifiant les points 1. et 2. sont du type $x \mapsto x^k$ avec k un entier petit, $x \mapsto g \cdot x$, $x \mapsto h \cdot x$, ou des composées de ces trois primitives. L'approche originelle de Pollard consiste à alterner entre plusieurs fonctions de ce type : on partitionne G en trois sous-ensembles G_1, G_2 et G_3 de tailles comparables, et on pose

$$F(x) = \begin{cases} x^2 & \text{si } x \in G_1, \\ g \cdot x & \text{si } x \in G_2, \\ h \cdot x & \text{si } x \in G_3. \end{cases}$$

En pratique, les performances peuvent être améliorées en alternant de cette façon entre au moins 20 fonctions élémentaires [Tes01].

En partant d'un élément $u_0 = g^{\alpha_0} h^{\beta_0}$, on calcule la suite (u_i) de ces itérés ainsi que les suites $(\alpha_i), (\beta_i)$ de telle sorte que $u_i = F(u_{i-1}) = F^i(u_0) = g^{\alpha_i} h^{\beta_i}$. Comme le groupe est fini, la suite des itérés par F est ultimement périodique : il existe deux entiers i_0 et $\ell > 0$ tels que $u_{i_0} = u_{i_0+\ell}$ (et donc $u_i = u_{i+\ell}$ pour tout $i \geq i_0$). Si l'on dessine la suite des itérés de u_0 par F, on obtient une "queue" suivie d'un cycle, d'où le nom de rho donné à cet algorithme, voir Figure 4.2.

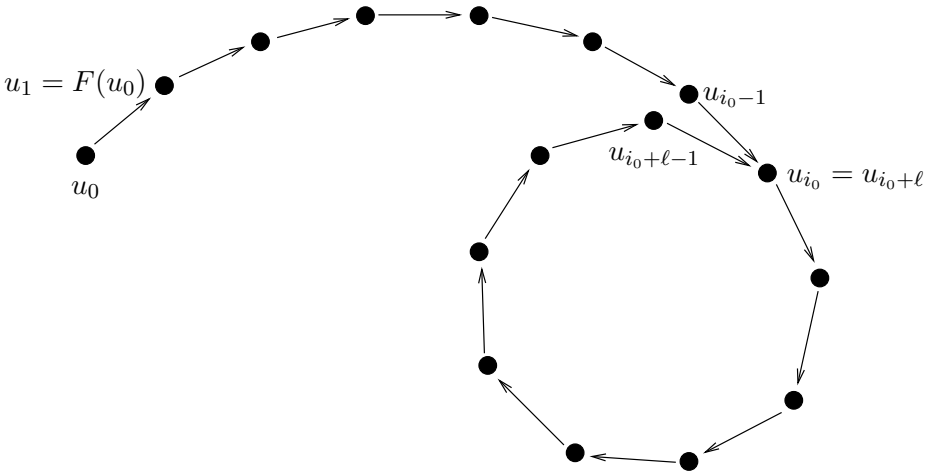


FIGURE 4.2 – La suite des itérés de u_0 par F dans Pollard-rho.

La collision entre u_{i_0} et $u_{j_0} = u_{i_0+\ell}$ donne $g^{\alpha_{i_0}} h^{\beta_{i_0}} = g^{\alpha_{j_0}} h^{\beta_{j_0}}$; si $\beta_{i_0} - \beta_{j_0}$ est premier avec $\#G$, on en déduit que le logarithme de h en base g est $-(\alpha_{i_0} - \alpha_{j_0})(\beta_{i_0} - \beta_{j_0})^{-1} \bmod \#G$. Si $\beta_{i_0} = \beta_{j_0} \bmod \#G$, alors il faut recommencer [1] en partant d'un autre élément u_0.

Il n'est pas immédiat que la complexité de cette méthode soit comparable avec celle de "pas-de-bébé pas-de-géant". Cependant, on peut montrer que si l'on itère une fonction aléatoire de G dans G, le temps moyen avant de trouver une collision (i.e. de parcourir un cycle) est en $O(\sqrt{\#G})$; en effet, si F est une fonction uniformément aléatoire, les éléments $u_0, u_1 = F(u_0), \ldots$ forment une suite aléatoire uniformément distribuée dans G jusqu'à la première collision, et une analyse type paradoxe des anniversaires montre alors que cela arrive au bout de $O(\sqrt{\#G})$ itérations. En pratique F n'est pas aléatoire, mais son comportement s'en approche suffisamment pour que cette analyse reste valide.

La difficulté restante est la détection des cycles : si l'on doit stocker toutes les valeurs des u_i jusqu'à obtenir une collision, la complexité en mémoire est encore en $O(\sqrt{\#G})$. Il existe plusieurs méthodes pour détecter les cycles dont le coût mémoire est en $O(1)$, on présente la plus simple et la plus ancienne due à Floyd [Flo67, Knu69]. On considère en plus de la suite (u_i) – la "tortue", la suite (v_i) telle que $v_i = u_{2i}$ – le "lièvre" – et on itère jusqu'à trouver une collision $u_i = v_i$. Une telle collision se produit dès que i est un multiple de ℓ (la longueur du cycle) supérieur à i_0 (l'entrée du cycle), donc pour une valeur de i nécessairement plus petite que $i_0 + \ell$.

Parallélisation

Il est possible de transformer l'algorithme de Pollard en une version parallélisable [OW99], de telle sorte que la complexité soit en $O(\sqrt{\#G}/m)$ où m est le nombre d'unités de calcul dont on

1. En pratique, on utilise toujours Pohlig-Hellman pour se ramener au cas où $\#G$ est premier, sauf si l'on ne connaît pas la factorisation de $\#G$. Dans ce cas, si $(\beta_{i_0} - \beta_{j_0}) \wedge \#G$ est un facteur non trivial de $\#G$, on utilise cette information pour simplifier avec Pohlig-Hellman avant de relancer Pollard-rho.

dispose. L'idée est de rechercher non plus des cycles mais des collisions entre des chemins. Plus précisément, on fixe à l'avance un certain nombre D de *points distingués* faciles à reconnaître (par exemple, les derniers bits de leur représentation mémoire sont égaux à une valeur donnée). Une machine joue le rôle de serveur central, et les autres machines (les clients) démarrent une marche pseudo-aléatoire en partant d'éléments u_0 distincts et en itérant la fonction F précédemment définie. Dès qu'une suite d'itérés rencontre un point distingué P, le client en informe le serveur central en lui donnant les valeurs correspondantes de α_i et β_i telles que $P = g^{\alpha_i} h^{\beta_i}$, puis redémarre à partir d'un nouveau point u_0. Dès que le serveur central détecte une collision, i.e. lorsqu'un même point distingué a été obtenu deux fois, on en déduit le logarithme discret à condition que la différence entre les deux valeurs de β soit première à $\#G$. À cause de la forme des collisions entre les marches aléatoires (voir Figure 4.3), cet algorithme est parfois appelé le "lambda" de Pollard [2].

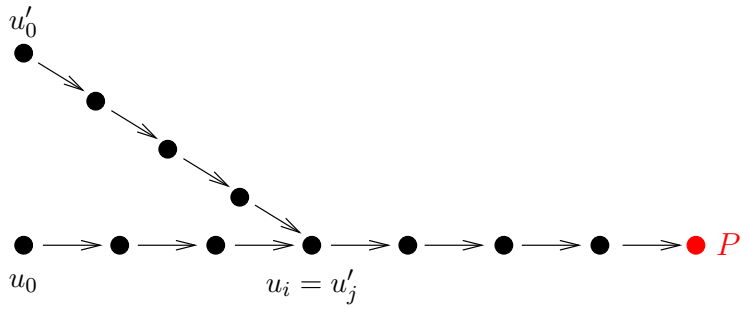


FIGURE 4.3 – Une collision entre deux chemins dans la version parallélisée de Pollard-rho.

La taille de l'ensemble des points distingués est un paramètre important : si D est trop petit, il faut beaucoup d'itérations avant de tomber sur un point distingué ce qui conduit à des calculs superflus (qui correspondent aux itérations entre u'_j et P dans la figure ci-dessus) ; mais si D est trop grand, le serveur central va devoir stocker beaucoup de points avant la collision et risque de saturer sous les rapports des clients. Suivant [SG00], la valeur de D optimale est de l'ordre de $\sqrt{\#G}$, et on obtient alors une complexité totale en $\sqrt{\#G}/m$.

4.2.4 Attaques génériques et complexité

Avec les méthodes présentées ci-dessus, on a vu que la complexité de la résolution du DLP dans un groupe cyclique G de taille n est en $O(\sqrt{p})$ où p est le plus grand facteur premier de n. De façon surprenante, cette complexité est en fait optimale : d'après un théorème de Shoup [Sho97], la probabilité de succès d'un algorithme générique de résolution du DLP utilisant m opérations dans G est en $O(m^2/p)$. Pour avoir une probabilité non négligeable de succès, il est donc nécessaire d'avoir m de l'ordre de $\sqrt{p}$. En ce sens, les algorithmes présentés dans cette section sont donc optimaux.

Au vu de ce résultat, un des buts en cryptographie est donc de trouver des groupes pour lesquels il n'existe pas – ou du moins on ne connaît pas – d'attaques meilleures que les attaques génériques. À ce jour, les groupes les plus intéressants et les plus étudiés ayant ces qualités sont les courbes elliptiques définies sur des corps premiers ou de cardinalité 2^p avec p premier, ainsi que les jacobiennes de courbes de genre 2 définies sur ces mêmes corps. Par opposition, le but du cryptanalyste est de trouver une façon d'exploiter l'information additionnelle fournie par la description du groupe pour attaquer plus rapidement le DLP. Par exemple, dans le cas du groupe

2. Ce qui est source de confusion, un autre algorithme de Pollard (aussi appelé "kangourou") portant le même nom.

multiplicatif $\mathbb{F}_p^*$, la présence d'une loi supplémentaire d'addition permet d'obtenir des algorithmes de complexité sous-exponentielle basés sur le calcul d'indices, que l'on va présenter dans la section suivante.

4.3 Calcul d'indices

Les méthodes de calcul d'indices ont été initialement développées pour permettre la factorisation d'entiers. Le principe en est assez ancien, et ses premières traces remontent à l'idée de Fermat consistant à rechercher des congruences de carrés modulo n pour factoriser l'entier n ; il faut cependant attendre la première moitié du vingtième siècle pour une formulation moderne [Kra26]. De nombreuses avancées ont eu lieu à partir des années 80, avec le développement du crible quadratique de Pomerance [Pom82] et des cribles algébriques basés sur les corps de nombres (NFS, [LL93]) et les corps de fonctions (FFS, [Adl94]) qui détiennent les records pour la factorisation et le logarithme discret sur corps finis. On ne présente ici que le schéma de base sur lequel s'appuient toutes ces méthodes.

4.3.1 Description générale

En vue des applications présentées dans les chapitres suivants, on considère dans cette section le problème du logarithme discret sur un groupe commutatif fini G noté additivement : soient $h, g \in G$, trouver – si possible – le secret x tel que $h = [x]g$. On suppose sans perte de généralité que le sous-groupe engendré par $g \in G$ est d'ordre premier r.

Pour faire un calcul d'indices, on définit d'abord une *base de factorisation* $\mathcal{F} = \{g_1, \ldots, g_N\}$ constituée de certains éléments du groupe G, qui engendrent tout le groupe. On cherche ensuite à obtenir suffisamment de relations faisant intervenir les éléments de cette base ainsi que des multiples de g et h, de façon à pouvoir en déduire par des techniques d'algèbre linéaire (type Wiedemann ou Lanczos, voir ci-dessous) le logarithme discret de h en base g. Pour simplifier, on supposera que $\#G = mr$ où $m \wedge r = 1$; en particulier, $[m]G = \{[m]f : f \in G\}$ est un sous-groupe isomorphe à $\mathbb{Z}/r\mathbb{Z}$.

On présente ici deux variantes : la variante 1, introduite en premier historiquement, ne permet de calculer qu'un seul logarithme discret, alors que la variante 2 est à utiliser lorsque l'on souhaite calculer l'ensemble des logarithmes discrets de la base de factorisation. En particulier dans cette deuxième variante, il est possible avec une phase de descente de retrouver le logarithme de n'importe quel élément $h \in \langle g \rangle$.

Variante 1

1. *Phase de recherche de relations :* pour un choix d'entiers a_i, b_i aléatoires dans $\mathbb{Z}/r\mathbb{Z}$, décomposer l'élément $[a_i]g + [b_i]h$ dans $\mathcal{F}$ afin d'obtenir des relations de la forme

$$[a_i]g + [b_i]h = \sum_{j=1}^{N} [c_{ij}]g_j, \text{ où } c_{ij} \in \mathbb{Z}. \tag{4.2}$$

2. *Phase d'algèbre linéaire :* une fois que k relations de la forme (4.2) sont obtenues, constuire les matrices $A = \begin{pmatrix} a_i & b_i \end{pmatrix}_{1 \leq i \leq k}$ et $M = \begin{pmatrix} c_{ij} \end{pmatrix}_{\substack{1 \leq i \leq k \\ 1 \leq j \leq N}}$, puis calculer un élément $v = \begin{pmatrix} v_1 & \ldots & v_k \end{pmatrix}$

dans le noyau à gauche de M tel que $vA \neq \begin{pmatrix} 0 & 0 \end{pmatrix} \bmod r$. Un tel vecteur v existe et peut être calculé facilement avec de l'algèbre linéaire élémentaire dès que $k \geq N$ et que les relations sont linéairement indépendantes. Le logarithme discret de h en base g est alors $x = -(\sum_i a_i v_i)/(\sum_i b_i v_i) \bmod r$.

Variante 2

1. *Phase de recherche de relations :* pour un choix d'entier a_i aléatoire dans $\mathbb{Z}/r\mathbb{Z}$, décomposer – lorsque cela est possible – l'élément $[a_i]g$ dans la base de factorisation, autrement dit trouver une relation de la forme
$$[a_i]g = \sum_{j=1}^{N} [c_{ij}]g_j, \text{ où } c_{ij} \in \mathbb{Z}. \tag{4.3}$$
2. *Phase d'algèbre linéaire :* une fois que l'on a obtenu k relations indépendantes de la forme (4.3), constuire le vecteur $A = (a_i)_{1 \leq i \leq k}$ ainsi que la matrice $M = (c_{ij})_{\substack{1 \leq i \leq k \\ 1 \leq j \leq N}}$, puis résoudre le système linéaire $MX = A \bmod r$. Dès lors que $k \geq N$ relations indépendantes, ce système admet une unique solution qui peut être calculée en utilisant de l'algèbre linéaire élémentaire. Le vecteur X solution du système contient alors tous les logarithmes en base $[m]g$ des éléments de la forme $[m]g_j$, $g_j \in \mathcal{F}$.
3. *Phase de descente :* trouver une équation de la forme
$$[a]g + [b]h = \sum_{j=1}^{N} [c_j]g_j, \text{ où } b \wedge r = 1. \tag{4.4}$$
En déduire le logarithme de h en calculant $\left(\sum_{j=1}^{N} c_j \alpha_j - a\right) b^{-1} \bmod r$, où α_j est le logarithme discret de $[m]g_j$ en base $[m]g$.

Dans certains cas, il est éventuellement possible de ne considérer que des relations de la forme
$$\sum_{j=1}^{N} [c_{ij}]g_j = 0, \text{ où } c_{ij} \in \mathbb{Z}. \tag{4.5}$$
(Ces deux formes (4.3) et (4.5) sont en fait similaires.) On cherche alors un vecteur non nul dans le noyau de la matrice $M = \left(c_{ij}\right)_{\substack{1 \leq i \leq k \\ 1 \leq j \leq N}} \bmod r$, qui donne les logarithmes des éléments de la forme $[m]g_j$, $g_j \in \mathcal{F}$, à une constante multiplicative près. On a ensuite besoin de deux relations du type (4.4) pour la phase de descente.

L'exemple fondamental est celui où $G = (\mathbb{Z}/p\mathbb{Z})^*$ avec p premier. On prend pour base de factorisation l'ensemble des classes d'équivalence des nombres premiers plus petits qu'une certaine borne B fixée. Un élément est alors décomposable dans cette base de factorisation si son représentant dans $\{1; \ldots; p-1\}$ est B-lisse. Pour rendre la méthode efficace, on doit trouver le meilleur compromis : en effet, si B est grand, alors la plupart des éléments sont décomposables ; en contrepartie, on a besoin de beaucoup de relations, et les matrices intervenant dans la phase d'algèbre linéaire sont d'autant plus grandes. D'un autre côté, si la base de factorisation est petite, alors on a besoin de peu de relations et la phase d'algèbre linéaire est rapide ; par contre, la probabilité d'obtenir une relation est beaucoup plus faible. Dans tous les cas, la matrice M est toujours très peu dense ; en particulier, on a intérêt à appliquer des techniques appropriées pour pouvoir calculer des éléments

de son noyau rapidement. On note que pour rendre cet exemple efficace, il est nécessaire d'utiliser des techniques de crible pour accélérer la phase de recherche de relations.

Pour l'analyse dans les sections suivantes, on aura souvent besoin de supposer que l'élément $[a_i]g$ (ou $[a_i]g + [b_i]h$) à décomposer est uniformément réparti dans G. Si le sous-groupe engendré par g n'est pas égal à G, cette propriété peut s'obtenir en considérant $g_1, \ldots, g_t \in \mathcal{F}$ tels que $\langle g, g_1, \ldots, g_t \rangle = G$; on essaie alors de décomposer des éléments de la forme

$$[a_i]g\,(+[b_i]h) + [\lambda_{i,1}]g_1 + \ldots + [\lambda_{i,t}]g_t,$$

où les $a_i, b_i, \lambda_{i,j}$ sont choisis aléatoirement dans $\mathbb{Z}/r\mathbb{Z}$.

Algèbre linéaire

En général, chaque décomposition ne comporte que très peu d'éléments de la base de factorisation. La matrice M qui intervient durant la phase d'algèbre linéaire est par conséquent très creuse, ce qui permet d'utiliser des techniques adaptées pour la résolution du système linéaire associé. L'idée est que la multiplication d'un vecteur par une matrice creuse est peu coûteuse et requiert seulement C multiplications, où C est le nombre de coefficients non nuls de M. Il existe principalement deux familles d'algorithmes itératifs de résolution de systèmes linéaires, basées sur les techniques de Lanczos et de Wiedemann (voir [CFA+06, §20.3.3] pour une introduction sur le sujet). Dans les deux cas, la complexité de la résolution est en $O(CN)$ opérations [3] dans $\mathbb{Z}/r\mathbb{Z}$, où N est la taille de la base de factorisation (i.e. la taille de M). En utilisant des variantes "par blocs", il est possible de distribuer en partie ces algorithmes mais de façon nettement moins efficace que durant la phase de recherche de relations où la parallélisation est immédiate.

La très grande taille de la matrice M est en général l'obstacle principal à l'implantation des algorithmes de calcul d'indices. Une solution partielle consiste à calculer beaucoup plus de relations que nécessaire, et de se servir de cette information redondante pour dans un premier temps diminuer la taille de M avant de procéder à la résolution du système associé : c'est le principe derrière les variations "large primes" (voir section suivante) et l'élimination gaussienne structurée, que l'on présente maintenant.

Quand G est le groupe multiplicatif d'un corps fini, on constate que la matrice M est non seulement creuse, mais aussi structurée dans la mesure où les petits éléments de $\mathcal{F}$ apparaissent beaucoup plus fréquemment que les grands dans les décompositions. Certaines colonnes de M sont donc plus denses que d'autres. De plus, comme déjà remarqué la plupart des coefficients sont égaux à ± 1. L'élimination gaussienne structurée [LO91] commence par regarder si certains facteurs n'apparaissent qu'une seule fois (i.e. une colonne ne contient qu'un seul coefficient non nul) ; on supprime dans ce cas la colonne et la ligne correspondante. On supprime de même toutes les colonnes ne contenant que des zéros. On sélectionne ensuite les colonnes les plus denses, et on cherche des lignes ne contenant qu'un seul coefficient non nul égal à ± 1 dans les colonnes restantes ; on se sert de ce coefficient comme pivot pour annuler tous les entrées non nulles de sa colonne, puis on supprime la ligne et la colonne du pivot. Si certaines lignes deviennent trop denses dans ce processus, on les supprime. On recommence alors à la première étape et on élargit l'ensemble des colonnes denses. On diminue ainsi la taille de la matrice, tout en préservant la faible densité.

Dans cette thèse, on considère principalement des groupes G associés à des courbes algébriques, et les matrices M intervenant dans le calcul d'indices sont en général peu structurées : les co-

3. Ces opérations sont en pratique majoritairement des additions/soustractions, puisque les coefficients de la matrice des relations sont presque toujours ± 1.

efficients sont plus uniformément répartis que dans le cas de $(\mathbb{Z}/p\mathbb{Z})^*$. Il est cependant possible d'utiliser dans ce contexte l'élimination gaussienne structurée, dont les performances restent correctes (cf. section 8.3). Pour l'étude des complexités, on préférera néanmoins utiliser les techniques présentées en section suivante, d'analyse plus aisée.

4.3.2 Variations "large primes"

Lorsque l'on fait du calcul d'indices sur $\mathbb{F}_p^*$, on rencontre fréquemment des éléments qui sont presque décomposables ou B-lisses, au sens où tous les facteurs premiers de leurs représentants sauf un ou deux sont dans la base de factorisation. Plutôt que de rejeter ces relations, on peut essayer de les combiner pour obtenir de nouvelles relations ne faisant intervenir plus que des éléments de $\mathcal{F}$: c'est l'idée des méthodes "one large prime" et "double large primes" [Pom82, LM94]. Ces méthodes ont ensuite été appliquées avec succès au calcul d'indices sur variété jacobienne, voir [GTTD07] et section 5.4.2.

Dans ces variations, on introduit en plus de la base de factorisation $\mathcal{F} = \{g_1, \ldots, g_n\}$ (les petits nombres premiers quand on travaille sur $\mathbb{F}_p^*$), une base secondaire $\mathcal{F}' \subset G$, dont on peut voir les éléments comme des "grands nombres premiers" ; dans les applications, la distinction entre $\mathcal{F}$ et $\mathcal{F}'$ sera le plus souvent complètement arbitraire. Le cardinal de la base secondaire $\mathcal{F}'$ est en général beaucoup plus gros que celui de $\mathcal{F}$.

Variation "one large prime"

Dans cette méthode, on stocke les décompositions obtenues qui sont de la forme

$$[a_i]g + [b_i]h = \sum_{j=1}^{N} [c_{i,j}]g_j \pm g', \quad g' \in \mathcal{F}', \tag{4.6}$$

idéalement en utilisant une table de hachage suivant la valeur de g'. Dès que l'on obtient deux relations faisant intervenir le même "grand facteur" g', on les combine pour obtenir une nouvelle relation où le facteur g' est éliminé.

Le paradoxe des anniversaires montre que de telles collisions vont arriver assez fréquemment. Plus précisément, si l'on suppose que chaque "grand facteur" apparaît avec la même probabilité, pour obtenir de l'ordre de $\#\mathcal{F}$ relations usuelles de la forme (4.3) ou (4.2), il faut disposer en moyenne de l'ordre de $\sqrt{\#\mathcal{F} \cdot \#\mathcal{F}'}$ relations "one large prime".

Variation "double large primes"

On conserve dans cette variante les décompositions obtenues qui sont de la forme

$$[a_i]g + [b_i]h = \sum_{j=1}^{N} [c_{i,j}]g_j \pm g'_1 \pm g'_2, \quad g'_1, g'_2 \in \mathcal{F}'. \tag{4.7}$$

On construit pour cela un graphe, dont les sommets sont les éléments de $\mathcal{F}'$ plus un sommet distingué, noté $*$; initialement le graphe ne contient aucune arête. À chaque relation de la forme (4.7) on fait correspondre une arête entre les sommets g'_1 et g'_2, et à chaque relation de la forme (4.6)

on fait correspondre une arête entre les sommets g' et $*$. Pendant la phase de recherche de relations, pour chaque nouvelle relation obtenue, on considère l'arête correspondante dans le graphe :

- si l'arête ne crée pas de cycle, alors on la rajoute au graphe avec comme étiquette la relation elle-même ;
- si l'arête crée un cycle, on essaie de faire une combinaison linéaire des relations correspondant aux arêtes du cycle de façon à éliminer les éléments de $\mathcal{F}'$; deux cas sont possibles :
  - soit on obtient une nouvelle relation sans élément de $\mathcal{F}'$; c'est le cas en particulier lorsque le cycle contient le sommet $*$;
  - soit on obtient une nouvelle relation contenant un seul élément de $\mathcal{F}'$, de la forme (4.6) ; on ajoute alors au graphe l'arête correspondante ayant $*$ comme extrémité.

L'analyse est plus compliquée que pour la variante "one large prime". Si l'on suppose encore que chaque grand facteur apparaît avec la même probabilité, on peut utiliser des résultats sur les graphes aléatoires, voir par exemple [Bol01]. Heuristiquement, tant que le nombre d'arêtes n'est pas de l'ordre de $\#\mathcal{F}'$, la probabilité d'obtenir un cycle est extrêmement faible. Par contre, à partir du moment où le nombre d'arêtes est de l'ordre de $\#\mathcal{F}'$, la majorité des sommets du graphe appartient à une même composante connexe de diamètre en $O(\log(\#\mathcal{F}'))$ et quasiment chaque nouvelle arête va créer un cycle, de longueur bornée par $O(\log(\#\mathcal{F}'))$. En particulier, pour obtenir de l'ordre de $\#\mathcal{F}$ relations usuelles de la forme (4.3) ou (4.2), il faut disposer en moyenne de l'ordre de $\#\mathcal{F} + \#\mathcal{F}'$ relations "double large primes".

4.4 Problèmes non standards

Comme on l'a déjà vu avec les exemples de l'échange de clef de Diffie-Hellmann ou du chiffrement d'ElGamal, il n'est pas rare que la sécurité d'un protocole se base sur un problème a priori moins difficile que le DLP, i.e. des problèmes que l'on peut résoudre lorsque l'on sait résoudre le DLP. On présente dans cette section quelques problèmes de ce type, appelés "non-standards" par Koblitz et Menezes [KM08].

4.4.1 Diffie-Hellman statique assisté d'un oracle

On rappelle la définition du *problème Diffie-Hellman statique assisté d'un oracle* (noté SDHP) :

Définition 4.4.1. *Soient G un groupe fini d'ordre $\#G$ et $g, h \in G$ deux éléments tels que $h = g^x$ où $x \in [\![1; \#G - 1]\!]$ est un entier secret. On dit qu'un algorithme $\mathcal{A}$ sait résoudre SDHP dans G, si étant donnés g, h et un challenge $y \in G$, il peut retourner $y^x \in G$.*
Un algorithme $\mathcal{A}$ qui sait résoudre SDHP est dit assisté d'un oracle *si :*

(i) durant une phase d'apprentissage, il peut demander n'importe quelles requêtes $y_1, \ldots, y_l$ à un oracle qui retourne $y_1^x, \ldots, y_l^x$;

(ii) après la phase d'apprentissage, pour un challenge y non préalablement vu, il est capable de retourner y^x.

Ce problème apparaît naturellement dans la sécurité de certains protocoles :

- Dans le schéma de chiffrement d'ElGamal, si l'on dispose d'un couple clair/chiffré, on connaît g^t et $(g^t)^a$ où a est le secret. Lors d'une attaque à chiffrés choisis, on dispose ainsi d'un moyen de calculer la puissance a-ième de n'importe quel élément. Être ensuite capable d'élever à la puissance a permet de déchiffrer tous les messages suivants. La sécurité d'ElGamal contre une attaque à chiffrés choisis repose donc sur la sécurité du SDHP assisté d'un oracle.

- Dans le schéma d'authentification de [Fre05], Bob dispose d'un couple (D, D^a) à chaque fois qu'Alice doit s'authentifier, et il ne faut pas qu'après un certain nombre d'authentifications, Bob puisse usurper l'identité d'Alice.
- De même, dans le schéma de signature non-répudiable de Chaum-van Antwerpen, Bob peut faire calculer à Alice $y^{1/a}$ pour un élément y de son choix à chaque demande de vérification de signature, et il ne faut pas qu'après un certain nombre de demandes de vérification, Bob puisse vérifier lui-même les signatures d'Alice.

D'un point de vue général, si l'on est capable de décomposer des éléments de G dans une base de factorisation $\mathcal{F} = \{g_1, \ldots, g_l\}$, on a alors l'algorithme assisté d'un oracle suivant (voir [KM08]) :

1. durant la phase d'apprentissage, on demande à l'oracle de calculer $h_i = g_i^x$ pour tout $i \in [\![1; l]\!]$,
2. on décompose ensuite le challenge y sous la forme $y = \prod_i g_i^{c_i}$ et on répond $\prod_i h_i^{c_i}$.

En particulier, on verra en section 7.3.3 comment il est possible de mener une attaque efficace contre SDHP assisté d'un oracle dans le cas des courbes elliptiques définies sur des extensions de corps finis.

4.4.2 Autres problèmes

De façon analogue au problème précédent, on peut définir le problème du logarithme discret assisté d'un oracle : après une phase d'apprentissage où l'on a accès à un oracle de résolution du DLP en base g, il faut être capable de calculer le logarithme discret (en base g) d'un challenge non rencontré précédemment. Il est clair que les techniques basées sur le calcul d'indices permettent de résoudre facilement ce problème ; en fait, la phase d'apprentissage remplace la phase de recherche de relations ainsi que l'algèbre linéaire, et seule la descente reste à effectuer. Cependant, on ne connaît pas d'exemple d'attaques sur un protocole dont la sécurité serait équivalente à ce problème (cf. [KM08, Open Problem 1]).

D'autres variantes plus intéressantes sur le même thème sont les problèmes type "one more" DHP (ou DLP), ainsi que DHP1 et DLP1 : dans ce contexte, on dispose d'un oracle de résolution du DHP (ou du DLP), ainsi que d'un générateur de challenges aléatoires ; il faut alors être capable de résoudre $n+1$ challenges après avoir fait au plus n appels à l'oracle (sur des valeurs quelconques qui ne sont pas nécessairement choisies parmi des $n+1$ challenges générés). Le fait d'avoir moins de contrôle sur les requêtes à l'oracle ne permet dans ce cas d'utiliser des méthodes type calcul d'indices pour la résolution de ces problèmes que lorsque n est suffisamment grand.

Chapitre 5

Courbes algébriques

L'un des objectifs de la cryptographie à clef publique est de trouver des familles de groupes dans lesquels le problème du logarithme discret est difficile. Comme il est souvent délicat de prouver la difficulté d'un problème, on considère en pratique qu'un groupe est sûr lorsqu'il résiste depuis suffisamment longtemps aux attaques répétées des cryptanalystes. À cette aune, les groupes associés aux courbes algébriques bénéficient d'une renommée particulière puisque depuis leur introduction dans le domaine par Koblitz et Miller au milieu des années 80 [Kob87, Mil86a], aucune attaque générale n'a remis en cause leur utilisation en cryptographie. De plus, ces groupes peuvent être décrits assez simplement, au moins dans le cas (hyper-)elliptique, et l'on dispose de formules explicites et d'algorithmes rapides pour calculer la loi de composition. Malgré cette simplicité apparente, la théorie mathématique sous-jacente est particulièrement riche et reste un sujet de recherche actif. Pour comprendre les propriétés de ces groupes et développer de nouvelles attaques, il est indispensable de bien connaître les fondements de cette partie de la géométrie algébrique.

Dans la première moitié de ce chapitre, on propose un bref résumé des résultats principaux qui permettent d'expliquer l'utilisation qui est faite des courbes algébriques en cryptographie. On détaille particulièrement la structure des variétés jacobiennes de courbes définies sur des corps finis, en se concentrant sur les cas elliptiques et hyperelliptiques. On explique ensuite les attaques connues du DLP sur variétés jacobiennes : possibilité de transfert du DLP sur un groupe plus fragile, et méthodes de calcul d'indices suivant le genre et le degré de la courbe. À ces attaques ne sont vulnérables qu'une minorité de courbes elliptiques ou hyperelliptiques de genre 2 ; par contre, les courbes de genre supérieur ou égal à 3 ne sont plus considérées comme suffisamment sûres pour les applications cryptographiques.

5.1 Préliminaires

On donne dans cette section les éléments de géométrie algébrique qui seront nécessaires dans la suite de cette thèse. Le but n'est pas de faire un panaroma d'un domaine aussi vaste ; on renvoie aux premiers chapitres de [Sil86] pour une introduction détaillée aux courbes algébriques, et à [Mil86c] pour la construction des variétés jacobiennes. La plupart des démonstrations des résultats donnés seront omises.

5.1.1 Courbes

Variétés affines

Soient $\mathbb{K}$ un corps algébriquement clos et I un idéal de $\mathbb{K}[X_1, \ldots, X_n]$. On rappelle que la *variété affine* associée à I est l'ensemble

$$\mathbb{V}(I) = \{(x_1, \ldots, x_n) \in \mathbb{K}^n : f(x_1, \ldots, x_n) = 0,\ \forall f \in I\}.$$

Réciproquement, un sous-ensemble de $\mathbb{K}^n$ est une variété s'il peut s'écrire sous la forme $\mathbb{V}(I)$ pour un certain idéal $I \subset \mathbb{K}[X_1, \ldots, X_n]$. La topologie de Zariski de $\mathbb{K}^n$ est celle pour laquelle ces ensembles sont exactement les fermés. Il est possible que deux idéaux différents engendrent la même variété ; cependant si l'on se restreint aux idéaux *radicaux* (i.e. tels que $f^n \in I \Rightarrow f \in I$), le Nullstellensatz de Hilbert montre que la correspondance entre l'ensemble des variétés affines de $\mathbb{K}^n$ et l'ensemble des idéaux radicaux de $\mathbb{K}[X_1, \ldots, X_n]$ est une bijection. On note $I(V)$ l'idéal radical correspondant à une variété V.

Un espace topologique E est dit *irréductible* si pour tous fermés F_1, F_2 de E tels que $E = F_1 \cup F_2$, on a $E = F_1$ ou $E = F_2$ (i.e. E n'est pas la réunion de deux sous-ensembles fermés stricts). Une variété affine est alors irréductible pour la topologie de Zariski si et seulement si elle est de la forme $\mathbb{V}(I)$ où I est un idéal premier. En pratique, toute variété algébrique admet une décomposition finie en composantes irréductibles $V = \cup_i V_i$ (unique à permutation des facteurs près) telle que $V_i \nsubseteq V_j$ pour tout $i \neq j$. On définit alors la *dimension* de V comme le maximum des entiers d tels qu'il existe une suite $Z_0 \subset Z_1 \subset \cdots \subset Z_d$ de fermés distincts irréductibles de V ; en particulier, la dimension de V est égale au maximum des dimensions de ses composantes irréductibles. On peut montrer que cette notion coïncide avec la dimension de l'idéal I telle que définie au chapitre 1.

Soit k un corps dont la clôture algébrique est $\mathbb{K}$. Un idéal I de $\mathbb{K}[X_1, \ldots, X_n]$ est dit *défini sur* k s'il admet un système de générateurs dans $k[X_1, \ldots, X_n]$. Similairement, une variété V est *définie sur* k, et on note $V_{|k}$, si $I(V)$ est défini sur k. L'ensemble $V(k)$ des points k-rationnels de V est alors par définition l'intersection $V \cap k^n$. Si $f_1, \ldots, f_m \in k[X_1, \ldots, X_n]$ est un système de générateurs de I définis sur k, une autre caractérisation de l'ensemble des points k-rationnels de $V = \mathbb{V}(I)$ est $V(k) = \{(x_1, \ldots, x_n) \in k^n : f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0\}$. Soit $\mathrm{Gal}(\mathbb{K}/k)$ le groupe de Galois absolu de k ; il agit naturellement sur $\mathbb{K}[X_1, \ldots, X_n]$ via l'action sur les coefficients des polynômes, ainsi que sur l'espace affine $\mathbb{K}^n$ coordonnées par coordonnées, et on a $\sigma(f(P)) = f^\sigma(\sigma(P))$ pour tout $f \in \mathbb{K}[X_1, \ldots, X_n]$ et tout $P \in \mathbb{K}^n$. Si V est définie sur k, alors $I(V)$ et V sont laissés globalement invariants par ces actions et on a $V(k) = \{P \in V : \sigma(P) = P, \forall \sigma \in \mathrm{Gal}(\mathbb{K}/k)\}$. On remarque finalement qu'une variété V définie sur k est en particulier définie sur L pour tout corps $k \subset L \subset \mathbb{K}$, ce qui permet de considérer l'ensemble $V(L)$ de ses points L-rationnels pour toute extension algébrique L de k.

Si V est une variété affine définie sur k, son *anneau de coordonnées affines* est le quotient

$$k[V] = k[X_1, \ldots, X_n]/(I(V) \cap k[X_1, \ldots, X_n]).$$

Quand V est irréductible, cet anneau est intègre et son corps de fraction, noté $k(V)$, est appelé *corps de fonctions* de V ; son degré de transcendance sur k est égal à la dimension de V. Ici encore, $\mathrm{Gal}(\mathbb{K}/k)$ agit naturellement sur $\mathbb{K}[V]$, resp. $\mathbb{K}(V)$, et l'ensemble des éléments laissés fixes est $k[V]$, resp. $k(V)$. À tout élément de $k[V]$ correspond une application de $V(k) \to k$ donnée par l'évaluation en un point de $V(k)$. Il n'y a pas d'équivalent pour $k(V)$: tout élément $f \in k(V)$ correspond à une fonction à valeurs dans k définie seulement sur un ouvert dense de Zariski de $V(k)$, en général

différent de $V(k)$ lui-même. Les points en dehors du domaine de définition correspondent soit à des pôles de f, soit à des points d'indétermination (de la forme $0/0$).

Soit P un point de V. On note $M_P = \{f \in \mathbb{K}[V] : f(P) = 0\}$, c'est un idéal maximal de $\mathbb{K}[V]$. L'*anneau local* $\mathbb{K}[V]_P$ de V en P est le localisé de $\mathbb{K}[V]$ en M_p ; si V est irréductible, il s'identifie à un sous-anneau du corps de fonctions, i.e.

$$\mathbb{K}[V]_P = \{f \in \mathbb{K}(V) : \exists g, h \in \mathbb{K}[V], f = g/h \text{ et } h(P) \neq 0\}.$$

Une fonction f est *régulière en* P si elle appartient à l'anneau local en P et dans ce cas l'évaluation de f en P a un sens. On définit *l'espace cotangent* à V en P comme le $\mathbb{K}$-espace vectoriel M_P/M_P^2 ; si V est irréductible, une définition équivalente de l'espace cotangent est comme le quotient m_P/m_P^2, où m_P est l'idéal maximal de $\mathbb{K}[V]_P$. On dit que V est *lisse* au point P si la dimension de l'espace cotangent en P est égale à la dimension de V, et que V est lisse si elle l'est en chacun de ses points.

Variétés projectives

Soit I un idéal homogène de $\mathbb{K}[X_0, \ldots, X_n]$. La *variété projective* associée à I est l'ensemble

$$\mathbb{V}(I) = \{[x_0 : \ldots : x_n] \in \mathbb{P}^n(\mathbb{K}) : f(x_0, \ldots, x_n) = 0,\ \forall f \in I, f \text{ homogène}\}.$$

Comme dans le cas affine, l'ensemble des variétés projectives forme les fermés de la topologie de Zariski de $\mathbb{P}^n$, et on a une correspondance bijective entre les idéaux homogènes radicaux et les variétés projectives de $\mathbb{P}^n$. On dit aussi qu'un idéal homogène est défini sur k s'il est engendré par des polynômes homogènes de $k[X_0, \ldots, X_n]$, et qu'une variété projective est définie sur k si l'idéal associé est défini sur k ; l'ensemble de ses points k-rationnels est encore

$$V(k) = V \cap \mathbb{P}^n(k) = \{P \in V : \forall \sigma \in \mathrm{Gal}(\mathbb{K}/k),\ \sigma(P) = P\}.$$

L'irréducibilité et la dimension se définissent similairement au cas affine.

Soit H un hyperplan de $\mathbb{P}^n$. Son complémentaire U_H a une structure naturelle d'espace affine $\mathbb{A}^n \simeq \mathbb{K}^n$, pour lequel H joue le rôle d'hyperplan à l'infini. On peut alors faire le lien entre les notions de variété affine et projective : en particulier, si V est une variété projective alors $V \cap U_H$ est une variété affine, et réciproquement si V' est une variété affine de U_H alors son adhérence $\overline{V'}$ dans $\mathbb{P}^n$ est une variété projective telle que $\overline{V'} \cap U_H = V'$. Si H est donné par une équation $X_i = 0$, on note alors U_i la *carte affine* correspondante, et pour toute variété projective V, l'idéal $I(V \cap U_i)$ s'obtient à partir de l'idéal homogène $I(V)$ en déshomogénéisant par rapport à X_i :

$$I(V \cap U_i) = \{f(X_0, \ldots, X_{i-1}, 1, X_{i+1}, \ldots, X_n) \in \mathbb{K}[X_0, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n] : f \in I(V)\}.$$

Réciproquement, si V' est une variété affine de U_i, alors $I(\overline{V'})$ est obtenu en homogénéisant $I(V')$ (cf section 1.1.4). Par ailleurs, si V (resp. V') est définie sur k alors $V \cap U_i$ (resp. $\overline{V'}$) est définie sur k.

Soit V une variété projective, on dit que V est *lisse* au point P s'il existe une carte affine U_H contenant le point P telle que $V \cap U_H$ est lisse au point P. Si V est irréductible (i.e. si $I(V)$ est premier), alors pour toute carte affine U_H, la variété affine $V \cap U_H$ est irréductible et on a soit $V \cap U_H = \emptyset$, soit $\overline{V \cap U_H} = V$. On définit alors le corps de fonctions $k(V)$ de V comme étant le corps de fonctions de $V \cap U_i$ pour tout choix de U_i tel que $V \cap U_i \neq \emptyset$. Une autre caractérisation de $k(V)$ est donnée par

$$k(V) = \{f/g \in k(X_0, \ldots, X_n) : g \notin I(V), f \text{ et } g \text{ homogènes de même degré}\}/\sim,$$

où $f/g \sim f'/g'$ si $fg' - f'g \in I(V)$. Similairement, l'anneau local de V en P est défini comme l'anneau local en P de $V \cap U_i$ pour toute carte U_i contenant P.

Morphismes

Soient $V_1 \subset \mathbb{P}^m$ et $V_2 \subset \mathbb{P}^n$ deux variétés projectives irréductibles. Une *application rationnelle* de V_1 dans V_2 est une fonction ϕ d'un ouvert dense $U \subset V_1$ dans V_2 qui est localement donnée par des fractions rationnelles : pour tout point $P_0 \in U$, il existe un ouvert $U' \subset U$ contenant P_0 et des fonctions $f_0, \ldots, f_n \in \mathbb{K}(V_1)$ régulières sur U' tels que $\phi(P) = [f_0(P) : \ldots : f_n(P)]$ pour tout $P \in U'$. On demande de plus que le domaine de définition U soit maximal pour cette propriété. Une application rationnelle est *régulière* en un point P si P appartient à son domaine de définition, et est appelée *morphisme* si elle est régulière en tout point de V_1. Si V_1 et V_2 sont définies sur k, le groupe de Galois $\mathrm{Gal}(\mathbb{K}/k)$ agit sur les applications rationnelles de V_1 dans V_2 de façon naturelle par $\phi^\sigma(P) = [f_0^\sigma(P) : \ldots : f_n^\sigma(P)]$. Une application rationnelle ou un morphisme ϕ est définie sur k si $\phi^\sigma = \phi$ pour tout $\sigma \in \mathrm{Gal}(\mathbb{K}/k)$ (ou de façon équivalente, si l'on peut choisir $f_0, \ldots, f_n \in k(V_1)$).

Si $\phi : V_1 \to V_2$ et $\phi' : V_2 \to V_3$ sont deux applications rationnelles, il n'est pas forcément possible de les composer car l'image de ϕ peut ne pas rencontrer le domaine de définition de ϕ'. Pour garantir l'existence de la composition, on demande que $\phi(V_1)$ soit dense dans V_2 : une telle application rationnelle est appelée *dominante*. Si ϕ définie sur k est dominante et $f \in k(V_2)$, le *tiré en arrière* $\phi^*(f) = f \circ \phi$ est dans $k(V_1)$. L'application ϕ^* induit alors une extension de corps $k(V_2) \to k(V_1)$ fixant k ; le degré de cette extension $[k(V_1) : \phi^*(k(V_2))]$ est appelé *degré de* ϕ.

Enfin, une application rationnelle dominante $\phi : V_1 \to V_2$ est *birationnelle* si elle admet un inverse, i.e. une application rationnelle $\psi : V_2 \to V_1$ telle que $\phi \circ \psi$ et $\psi \circ \phi$ soient l'identité sur des ouverts denses. On dit alors que V_1 et V_2 sont *birationnellement équivalentes* ; en particulier leurs corps de fonctions sont isomorphes (cette condition est en fait suffisante, voir ci-dessous pour le cas des courbes).

Courbes

Une courbe est une variété (affine ou projective) irréductible de dimension 1. La propriété principale d'une courbe $\mathcal{C}$ est que pour tout point P lisse de $\mathcal{C}$, l'anneau local en P est un *anneau de valuation discrète*. Si $f \in \mathbb{K}(\mathcal{C})$ est régulière en P, on définit son *ordre d'annulation* (ou juste ordre) en P comme étant le plus grand entier $v = \mathrm{ord}_P(f)$ tel que $f \in m_P^v$. Sinon $1/f$ est régulière en P et on pose $\mathrm{ord}_P(f) = -\mathrm{ord}_P(1/f)$; on dit alors que f a un pôle d'ordre $\mathrm{ord}_P(1/f)$ en P. On appelle *uniformisante* en P toute fonction $t \in \mathbb{K}(\mathcal{C})$ telle que $\mathrm{ord}_P(t) = 1$. Une autre conséquence est qu'une application rationnelle $\phi : \mathcal{C} \to V$ où V est une variété projective, est régulière en tout point lisse de $\mathcal{C}$; en particulier, si $\mathcal{C}$ est lisse, alors toute application rationnelle de $\mathcal{C}$ dans une variété projective V est en fait un morphisme.

Soient $\mathcal{C}_1$ et $\mathcal{C}_2$ deux courbes projectives définies sur k, et $\phi : \mathcal{C}_1 \to \mathcal{C}_2$ définie sur k ; on peut montrer que ϕ est soit constante soit surjective. Dans le second cas, on a vu que ϕ induit une extension de corps $k(\mathcal{C}_1)/\phi^*(k(\mathcal{C}_2))$ fixant k : cette extension est nécessairement algébrique (puisque le degré de transcendance des deux corps est 1) de degré fini. Réciproquement, si l'on se donne une injection $\imath : k(\mathcal{C}_2) \to k(\mathcal{C}_1)$ fixant k, il existe une unique application rationnelle ϕ définie sur k de $\mathcal{C}_1$ vers $\mathcal{C}_2$ telle que $\phi^* = \imath$.

On a ainsi une correspondance étroite entre les courbes et leurs corps de fonctions, que l'on va préciser. Un corps F est appelé *corps de fonctions* sur k, et on note F/k, si F est une extension de degré de transcendance 1 de k. Le *corps des constantes* de F est égal à $F \cap \mathbb{K}$ (où $\mathbb{K}$ est la clôture algébrique de k). En particulier, le corps de fonctions $k(\mathcal{C})$ d'une courbe $\mathcal{C}$ définie sur k est bien un corps de fonctions au sens précédent, de corps de constantes k ; on peut montrer que tout corps

de fonctions F/k de corps de constantes k s'obtient ainsi pour une courbe unique à application birationnelle près. Comme par ailleurs toute courbe est birationnellement équivalente à une courbe lisse, on obtient une équivalence (contravariante) de catégories :

$$\begin{array}{ccc} \left\{\begin{array}{c}\text{Courbes lisses définies sur } k\\ \text{Morphismes non constants}\\ \text{définis sur } k\end{array}\right\} & \longrightarrow & \left\{\begin{array}{c}\text{Corps de fonctions } F/k\\ \text{de corps de constantes } k\\ \text{Extensions de corps fixant } k\end{array}\right\} \\ \mathcal{C}_{|k} & \longmapsto & k(\mathcal{C}) \\ \phi : \mathcal{C}_1 \to \mathcal{C}_2 & \longmapsto & \phi^* : k(\mathcal{C}_2) \to k(\mathcal{C}_1) \end{array} \tag{5.1}$$

5.1.2 Diviseurs, genre

Soit $\mathcal{C}$ une courbe projective lisse. Le groupe des diviseurs de $\mathcal{C}$, noté $\mathrm{Div}(\mathcal{C})$, est le groupe abélien libre engendré par les points de $\mathcal{C}$, i.e. un diviseur est une somme formelle $\sum_{P\in\mathcal{C}} n_P(P)$ à coefficients dans $\mathbb{Z}$ de points de $\mathcal{C}$, où tous les coefficients sont nuls sauf un nombre fini. Le *degré* d'un diviseur $D = \sum_{P\in\mathcal{C}} n_P(P)$ est donné simplement par $\deg(D) = \sum_{P\in\mathcal{C}} n_P$; on s'intéresse principalement au sous-groupe des diviseurs de degré 0 noté $\mathrm{Div}^0(\mathcal{C})$. Si $\mathcal{C}$ est définie sur k, on a encore une action de $\mathrm{Gal}(\mathbb{K}/k)$ sur $\mathrm{Div}(\mathcal{C})$ et $\mathrm{Div}^0(\mathcal{C})$, telle que $D^\sigma = \sum_{P\in\mathcal{C}} n_P(\sigma(P)) = \sum_{P\in\mathcal{C}} n_{\sigma^{-1}(P)}(P)$. Un diviseur D est alors défini sur k si $D^\sigma = D$ pour tout $\sigma \in \mathrm{Gal}(\mathbb{K}/k)$, et on note $\mathrm{Div}_k(\mathcal{C})$ et $\mathrm{Div}^0_k(\mathcal{C})$ les sous-groupes correspondants ; on remarque que $\mathrm{Div}_k(\mathcal{C})$ est plus gros que le groupe abélien libre engendré par $\mathcal{C}(k)$.

Si f est un élément non nul de $\mathbb{K}(\mathcal{C})$, elle n'a qu'un nombre fini de zéros et de pôles, ce qui permet de lui associer le diviseur $\mathrm{div}(f) = \sum_P \mathrm{ord}_P(f)\,(P)$ dont on peut montrer que le degré est 0. Lorsque f est définie sur k, le diviseur correspondant est également défini sur k. On dit que D est un diviseur *principal* s'il existe $f \in \mathbb{K}(\mathcal{C})^*$ telle que $D = \mathrm{div}(f)$. Comme $\mathrm{div}(f_1 f_2) = \mathrm{div}\,(f_1) + \mathrm{div}\,(f_2)$, l'ensemble des diviseurs principaux forme un sous-groupe de $\mathrm{Div}^0(\mathcal{C})$.

Un diviseur $D = \sum_{P\in\mathcal{C}} n_P(P)$ est *effectif* si $n_P \geq 0$ pour tout $P \in \mathcal{C}$; on note $D \geq 0$. On peut mettre une relation d'ordre partiel sur l'ensemble des diviseurs en posant $D_1 \geq D_2$ si $D_1 - D_2 \geq 0$. À un diviseur $D \in \mathrm{Div}(\mathcal{C})$, on associe l'ensemble

$$\mathcal{L}(D) = \{f \in \mathbb{K}(\mathcal{C})^* : \mathrm{div}\,(f) \geq -D\} \cup \{0\},$$

qui est un $\mathbb{K}$-espace vectoriel de dimension finie ; on note $\ell(D)$ sa dimension. On vérifie aisément que $\mathcal{L}(D) = \{0\}$ si $\deg(D) < 0$, et que $\mathcal{L}(0)$ est l'ensemble des fonctions constantes. Par ailleurs, si $\mathcal{C}$ est définie sur k et que $D \in \mathrm{Div}_k(\mathcal{C})$, alors il existe une base de $\mathcal{L}(D)$ constituée d'éléments de $k(\mathcal{C})^*$.

Le lien entre le degré d'un diviseur D et la dimension de l'espace $\mathcal{L}(D)$ associé est donné par le théorème suivant, qui permet de définir un invariant fondamental des courbes algébriques.

Théorème 5.1.1 (Riemann-Roch)**.** *Soit $\mathcal{C}$ une courbe projective lisse. Il existe un unique entier g, appelé* genre *de $\mathcal{C}$, et un diviseur $W \in \mathrm{div}\,(\mathcal{C})$ tels que pour tout $D \in \mathrm{div}\,(\mathcal{C})$,*

$$\ell(D) - \ell(W - D) = \deg(D) - g + 1.$$

Remarque 5.1.2.

1. *Un tel diviseur W est appelé* diviseur canonique*. En prenant $D = 0$, on obtient que $\ell(W) = g$, en prenant ensuite $D = W$, on trouve que* $\deg(W) = 2g - 2$.

2. Si $\deg(D) > 2g-2$*, alors* $\deg(W-D) < 0$ *et donc* $\ell(D) = \deg(D) - g + 1$*.*

À isomorphisme près, il existe une seule courbe de genre 0, la droite projective $\mathbb{P}^1(k)$. Une courbe de genre 1 admettant un point rationnel est appelée *courbe elliptique*, ces courbes constituent le sujet d'étude central de cette thèse. Par analogie, on peut définir le genre d'un corps de fonctions comme étant le genre de la courbe projective lisse associée. Un corps de fonctions de genre 0 est donc rationnel, i.e. de la forme $k(x)$ pour un élément x transcendant sur k.

Un premier résultat faisant intervenir le genre est la *borne de Hasse*. Si $\mathcal{C}$ est une courbe projective lisse définie sur un corps fini $\mathbb{F}_q$, il est clair que la cardinalité de l'ensemble de ses points $\mathbb{F}_q$-rationnels est finie ; le genre permet d'en donner un encadrement. Une conséquence des célèbres *conjectures de Weil* est la formule

$$q + 1 - 2g\sqrt{q} \leq \#\mathcal{C}(\mathbb{F}_q) \leq q + 1 + 2g\sqrt{q} \; ; \tag{5.2}$$

il existe de plus des relations entre les cardinalités des $\mathcal{C}(\mathbb{F}_{q^n})$, $n \in \mathbb{N}^*$, voir [CFA+06, §8.1.1].

5.1.3 Groupe de Picard et jacobienne

Deux diviseurs D_1 et D_2 sont *linéairement équivalents*, et on note $D_1 \sim D_2$, s'il existe une fonction f telle que $D_1 - D_2 = \operatorname{div}(f)$. Le *groupe de Picard* (ou groupe de classes) est le quotient du groupe des diviseurs par le sous-groupe des diviseurs principaux, c'est donc l'ensemble des classes d'équivalence pour la relation $\sim$. On s'intéresse principalement à sa partie de degré 0, notée $\mathrm{Pic}^0(\mathcal{C})$, obtenue comme quotient de $\mathrm{Div}^0(\mathcal{C})$ par les diviseurs principaux. Comme précédemment, on définit $\mathrm{Pic}^0_k(\mathcal{C})$ comme étant le sous-groupe des éléments laissés fixes par l'action de $\mathrm{Gal}(\mathbb{K}/k)$. Dans la suite, on notera souvent de la même façon les éléments de $\mathrm{Div}^0(\mathcal{C})$ et leurs classes dans le groupe de Picard.

Si $\mathcal{C}_1$ et $\mathcal{C}_2$ sont deux courbes projectives lisses, à tout morphisme $\phi : \mathcal{C}_1 \to \mathcal{C}_2$ correspondent des homomorphismes entre les groupes des diviseurs et les groupes de Picard de $\mathcal{C}_1$ et $\mathcal{C}_2$. Si P est un point de $\mathcal{C}_1$, on commence par définir l'*indice de ramification* de ϕ en P comme étant l'entier $e_\phi(P) = \mathrm{ord}_P(\phi^*(t_{\phi(P)}))$, où $t_{\phi(P)} \in \mathbb{K}(\mathcal{C}_2)$ est une uniformisante en $\phi(P)$. Pour un diviseur élémentaire $(Q) \in \mathrm{Div}(\mathcal{C}_2)$ (i.e. $n_P = 0$ si $P \neq Q$ et $n_Q = 1$), on définit son *tiré en arrière* par ϕ comme étant le diviseur

$$\phi^*(Q) = \sum_{P \in \phi^{-1}(\{Q\})} e_\phi(P)(P) \; ;$$

par linéarité, on étend cette définition à un homomorphisme $\phi^* : \mathrm{Div}(\mathcal{C}_2) \to \mathrm{Div}(\mathcal{C}_1)$. Cette opération est compatible avec le tiré en arrière pour les fonctions : pour tout $f \in \mathbb{K}(\mathcal{C}_2)^*$, on a $\phi^*(\operatorname{div}(f)) = \operatorname{div}(\phi^*(f))$. Comme de plus ϕ^* envoie un diviseur de degré 0 sur un diviseur de degré 0, ce tiré en arrière induit un homomorphisme encore noté ϕ^* de $\mathrm{Pic}^0(\mathcal{C}_2)$ vers $\mathrm{Pic}^0(\mathcal{C}_1)$, qui se restreint en un homomorphisme $\mathrm{Pic}^0_k(\mathcal{C}_2) \to \mathrm{Pic}^0_k(\mathcal{C}_1)$ si $\mathcal{C}_1, \mathcal{C}_2$ et ϕ sont définis sur k. On définit aussi un homomorphisme *poussé en avant* $\phi_* : \mathrm{Div}(\mathcal{C}_1) \to \mathrm{Div}(\mathcal{C}_2)$, tel que

$$\phi_*(\sum n_P(P)) = \sum n_P(\phi(P)).$$

L'image d'un diviseur de degré 0 est encore de degré 0, et l'image d'un diviseur principal est encore principal ; plus précisément, $\phi_*(\operatorname{div}(f)) = \operatorname{div}(\phi_*(f))$, où $\phi_* : \mathbb{K}(\mathcal{C}_1) \to \mathbb{K}(\mathcal{C}_2)$ est l'application norme relative à l'extension $\mathbb{K}(\mathcal{C}_1)/\phi^*(\mathbb{K}(\mathcal{C}_2))$. En conséquence, le poussé en avant induit bien un homomorphisme de $\mathrm{Pic}^0(\mathcal{C}_1)$ dans $\mathrm{Pic}^0(\mathcal{C}_2)$, qui se restreint aux sous-groupes définis sur k si ϕ est définie sur k. De plus, on a $(\phi_1 \circ \phi_2)^* = \phi_2^* \circ \phi_1^*$ et $(\phi_1 \circ \phi_2)_* = \phi_{1*} \circ \phi_{2*}$.

Le théorème de Riemann-Roch permet de donner une description des éléments de $\mathrm{Pic}^0(\mathcal{C})$. Soient $\mathcal{O}$ un point de $\mathcal{C}$, et D un diviseur de degré 0. D'après Riemann-Roch, on a $\ell(D+g(\mathcal{O})) \geq \deg(D+g(\mathcal{O}))-g+1=1$, donc $\mathcal{L}(D+g(\mathcal{O}))$ n'est pas réduit à $\{0\}$. Soit f un élément non nul de $\mathcal{L}(D+g(\mathcal{O}))$; le diviseur $\mathrm{div}\,(f)+D+g(\mathcal{O})$ est effectif de degré g, donc de la forme $(P_1)+\cdots+(P_g)$ où $P_1,\ldots,P_g \in \mathcal{C}$. On a ainsi montré que tout diviseur D de degré 0 est linéairement équivalent à un diviseur de la forme $(P_1)+\cdots+(P_g)-g(\mathcal{O})$. Quitte à supprimer les points P_i égaux à $\mathcal{O}$, on a finalement $D \sim (P_1)+\cdots+(P_r)-r(\mathcal{O})$ où $r \leq g$. On peut alors montrer sans difficulté que cette écriture est unique lorsque r est minimal.

Si $\mathcal{C}$ est définie sur k, elle est en particulier définie sur tout corps intermédiaire entre k et $\mathbb{K}$, ce qui permet de définir les groupes $\mathrm{Pic}^0_L(\mathcal{C})$ pour toute extension algébrique L de k. Un résultat fondamental de la théorie des courbes algébriques est qu'il existe une variété projective lisse irréductible définie sur k, appelée *variété jacobienne* de $\mathcal{C}$ et notée $\mathrm{Jac}_{\mathcal{C}}$, telle que les éléments de $\mathrm{Pic}^0_L(\mathcal{C})$ s'identifient avec l'ensemble des points L-rationnels de $\mathrm{Jac}_{\mathcal{C}}$. De plus, la loi de groupe sur $\mathrm{Jac}_{\mathcal{C}}$ correspondant à celle de $\mathrm{Pic}^0(\mathcal{C})$ est algébrique, i.e. les opérations de groupe sur $\mathrm{Jac}_{\mathcal{C}}$ sont données par des applications rationnelles régulières. Autrement dit, la variété jacobienne d'une courbe est une *variété abélienne*, c'est-à-dire une variété algébrique projective qui est aussi un groupe algébrique.

Pour une courbe projective $\mathcal{C}_{|k}$ donnée, on définit son *produit symétrique* $\mathcal{C}^{(g)} = \mathcal{C}^g/\mathfrak{S}_g$ comme étant l'ensemble des g-uplets non ordonnés d'éléments de $\mathcal{C}$; il s'agit d'une variété projective de dimension g définie sur k. Si l'on se fixe un point $\mathcal{O} \in \mathcal{C}(k)$, on a vu à l'aide du théorème de Riemann-Roch que l'application

$$\begin{array}{rcl} \psi_{\mathcal{O}} : \mathcal{C}^{(g)} & \to & \mathrm{Pic}^0(\mathcal{C}) \\ (P_1,\ldots,P_g) & \mapsto & (P_1)+\cdots+(P_g)-g(\mathcal{O}) \end{array}$$

est surjective, et on peut en fait montrer que $\mathcal{C}^{(g)}$ est birationnellement équivalent à $\mathrm{Jac}_{\mathcal{C}}$. Dans la pratique, on travaillera presque exclusivement avec les jacobiennes de courbes elliptiques ou hyperelliptiques, pour lesquelles on a une description plus précise.

Si $\mathcal{C}$ est définie sur un corps fini $\mathbb{F}_q$, les conjectures de Weil permettent aussi d'encadrer la cardinalité de la jacobienne en fonction du genre de $\mathcal{C}$:

$$(\sqrt{q}-1)^{2g} \leq \#\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_q) \leq (\sqrt{q}+1)^{2g}. \tag{5.3}$$

En particulier, à g fixé la cardinalité de la jacobienne de $\mathcal{C}$ sur $\mathbb{F}_q$ est égale à $q^g\left(1+O(1/\sqrt{q})\right)$.

5.1.4 Courbes elliptiques

Par définition, une *courbe elliptique* E est une courbe projective lisse de genre 1, munie d'un point distingué $\mathcal{O}$ (k-rationnel si E est défini sur k). On peut alors raffiner la description donnée ci-dessus du groupe de Picard :

Propriété 5.1.3. *L'application $\psi_{\mathcal{O}} : E \to \mathrm{Pic}^0(E)$, $P \mapsto (P)-(\mathcal{O})$ est bijective. En particulier, la courbe E s'identifie à sa variété jacobienne.*

Démonstration. On a déjà vu que l'application $\psi_{\mathcal{O}}$ est surjective. Si maintenant $(P)-(\mathcal{O}) \sim (P')-(\mathcal{O})$, alors il existe $f \in \mathbb{K}(E)$ telle que $\mathrm{div}\,(f)=(P)-(P')$, et en particulier $f \in \mathcal{L}((P'))$. D'après Riemann-Roch $\ell((P'))=1$, et comme $\mathcal{L}((P'))$ contient les fonctions constantes, on a $\mathcal{L}((P'))=\mathbb{K}$. Donc $\mathrm{div}\,(f)=0$, et $P=P'$; l'application $\psi_{\mathcal{O}}$ est aussi injective. □

Une courbe elliptique est habituellement donnée par une équation de Weierstrass, qui s'obtient en considérant la suite des espaces $\mathcal{L}(i(\mathcal{O}))$ pour $i = 1 \dots 6$. D'après Riemann-Roch, on a $\ell(i(\mathcal{O})) = i$, et clairement $\mathcal{L}((\mathcal{O})) = \mathbb{K}$. On peut ensuite trouver des fonctions $x \in \mathcal{L}(2(\mathcal{O})) \setminus \mathcal{L}((\mathcal{O}))$ et $y \in \mathcal{L}(3(\mathcal{O})) \setminus \mathcal{L}(2(\mathcal{O}))$, de telle sorte que $\mathrm{ord}_{\mathcal{O}}(x) = -2$ et $\mathrm{ord}_{\mathcal{O}}(y) = -3$. L'espace $\mathcal{L}(6(\mathcal{O}))$ contient alors les fonctions $1, x, y, x^2, xy, x^3$ et y^2. Comme $\ell(6(\mathcal{O})) = 6$, il existe une relation de dépendance linéaire entre ces sept fonctions, qui s'écrit (quitte à multiplier x et y par des constantes)

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \tag{5.4}$$

On montre alors que la courbe E est isomorphe à la courbe dont la partie affine a pour équation (5.4), le point $\mathcal{O}$ correspondant au point à l'infini $[0 : 1 : 0]$. En pratique, on peut simplifier cette équation par des changements de variables simples, et on considère principalement des équations de Weierstrass *réduites* de la forme $y^2 = x^3 + Ax + B$ en caractéristique différente de 2 ou 3, et de la forme $y^2 + xy = x^3 + \alpha x^2 + \beta$ ou $y^2 + ay = x^3 + bx + c$ en caractéristique 2.

La classification des courbes elliptiques à isomorphisme près est relativement simple. On peut associer à chaque courbe elliptique E un élément de $\mathbb{K}$, appelé son *j-invariant*, qui s'exprime comme une fraction rationnelle en les coefficients de l'équation de Weierstrass. Comme cette expression est relativement compliquée, on ne la donne que pour les équations réduites :

$$j(E) = \begin{cases} 1\,728\dfrac{4A^3}{4A^3 + 27B^2} & \text{lorsque } y^2 = x^3 + Ax + B, \\ 1/\beta & \text{lorsque } y^2 + xy = x^3 + \alpha x^2 + \beta, \\ 0 & \text{lorsque } y^2 + ay = x^3 + bx + c. \end{cases}$$

On montre alors que :
- deux courbes elliptiques sont isomorphes si et seulement si elles ont le même j-invariant ;
- pour tout $j \in \mathbb{K}$, il existe une courbe elliptique E de j-invariant égal à j ; de plus si j appartient à un sous-corps k, on peut choisir E définie sur k.

En revanche, si E et E' sont deux courbes elliptiques définies sur k de même j-invariant, l'isomorphisme entre E et E' n'est pas forcément défini sur k : si l'on exclut les cas particuliers $j = 0$ et $j = 1\,728$, un tel isomorphisme est défini soit sur k, soit sur une extension quadratique de k. Une courbe $E'_{|k}$ qui est isomorphe à $E_{|k}$ sans lui être k-isomorphe est appelée une *tordue* de E, quadratique si $j \notin \{0; 1\,728\}$ (il existe aussi des tordues cubiques et sextiques si $j = 0$ ou $1\,728$). Si k est un corps fini, il admet une unique extension quadratique, et chaque j-invariant différent de $0, 1\,728$ correspond donc à deux classes de k-isomorphisme.

5.1.5 Courbes hyperelliptiques

Une courbe *hyperelliptique* $\mathcal{H}$ est une courbe projective lisse telle qu'il existe un morphisme de degré 2 de $\mathcal{H}$ sur la droite projective $\mathbb{P}^1$. En conséquence, son corps de fonctions $\mathbb{K}(\mathcal{H})$ est une extension algébrique de degré 2 du corps de fonctions rationnelles $\mathbb{K}(\mathbb{P}^1) = \mathbb{K}(x)$ et est de la forme $\mathbb{K}(x, y)$ où y vérifie une équation

$$y^2 + h_0(x)y = h_1(x), \qquad h_0, h_1 \in \mathbb{K}[x]. \tag{5.5}$$

Une courbe elliptique est donc un cas particulier de courbe hyperelliptique. En général, la courbe projective plane dont la partie affine a pour équation (5.5) n'est pas lisse et est seulement birationnelle à $\mathcal{H}$; on travaille cependant le plus souvent avec un modèle plan dont les seuls points

singuliers étant à l'infini. L'application $\iota : \mathcal{H} \to \mathcal{H}$, $(x, y) \mapsto (x, -y - h_0(x))$, s'appelle l'*involution hyperelliptique* : la projection sur $\mathbb{P}^1$ envoie un élément de $\mathcal{H}$ et son image par ι sur le même point.

Le genre g de la courbe $\mathcal{H}$ dépend des degrés de h_0 et h_1 (et du type de la singularité à l'infini). En caractéristique différente de 2, un simple changement de variable $y \mapsto y - h_0/2$ permet de se ramener à une équation de la forme $y^2 = h_1(x)$, où le degré de h_1 est égal à $2g + 1$ ou $2g + 2$. On travaille principalement dans le cas où $\deg h_1 = 2g + 1$; il y a alors un seul point à l'infini $\mathcal{O}_{\mathcal{H}}$, et on parle de modèle (ou équation) *imaginaire*. Lorsque $\deg h_1 = 2g + 2$, le modèle est dit *réel* et possède deux points à l'infini. Si h_1 possède une racine x_0 dans k, un changement de variable de la forme $(x, y) \mapsto \big((ax + b)/(x - x_0), y/(x - x_0)^{g+1}\big)$ permet de se ramener à un modèle imaginaire.

Représentation de Mumford

Soit $\mathcal{H}$ une courbe hyperelliptique de genre g admettant un modèle imaginaire d'équation $y^2 + h_0(x)y = h_1(x)$, où $h_0, h_1 \in k[x]$. Si D est un diviseur de $\mathrm{Div}^0(\mathcal{H})$, on a vu qu'il est linéairement équivalent à un unique diviseur, dit *réduit*, de la forme $(P_1) + \cdots + (P_r) - r(\mathcal{O}_{\mathcal{H}})$ avec r minimal. On a nécessairement $P_i \neq \iota(P_j)$ pour tout $i \neq j$, sinon on pourrait simplifier l'écriture de D en utilisant la fonction $x - x_{P_i}$ de diviseur $(P_i) + (\iota(P_i)) - 2(\mathcal{O}_{\mathcal{H}})$. On associe alors à D les polynômes $u(x) = \prod(x - x_{P_i})$ et $v(x)$ tel que $v(x_{P_i}) = y_{P_i}$ pour tout i de 1 à r, $\deg v < r$ et $u | (v^2 + vh_0 - h_1)$. Si $P_i \neq P_j$ pour tout $i \neq j$, $v(x)$ s'obtient directement par interpolation de Lagrange ; la condition de divisibilité sert juste à déterminer v dans le cas où u a des racines multiples.

Proposition 5.1.4. *L'ensemble des points k-rationnels de la jacobienne de $\mathcal{H}$ est en bijection avec les paires de polynômes $(u, v) \in k[x]^2$ tels que u est unitaire,* $\deg(v) < \deg(u) \leq g$, *et* $u | (v^2 + vh_0 - h_1)$.

5.2 Arithmétique des courbes elliptiques et hyperelliptiques

On explicite dans cette section les lois de composition interne définies sur les courbes elliptiques et les jacobiennes de courbes hyperelliptiques. On détaille aussi la structure du groupe des points $\mathbb{F}_q$-rationnels pour une courbe définie sur un corps fini.

5.2.1 Genre 1

Soit E une courbe elliptique d'équation de Weierstrass $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$. Pour déterminer la loi de groupe sur $E = \mathrm{Jac}_E$, on considère les diviseurs associés à des équations de droites.

1. Si $f = x - c$, la droite verticale d'équation $f = 0$ coupe E en deux points (comptés avec multiplicité), et on a $\mathrm{div}\,(f) = (P_1) + (P_2) - 2(\mathcal{O})$. En particulier, $(P_2) - (\mathcal{O}) \sim -\big((P_1) - (\mathcal{O})\big)$.
2. Si $f = y - (\lambda x + \mu)$, la droite d'équation $f = 0$ coupe E en trois points (comptés avec multiplicité), et on a $\mathrm{div}\,(f) = (P_1) + (P_2) + (P_3) - 3(\mathcal{O})$. En particulier, $\big((P_1) - (\mathcal{O})\big) + \big((P_2) - (\mathcal{O})\big) \sim -\big((P_3) - (\mathcal{O})\big)$.

On rappelle que E est isomorphe à $\mathrm{Jac}_E \simeq \mathrm{Pic}^0(E)$ via l'identification $\psi_{\mathcal{O}} : P \mapsto (P) - (\mathcal{O})$. On déduit de ce qui précède les formules suivantes : pour $P_1 = (x_1, y_1)$ et $P_2 = (x_2, y_2)$, on a

$$-P_1 = (x_1, -y_1 - a_1x_1 - a_3) \quad \text{et} \quad P_1 + P_2 = (x_3, y_3) \text{ où } \begin{cases} x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 - a_1x_3 - a_3 \end{cases}$$

$$\text{avec } \lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{si } x_1 \neq x_2, \\ \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} & \text{si } P_1 = P_2. \end{cases}$$

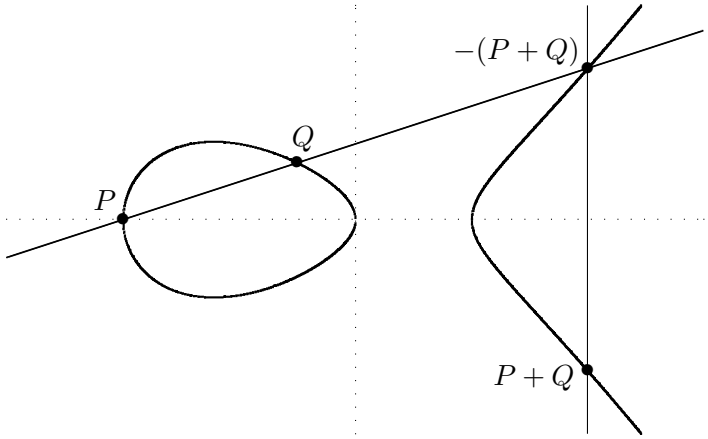


FIGURE 5.1 – Exemple d'addition de points sur une courbe elliptique.

En pratique, on travaille avec des équations réduites où les coefficients a_i sont tous nuls sauf un ou deux, ce qui simplifie ces formules. Il existe de nombreuses méthodes pour accélérer ces calculs en utilisant le moins possible de multiplications et surtout d'inversions, comme par exemple l'utilisation de coordonnées projectives, jacobiennes, de Chudnovsky, mixtes, d'Edwards... (voir la base de données disponible sur http://hyperelliptic.org/EFD/ pour plus de détails).

Soient E_1 et E_2 deux courbes elliptiques. Une *isogénie* φ de E_1 dans E_2 est un morphisme (i.e. une application rationnelle régulière) tel que $\varphi(\mathcal{O}_1) = \mathcal{O}_2$ où $\mathcal{O}_1$ et $\mathcal{O}_2$ sont les points à l'infini de E_1 et E_2 respectivement. Une propriété fondamentale est qu'une isogénie est aussi un morphisme de groupes :

$$\varphi(P + Q) = \varphi(P) + \varphi(Q), \ \forall P, Q \in E_1.$$

Cela découle directement de la commutativité du diagramme suivant :

$$\begin{array}{ccc} E_1 & \xrightarrow{\varphi} & E_2 \\ {\scriptstyle \psi_{\mathcal{O}_1}}\downarrow\wr & & \wr\downarrow{\scriptstyle \psi_{\mathcal{O}_2}} \\ \mathrm{Pic}^0(E_1) & \xrightarrow{\varphi_*} & \mathrm{Pic}^0(E_2) \end{array}$$

Une isogénie $\phi : E \to E$ est appelée *endomorphisme*. En itérant la loi de groupe, on définit l'application de multiplication par un scalaire $m \in \mathbb{Z}$, que l'on note habituellement $[m] : P \mapsto [m]P$; c'est un endomorphisme de E de degré m^2. On appelle *point de m-torsion* un élément du noyau de $[m]$, et on note leur ensemble $E[m] = \{P \in E : [m]P = \mathcal{O}\}$; on note aussi $E(k)[m]$ l'ensemble

des points k-rationnels de m-torsion. Soit $p = \mathrm{char}(\mathbb{K})$ la caractéristique du corps ; le fait que la multiplication par m soit de degré m^2 permet de montrer que $E[m] \simeq (\mathbb{Z}/m\mathbb{Z})^2$ si $p = 0$ ou $m \wedge p = 1$, et que $E[p^\alpha] \simeq \mathbb{Z}/p^\alpha\mathbb{Z}$ ou $\{\mathcal{O}\}$ si $p \neq 0$.

Si E est définie sur un corps fini $\mathbb{F}_q$, l'ensemble de ses points rationnels forme un groupe commutatif fini $E(\mathbb{F}_q)$; en particulier tous ses éléments sont de torsion. En utilisant le théorème de structure des groupes commutatifs finis ainsi que la forme des groupes de torsion, on obtient un isomorphisme

$$E(\mathbb{F}_q) \simeq \mathbb{Z}/n_1\mathbb{Z} \times \mathbb{Z}/n_2\mathbb{Z}, \text{ avec } n_1|n_2 \text{ et } p \nmid n_1.$$

On peut de plus montrer à l'aide du couplage de Weil (voir section 5.3.1) que $n_1|(q-1)$.

5.2.2 Genre supérieur

Soit $\mathcal{H}$ une courbe hyperelliptique de genre g admettant un modèle imaginaire de la forme $y^2 + h_0(x)y = h_1(x)$, avec $\deg(h_1) = 2g+1$ et $\deg(h_0) \leq g$. On rappelle le principe de la représentation de Mumford : à tout diviseur $D \in \mathrm{Div}^0(\mathcal{H})$ de la forme $(P_1)+\ldots+(P_r)-r(\mathcal{O}_\mathcal{H})$ tel que $\iota(P_i) \neq P_j$ pour tout $i \neq j$, on associe l'unique couple de polynômes (u, v) tels que $u = \prod(x - x_{P_i})$, $v(x_{P_i}) = y_{P_i}$, $\deg v < \deg u$ et $u|(v^2 + vh_0 - h_1)$; cette représentation est valable y compris pour des diviseurs qui ne sont pas minimaux. Avec cette représentation, on peut décrire l'algorithme d'addition dans $\mathrm{Jac}_\mathcal{H}$ dû à Cantor [Can87], qui reprend des techniques de composition et réduction de formes quadratiques remontant à Gauss et Lagrange.

Soient $D_1 = (u_1, v_1)$ et $D_2 = (u_2, v_2)$ deux diviseurs réduits sous forme de Mumford ; pour simplifier, on va supposer que $u_1 \wedge u_2 = 1$. La première étape consiste à trouver la représentation de Mumford (u_3, v_3) de $D_3 = D_1 + D_2$. Clairement, on a $u_3 = u_1u_2$; pour trouver v_3, on calcule avec l'algorithme d'Euclide étendu les polynômes a_1 et a_2 tels que $a_1u_1 + a_2u_2 = 1$ et on pose $v_3 = a_1u_1v_2 + a_2u_2v_1 \bmod u_3$. La deuxième étape consiste à réduire le diviseur $D_3 = (P_1) + \cdots + (P_m) - m(\mathcal{O}_\mathcal{H})$ si $m > g$. L'idée est d'utiliser pour cela le diviseur principal associé à $y - v_3$, de la forme $(P_1)+\cdots+(P_m)+(Q_1)+\cdots+(Q_\ell)-(m+\ell)(\mathcal{O}_\mathcal{H})$ avec $\ell < m$, pour remplacer D_3 par le diviseur linéairement équivalent $D_4 = (\iota Q_1)+\cdots+(\iota Q_\ell)-\ell(\mathcal{O}_\mathcal{H})$. Pour trouver la représentation de Mumford de D_4, on constate que $(y - v_3)(y + h_0 + v_3) = h_1 - v_3h_0 - v_3^2 = \prod_{i=1}^m(x - x_{P_i}) \prod_{i=1}^\ell(x - x_{Q_i}) = u_3 \prod_{i=1}^\ell(x - x_{\iota(Q_i)})$. On a donc $u_4 = (h_1 - v_3h_0 - v_3^2)/u_3$ et $v_4 = -v_3 - h_0 \bmod u_4$. On recommence ce processus avec D_4 jusqu'à obtenir un diviseur réduit. On donne en algorithme 20 le pseudo-code de cette méthode d'addition, en prenant en compte le cas où u_1 et u_2 ne sont pas premiers entre eux.

Algorithme 20: Algorithme de Cantor

Entrées : Deux diviseurs sous forme de Mumford $D_1 = (u_1, v_1)$ et $D_2 = (u_2, v_2)$, les polynômes h_0 et h_1 définissant la courbe $\mathcal{H} : y^2 + h_0y = h_1$.
Sortie : $D_3 = (u_3, v_3)$ diviseur réduit linéairement équivalent à $D_1 + D_2$

1. Calculer avec Euclide étendu a_1, a_2, a_3, d tels que $d = u_1 \wedge u_2 \wedge (v_1 + v_2 + h_0) = a_1u_1 + a_2u_2 + a_3(v_1 + v_2 + h_0)$;
2. $u_3 \leftarrow u_1u_2/d^2$;
3. $v_3 \leftarrow (a_1u_1v_2 + a_2u_2v_1 + a_3(v_1v_2 + h_1))/d \bmod u_3$;
4. **tant que** $\deg u_3 > g$ **faire**
5. $\quad u_3 \leftarrow (h_1 - v_3h_0 - v_3^2)/u_3$;
6. $\quad v_3 \leftarrow -v_3 - h_0 \bmod u_3$;
7. **retourner** $(u_3/LC(u_3), v_3)$;

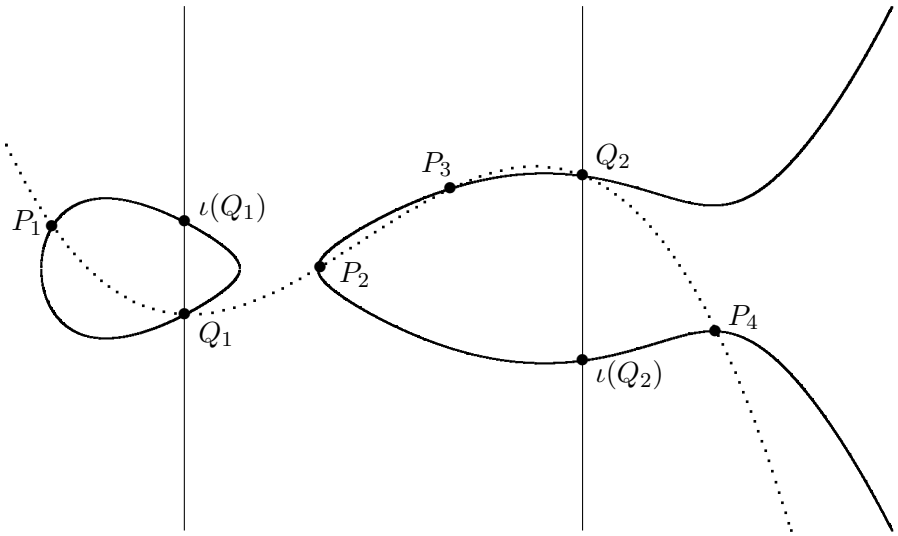


FIGURE 5.2 – Exemple d'addition de points sur une courbe hyperelliptique de genre 2.

On montre qu'en utilisant des techniques classiques pour la multiplication et le pgcd de polynômes, la complexité du calcul de la loi de groupe est en $O(g^2)$ opérations dans le corps de base $\mathbb{F}_q$. Il existe un certain nombre d'améliorations à l'algorithme de Cantor. On peut simplifier son déroulement pour certaines entrées spécifiques, typiquement lorsque $D_1 = D_2$ où les calculs de pgcd sont facilités. Quand le genre est grand, on peut jouer sur la phase de réduction pour l'accélérer, et on peut utiliser des algorithmes de calcul du produit et du pgcd de deux polynômes à base de transformée de Fourier rapide, pour une complexité asymptotique en $O(g \log^2 g \log\log g)$ opérations dans $\mathbb{F}_q$. Il existe aussi des optimisations spécifiques aux genres 2 et 3, et pour certains types de courbes ; on renvoie à [BSS05, chapitre VII] pour plus de détails.

Similairement au cas elliptique, si $\mathcal{H}$ est une courbe hyperelliptique de genre g définie sur $\mathbb{F}_q$, l'ensemble des points $\mathbb{F}_q$-rationnels de sa jacobienne forme un groupe commutatif fini dont la structure est

$$\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q) \simeq \mathbb{Z}/n_1\mathbb{Z} \times \cdots \times \mathbb{Z}/n_{2g}\mathbb{Z},$$

où $n_i | n_{i+1}$ pour $1 \leq i < 2g$ et $n_i | (q-1)$ pour $i \leq g$.

5.3 Attaques spécifiques du logarithme discret

Soit $\mathcal{H}$ une courbe (hyper-)elliptique définie sur un corps fini $\mathbb{F}_q$. On a vu que l'ensemble $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ des points $\mathbb{F}_q$-rationnels de sa jacobienne forme un groupe commutatif fini, dont la loi est calculable en un temps polynomial en la taille du groupe. De tels groupes sont donc des candidats naturels pour la pratique de la cryptographie basée sur le problème du logarithme discret. Il est à noter que pour être réellement utile en cryptographie, il faut être capable de produire des courbes $\mathcal{H}$ telles que $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ admette un sous-groupe d'ordre grand premier, ce que l'on sait faire actuellement de deux façons différentes : soit en tirant des courbes aléatoires et en utilisant des algorithmes efficaces de comptage de points jusqu'à obtenir un grand facteur premier, soit en cherchant directement une jacobienne de cardinalité prescrite en utilisant les méthodes de multiplication complexe [CFA+06, chapitres 17 et 18].

Les courbes algébriques sont particulièrement intéressantes pour la cryptographie dans la mesure où, en général, on ne connaît pas d'attaque du DLP plus performante que les génériques quand le genre g est inférieur ou égal à 2 ; dans le cas où $g \geq 3$, on peut mettre en place des algorithmes de calcul d'indices (voir section 5.4). Cependant dans des situations bien particulières, on peut transférer le DLP vers un groupe où sa sécurité est plus faible. On présente ici deux attaques par transfert, l'une vers le groupe multiplicatif d'un corps fini, et l'autre vers un corps p-adique. Un autre type de transfert (par descente de Weil) sera détaillé en chapitre 6. Pour simplifier, on se limitera au cas des courbes elliptiques, bien que les deux attaques s'appliquent également aux variétés jacobiennes.

5.3.1 Transfert par couplage

L'existence de groupes sur lesquels on peut définir un *couplage* effectivement calculable fournit un nouveau type de fonction à sens unique, sur lesquelles on peut baser de nouveaux protocoles en cryptographie à clef publique. A contrario, ces couplages offrent une structure supplémentaire permettant éventuellement d'attaquer le DLP. On rappelle la définition de couplage dans un cadre général.

Définition 5.3.1. *Soient G_1 et G_2 deux groupes d'exposant n notés additivement, et G_3 un groupe cyclique d'ordre n noté multiplicativement. Un* couplage *est une application $e : G_1 \times G_2 \to G_3$, qui est*

- *bilinéaire : pour tout $g_1 \in G_1$, $g_2 \in G_2$, $a, b \in \mathbb{Z}/n\mathbb{Z}$, on a $e([a]g_1, [b]g_2) = e(g_1, g_2)^{ab}$;*
- *non dégénéré : pour tout $g_1 \in G_1 \setminus \{0\}$, il existe $g_2 \in G_2$ tel que $e(g_1, g_2) \neq 1$; de même pour tout $g_2 \in G_2 \setminus \{0\}$, il existe $g_1 \in G_1$ tel que $e(g_1, g_2) \neq 1$.*

Les groupes associés aux courbes algébriques sont naturellement munis de couplages. Un des plus important est le *couplage de Weil* vérifiant les propriétés suivantes :

Propriété 5.3.2. *Soient $E_{|k}$ une courbe elliptique et n un nombre premier à la caractéristique de k. Il existe un couplage $w_n : E[n] \times E[n] \to \mu_n \subset \mathbb{K}^*$ (où μ_n est le groupe des racines n-ièmes de l'unité), appelé* couplage de Weil*, qui est*

(i) antisymétrique : $w_n(P_1, P_2) = w_n(P_2, P_1)^{-1}$;

(ii) Galois-invariant : pour tout $\sigma \in \mathrm{Gal}(\mathbb{K}/k)$, $w_n(\sigma(P_1), \sigma(P_2)) = \sigma(w_n(P_1, P_2))$.

On s'intéresse surtout au cas où k est un corps fini $\mathbb{F}_q$. L'invariance par Galois montre que si P_1 et P_2 sont définis sur une extension $\mathbb{F}_{q^d}$, le couplage $w_n(P_1, P_2)$ appartient à $\mathbb{F}_{q^d}$. L'algorithme de Miller permet alors de calculer ce couplage de façon effective avec une complexité en $O(\log n)$ opérations dans $E(\mathbb{F}_{q^d})$ [Mil04].

La non-dégénérescence et l'invariance par Galois impliquent que si toute la n-torsion de E est définie sur $\mathbb{F}_{q^d}$, alors $\mu_n \subset \mathbb{F}_{q^d}^*$, i.e. $n|(q^d - 1)$. Ce résultat admet une réciproque partielle :

Proposition 5.3.3 (Balasubramanian-Koblitz [BK98])**.**
Soient E une courbe elliptique définie sur $\mathbb{F}_q$ et r un nombre premier tel que $r \wedge q = 1$ et $r|\#E(\mathbb{F}_q)$. Si $r \nmid (q - 1)$, alors $E[r] \subset E(\mathbb{F}_{q^d})$ si et seulement si $r|(q^d - 1)$.

Le plus petit entier d_p tel que $n|(q^{d_p} - 1)$ est appelé *degré de plongement* (associé à q et à n) ; ainsi, le couplage de Weil w_n est à valeurs dans $\mathbb{F}_{q^{d_p}}$. Le degré de plongement est un paramètre

fondamental pour mesurer la sécurité des courbes utilisées en cryptographie et l'efficacité des protocoles à base de couplages. En effet, on peut définir d'autres couplages sur une courbe elliptique (par exemple pour s'affranchir de la propriété d'antisymétrie de w_n), mais tous font intervenir le degré de plongement ; le plus utilisé en cryptographie est celui de Tate-Lichtenbaum :

$$t_n : E(\mathbb{F}_{q^{d_p}})[n] \times E(\mathbb{F}_{q^{d_p}})/[n]E(\mathbb{F}_{q^{d_p}}) \to \mathbb{F}^*_{q^{d_p}}/(\mathbb{F}^*_{q^{d_p}})^n \simeq \mu_n.$$

Lorsqu'on dispose d'un couplage $e : G_1 \times G_2 \to G_3$, on peut naturellement transférer le DLP de G_1 dans G_3. Soient $g \in G_1$ d'ordre n et $h \in \langle g \rangle$ un élément de G_1 dont on veut calculer le logarithme x en base g. La non-dégénérescence du couplage implique qu'il existe $g_2 \in G_2$ tel que $e(g, g_2) \neq 1$ soit d'ordre n. Comme $e(h, g_2) = e([x]g, g_2) = e(g, g_2)^x$, on se ramène à calculer le logarithme de $e(h, g_2)$ en base $e(g, g_2)$. Dans le cas où $G_1 = E(\mathbb{F}_q)[r]$ et $G_3 = \mu_r \subset \mathbb{F}^*_{q^{d_p}}$ où r premier divise $\#E(\mathbb{F}_q)$ et d_p est le degré de plongement associé à q et r, on retrouve l'attaque de Menezes-Okamoto-Vanstone [MOV93] avec le couplage de Weil et l'attaque de Frey-Rück [FR94] avec le couplage de Tate. Comme il existe des algorithmes de calcul d'indices de complexité sous-exponentielle pour résoudre le DLP dans $\mathbb{F}^*_{q^{d_p}}$, ce transfert va être efficace dès lors que le degré de plongement n'est pas trop gros. En particulier, si $E_{|\mathbb{F}_q}$ est *supersingulière* (i.e. $E(\overline{\mathbb{F}_q})[q] = \{\mathcal{O}\}$), on peut montrer que le degré de plongement pour tout diviseur de $\#E(\mathbb{F}_q)$ est inférieur ou égal à 6, et même à 2 si q est premier strictement supérieur à 3. Utiliser de telles courbes en cryptographie impose donc de travailler avec des tailles de clefs plus importantes que celles nécessaires pour résister aux attaques génériques.

Il est à noter cependant que pour une courbe elliptique *ordinaire* (i.e. non supersingulière) quelconque définie sur $\mathbb{F}_q$, le degré de plongement attendu est généralement grand (de l'ordre de q), de telle sorte que le simple calcul d'un couplage est inaccessible. Ce transfert ne peut donc pas s'appliquer à la grande majorité des courbes elliptiques utilisées en cryptographie.

5.3.2 Courbes anomales

Soient $\mathbb{F}_q$ un corps fini avec $q = p^\alpha$ (p premier) et $\mathbb{Q}_q$ le corps des nombres q-adiques correspondant [CFA+06, chapitre 3]. L'application de réduction modulo p associe à toute courbe elliptique $\mathcal{E}$ définie sur $\mathbb{Q}_q$ une courbe E définie sur $\mathbb{F}_q$, éventuellement singulière, et induit une application $\pi : \mathcal{E}(\mathbb{Q}_q) \to E(\mathbb{F}_q)$ [Sil86, chapitre VII]. Dans le cas où E est non singulière, cette application est un homomorphisme dont on note $\mathcal{E}_1(\mathbb{Q}_q) = \{\tilde{P} \in \mathcal{E}(\mathbb{Q}_q) : \tilde{P} = \mathcal{O}_E \bmod p\}$ le noyau. Il se trouve que la structure de $\mathcal{E}_1$ est relativement simple, en particulier c'est un groupe où le DLP est facile. En effet, il existe une fonction logarithme q-adique $\vartheta_q : \mathcal{E}_1(\mathbb{Q}_q) \to p\mathbb{Z}_q$ telle que $\vartheta_q(P_1+P_2) = \vartheta_q(P_1)+\vartheta_q(P_2)$, et cette fonction est donnée par une série entière explicite en la coordonnée $z = -x/y$. De plus, si l'on note $\mathcal{E}_2(\mathbb{Q}_q) = [p]\mathcal{E}_1(\mathbb{Q}_q)$ l'image de $\mathcal{E}_1$ par la multiplication par p, on a un isomorphisme $\mathcal{E}_1(\mathbb{Q}_q)/\mathcal{E}_2(\mathbb{Q}_q) \simeq p\mathbb{Z}_q/p^2\mathbb{Z}_q \simeq (\mathbb{F}_q, +)$ donné par le logarithme q-adique ϑ_q modulo p^2.

Il est alors naturel d'essayer de transférer le DLP d'une courbe $E_{|\mathbb{F}_q}$ vers le groupe $\mathcal{E}_1(\mathbb{Q}_q)$ pour un relevé $\mathcal{E}_{|\mathbb{Q}_q}$ de E. Cette technique va fonctionner pour les sous-groupes de $E(\mathbb{F}_q)$ d'ordre une puissance de p.

Définition 5.3.4. *Soit $\mathbb{F}_q$ un corps fini avec $q = p^\alpha$, p premier. On appelle* courbe anomale *toute courbe $E_{|\mathbb{F}_q}$ telle que $\#E(\mathbb{F}_q) = 0$ mod p.*

Soient E une courbe anomale, $P \in E(\mathbb{F}_q)$ un point d'ordre p et $Q \in \langle P \rangle$ un point dont on veut calculer le logarithme x en base P. On choisit un relevé q-adique $\mathcal{E}$ de E, ainsi que des relevés $\tilde{P}$

et $\tilde{Q}$ de P et Q, qui ne sont pas de p-torsion ; on note que $\tilde{R} = [x]\tilde{P} - \tilde{Q}$ est un élément de $\mathcal{E}_1$, en général non nul. Comme P et Q sont d'ordre p, on sait que les points $[p]\tilde{P}$ et $[p]\tilde{Q}$ sont dans $\mathcal{E}_1$, et que $[p]\tilde{R} = [x]([p]\tilde{P}) - [p]\tilde{Q}$ est dans $\mathcal{E}_2$. En prenant les logarithmes q-adiques, on trouve que $x\,\vartheta_q([p]\tilde{P}) - \vartheta_q([p]\tilde{Q}) = 0 \bmod p^2$, et que

$$x = \vartheta_q([p]\tilde{Q})/\vartheta_q([p]\tilde{P}) \bmod p.$$

En pratique, aucun calcul n'a pas besoin de dépasser la précision 3 et la complexité de cette attaque est polynomiale en $\log q$. Cette minorité de courbes est donc à éviter absolument pour toute application cryptographique.

5.4 Calcul d'indices en genre $g > 2$

Soit $\mathcal{H}_{|\mathbb{F}_q}$ une courbe hyperelliptique de genre g, admettant un modèle imaginaire. La représentation de Mumford des éléments de la jacobienne $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ de $\mathcal{H}$ permet de définir la *taille* d'un diviseur $D = (u, v)$ comme étant le degré du polynôme u, ce qui donne une notion de "petits" diviseurs, analogue de celle des petits nombres premiers du calcul d'indices sur $(\mathbb{Z}/p\mathbb{Z})^*$. Par ailleurs, la loi de composition explicitée en section 5.2.2 montre que si u se factorise sur $\mathbb{F}_q$ en $u = \prod_j u_j$, alors en posant $v_j = v \bmod u_j$, on a $D_j = (u_j, v_j) \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ et $D = \sum_j D_j$. Ce type de factorisation permet de mettre en place sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ un calcul d'indices tel que décrit en section 4.3.1. Pour l'analyse, on distinguera selon que le genre croît asymptotiquement vers l'infini ou reste fixé.

5.4.1 Genre grand

Le premier algorithme de calcul d'indices sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ est dû à Adleman, Demarrais et Huang [ADH94] ; une version améliorée plus simple est donnée dans [EG02], c'est celle qui est présentée dans cette section. On choisit une borne de lissité B (on verra ensuite comment trouver sa valeur optimale), et on pose comme base de factorisation

$$\mathcal{F} = \{D = (u, v) \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q) : u \text{ irréductible}, \ \deg(u) \leq B\}.$$

Soit D_2 le diviseur dont on veut trouver le logarithme discret en base D_1. Pendant la phase de recherche de relations, on considère des éléments aléatoires de $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ de la forme $[a_i]D_1$, et on calcule la décomposition en facteurs irréductibles dans $\mathbb{F}_q[x]$ du polynôme u associé. Cette décomposition s'obtient efficacement avec un algorithme probabiliste similaire à celui présenté en section 1.2.2 pour rechercher les racines d'un polynôme sur $\mathbb{F}_q$. Si le polynôme $u = \prod_j u_j$ n'a que des facteurs de degré inférieur ou égal à B, alors $[a_i]D_1$ se décompose dans la base $\mathcal{F}$. On recommence jusqu'à obtenir suffisamment de relations, après quoi on procède à la phase d'algèbre linéaire et à la phase de descente si nécessaire.

On remarque qu'il est possible d'utiliser l'involution hyperelliptique ι pour diminuer la taille de la base de factorisation d'un facteur 2, puisqu'on sait que tout diviseur D est linéairement équivalent à $-\iota(D)$. On choisit donc pour $\mathcal{F}$ un système de représentant du quotient par ι. Ce type de réduction de la base est général et sera appliqué systématiquement ; si la courbe ou la jacobienne possède d'autres automorphismes simples, ils peuvent aussi être utilisés pour réduire la base de factorisation.

Pour trouver la valeur optimale de la borne de lissité B, il faut estimer la taille de la base de factorisation correspondante ainsi que le nombre d'éléments B-lisses de $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$. Enge et Stein

ont montré qu'on retrouve le même comportement que dans le cas du groupe multiplicatif d'un corps fini :

Théorème 5.4.1 ([ES02])**.**
Soit $B = \lceil \log_q(L_{q^g}(1/2, \rho)) \rceil$ *une borne de lissité pour une constante positive* ρ *donnée. Le nombre de diviseurs* B*-lisses de* $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ *est minoré par*

$$\frac{q^g}{L_{q^g}(1/2, 1/2\rho + o(1))}.$$

Suivant Enge et Gaudry [EG02], la valeur asymptotiquement optimale de B est alors de l'ordre de $\log_q(L_{q^g}(1/2, 1/\sqrt{2}))$ lorsque q^g tend vers $+\infty$, et $\log(q) = o(g)$. On obtient ainsi une complexité asymptotique pour la résolution du DLP sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ en

$$L_{q^g}(1/2, \sqrt{2} + o(1)).$$

5.4.2 Genre petit

Lorsque le genre est petit, l'analyse précédente donnerait une borne de lissité strictement inférieure à 1, ce qui n'a bien sûr pas de sens. L'idée de Gaudry [Gau00] consiste alors à prendre $B = 1$; ceci revient à considérer une base de factorisation constituée de diviseurs admettant une représentation de Mumford de la forme $(x - a, b) \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$, la recherche de relations se résumant ainsi à tester si un diviseur donné admet dans sa représentation de Mumford un polynôme $u \in \mathbb{F}_q[x]$ qui est scindé.

Quand q est grand par rapport à g, la probabilité qu'un polynôme de degré g de $\mathbb{F}_q[x]$ soit scindé est de l'ordre de $1/g!$. Par ailleurs, la taille de la base de factorisation contient clairement de l'ordre de q éléments ; la phase de recherche de relations nécessite alors environ $q\,g!$ tests de décomposition, et la phase d'algèbre linéaire a une complexité en $O(g\,q^2)$ opérations dans $\mathbb{F}_q$ (cf. section 4.3.1).

À g fixé lorsque q tend vers l'infini, les deux phases du calcul d'indices ont un coût déséquilibré, l'algèbre linéaire en q^2 étant la plus coûteuse. Pour équilibrer les deux étapes, une idée – introduite en premier par Harley – est de réduire de façon arbitraire la base de factorisation en prenant pour $\mathcal{F}$ un sous-ensemble de $\{(x - a, b) \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)\}$. Soit $\alpha \in [0; 1]$ tel que $\#\mathcal{F} = q^\alpha$. La probabilité qu'un diviseur se décompose dans $\mathcal{F}$ devient alors de l'ordre de $q^{(\alpha-1)g}/g!$ pour un coût total de la phase de recherche de relations en $\tilde{O}(q^\alpha q^{(1-\alpha)g})$, et l'algèbre linéaire est en $\tilde{O}(q^{2\alpha})$. La valeur asymptotiquement optimale de α est donc $1 - 1/(g + 1)$, pour une complexité totale en

$$\tilde{O}\left(q^{2-2/(g+1)}\right), \text{ à } g \text{ fixé lorsque } q \to \infty.$$

On peut néanmoins faire mieux en appliquant les méthodes "large primes" présentées en section 4.3.2. En plus de la base réduite $\mathcal{F}$ de cardinal q^α, on considère l'ensemble $\mathcal{F}' = \{(x - a, b) \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)\} \setminus \mathcal{F}$ des "grands diviseurs". Avec la variante "one large prime", la probabilité d'obtenir une relation faisant intervenir $g-1$ éléments de $\mathcal{F}$ et un seul élément de $\mathcal{F}'$ est en $q^{(\alpha-1)(g-1)}/(g-1)!$. On a vu qu'il faut de l'ordre de $q^{(\alpha+1)/2}$ relations "one large prime" pour éliminer les grands diviseurs, donc le coût total de la phase de recherche de relations est en $\tilde{O}(q^{(\alpha+1)/2} q^{(1-\alpha)(g-1)})$. Comme l'algèbre linéaire reste en $\tilde{O}(q^{2\alpha})$, la valeur asymptotiquement optimale de α est $1 - 1/(g + 1/2)$, pour une complexité totale en

$$\tilde{O}\left(q^{2-2/(g+1/2)}\right), \text{ à } g \text{ fixé lorsque } q \to \infty.$$

Finalement, la meilleure complexité asymptotique est obtenue avec la méthode "double large primes" de [GTTD07]. La probabilité d'obtenir une relation avec deux facteurs dans $\mathcal{F}'$ et $g-2$ dans $\mathcal{F}$ est en $q^{(\alpha-1)(g-2)}/2(g-2)!$ et on a besoin d'environ $q+q^\alpha \approx q$ relations pour éliminer les éléments de $\mathcal{F}'$. Le coût de la première phase est donc en $\tilde{O}(q\, q^{(1-\alpha)(g-2)})$ et la valeur asymptotiquement optimale de α est $1-1/g$, pour une complexité totale en

$$\tilde{O}\left(q^{2-2/g}\right), \text{ à } g \text{ fixé lorsque } q \to \infty.$$

Par comparaison, si $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q)$ admet un sous-groupe d'ordre premier de taille comparable à $\#\,\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_q) \approx q^g$, la complexité d'une attaque générique sur le DLP est en $\tilde{O}(q^{g/2})$. Quand q tend vers $+\infty$, la variante "double large primes" du calcul d'indices est donc plus efficace dès que $g \geq 3$. On note cependant que la complexité de l'attaque est toujours exponentielle en la taille du groupe, et que les attaques génériques restent les plus efficaces en genre $g = 1$ ou 2. Il faut aussi insister sur le fait que ces complexités ne sont valables qu'asymptotiquement à g fixé : à cause des constantes cachées dans les notations de Landau, et en particulier du facteur en $g!$ intervenant dans la phase de recherche de relations, la valeur pratique optimale du paramètre α est en général supérieure à celle donnée.

5.5 Calcul d'indices en petit degré

Plutôt que de chercher à décomposer un multiple d'un diviseur donné, on peut essayer d'obtenir des relations de la forme (4.5) ne faisant intervenir que des éléments de la base de factorisation, par exemple en considérant des diviseurs principaux associés à des fonctions d'expression simple. Sur ce principe, Diem propose dans [Die06] une méthode de résolution du logarithme discret particulièrement efficace pour les jacobiennes de courbes planes de petit degré.

Soient $\mathcal{C} \subset \mathbb{P}^2(\mathbb{F}_q)$ une courbe projective plane admettant une équation de degré d et $\mathcal{C}'$ une courbe lisse qui lui est birationnellement équivalente ; on s'intéresse au DLP dans la jacobienne de $\mathcal{C}'$. Comme les points lisses de $\mathcal{C}$ s'identifient naturellement à des points de $\mathcal{C}'$, on identifiera les sommes formelles de points lisses de $\mathcal{C}$ avec des diviseurs de $\mathcal{C}'$. On note $[X : Y : Z]$ un système de coordonnées homogènes sur $\mathbb{P}^2$. Le diviseur noté D_∞ correspond à l'intersection de $\mathcal{C}$ avec la droite à l'infini, autrement dit $D_\infty = \text{zéros}(Z) = \sum_P \mathrm{ord}_P(Z/F_P)\,(P)$ où F_P est un polynôme homogène de degré 1 ne s'annulant pas en P de telle sorte que Z/F_P appartient au corps de fonctions de $\mathcal{C}$; en particulier, D_∞ est un diviseur de degré d. Soit $\mathcal{O}$ un point lisse de $\mathcal{C}(\mathbb{F}_q)$, on définit une base de factorisation

$$\mathcal{F} = \{(P) - (\mathcal{O}) : P \in \mathcal{C}(\mathbb{F}_q) \text{ lisse } \} \cup \{d(\mathcal{O}) - D_\infty\} \subset \mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q).$$

En dehors de la droite à l'infini, on note $x = X/Z$ et $y = Y/Z$ les coordonnées affines. Pour trouver des relations, on choisit deux points lisses $P_1, P_2 \in \mathcal{C}(\mathbb{F}_q)$ et on considère la droite passant par P_1 et P_2 d'équation $f = 0$, où f est de la forme $y - \lambda x - \mu$ (le cas $x - \mu$ se traite similairement). Cette droite coupe la courbe en au plus $d-2$ autres points (comptés avec multiplicité) que l'on peut facilement déterminer. En remplaçant y par $\lambda x + \mu$ dans l'équation de $\mathcal{C}$, on trouve un polynôme de degré d dont on connaît déjà deux racines ; à l'aide des algorithmes de recherche des solutions donnés en section 1.2.2, on peut facilement déterminer si ce polynôme est scindé sur $\mathbb{F}_q$ et trouver le cas échéant les $d-2$ autres points $P_3, \ldots, P_d$ de l'intersection de la droite avec $\mathcal{C}$. Si tous ces points sont lisses, alors on obtient une relation : en effet $\mathrm{div}(f) = (P_1) + (P_2) + (P_3) + \cdots + (P_d) - D_\infty$, et on a

$$[(P_1) - (\mathcal{O})] + \cdots + [(P_d) - (\mathcal{O})] + [d(\mathcal{O}) - D_\infty] = 0$$

dans $\mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q)$. On recommence ensuite à partir d'un autre couple de points lisses jusqu'à avoir obtenu suffisamment de relations. Dans la phase de descente, on calcule le diviseur réduit de la forme $(P_1)+\cdots+(P_r)-r(\mathcal{O})$ linéairement équivalent à une combinaison $[a]D_1+[b]D_2$ des diviseurs intervenant dans le DLP, jusqu'à ce que les points $P_1,\ldots,P_r$ soient dans la partie lisse de $\mathcal{C}(\mathbb{F}_q)$.

Heuristiquement, la base de factorisation contient de l'ordre de q éléments, de telle sorte que la phase d'algèbre linéaire a une complexité en $\tilde{O}(d\,q^2)$. Pour trouver une relation, il faut qu'un polynôme de degré $d-2$ soit scindé dans $\mathbb{F}_q$, ce qui arrive avec une probabilité d'environ $1/(d-2)!$, et comme il faut de l'ordre de q relations, la phase de recherche de relations est en $\tilde{O}(q\,(d-2)!)$. Encore une fois, il est naturel de vouloir équilibrer les deux phases avec une variation "double large primes". On choisit donc artificiellement une base réduite [1] $\mathcal{F}$ de cardinalité q^α, et on considère la base complémentaire $\mathcal{F}'$ contenant le reste des éléments de la base de factorisation initialement choisie. Pour les équations des droites, on ne prend évidemment en compte que des couples de points (P_1,P_2) tels que les diviseurs correspondants soient dans $\mathcal{F}$. Pour obtenir une relation "double large primes", il faut que l'intersection de $\mathcal{C}$ avec la droite passant par P_1,P_2 contienne $d-2$ autres points dont au plus deux sont dans $\mathcal{F}'$, ce qui arrive avec une probabilité d'environ $q^{(\alpha-1)(d-4)}/2(d-4)!$. Comme on a besoin de l'ordre de q relations pour éliminer les éléments de $\mathcal{F}'$, la complexité de la phase de recherche de relations est en $\tilde{O}(q\cdot q^{(1-\alpha)(d-4)})$ à d fixé lorsque q tend vers l'infini. La complexité de l'algèbre linéaire est toujours en $\tilde{O}(q^{2\alpha})$ et asymptotiquement le meilleur choix pour α est $1-1/(d-2)$, ce qui donne une complexité totale en

$$\tilde{O}(q^{2-2/(d-2)}),\ \text{à } g \text{ fixé, lorsque } q\to\infty.$$

Il est à noter que le choix de α est en fait limité puisqu'il faut pouvoir générer suffisamment de relations. On montre cependant qu'heuristiquement la plus petite taille de base de factorisation permettant d'engendrer de l'ordre de q relations est en $\tilde{O}(q^{1-1/(d-2)})$, de telle sorte que la complexité asymptotique ci-dessus reste valide.

Remarque 5.5.1. *Cette méthode n'a d'intérêt que lorsque $d\geq 4$: en effet, si $d=2$ la courbe est de genre 0 et son groupe de Picard est trivial, et si $d=3$, on ne peut pas appliquer la méthode "double large primes" puisqu'une droite passant par P_1 et P_2 ne recoupe $\mathcal{C}$ qu'en un seul autre point.*

Pour une courbe hyperelliptique, le degré minimal d'un modèle plan est égal à $2g+1$, ce qui rend cette attaque beaucoup moins intéressante que le calcul d'indices classique. Cependant, pour une courbe suffisamment générale de genre $g\geq 3$, on s'attend à ce que le degré minimal d'un modèle plan soit égal à $g+1$, ce qui rend de façon surprenante le DLP sur la jacobienne d'une telle courbe plus fragile que celui sur la jacobienne d'une courbe hyperelliptique de même genre. Cela est particulièrement vrai en genre 3, où toute courbe non-hyperelliptique admet un modèle plan de degré 4 qui correspond à une complexité en $\tilde{O}(q)$ avec l'attaque de Diem, alors que l'on a vu que la complexité du calcul d'indices sur la jacobienne d'une courbe hyperelliptique de même genre est en $\tilde{O}(q^{4/3})$ (à comparer au $\tilde{O}(q^{3/2})$ des attaques génériques).

La raison principale pour laquelle la complexité de l'attaque de Diem est meilleure repose sur le fait que les petits diviseurs sont mieux exploités. En effet, le choix des droites garantit déjà que deux diviseurs de la décomposition sont dans la petite base, ce qui améliore la probabilité qu'une relation soit de type "double large primes". Cette idée sera partiellement réexploitée pour l'analyse asymptotique de la technique de crible que l'on propose en section 7.3.5.

1. Il peut être avantageux de commencer par la phase de descente et d'inclure dans $\mathcal{F}$ tous les éléments apparaissant dans les décompositions obtenues.

Chapitre 6

Transfert du logarithme discret par descente de Weil

En 1998, Frey propose pour la première fois dans [Fre98] d'appliquer le principe de descente de Weil (ou restriction des scalaires) aux variétés abéliennes sur lesquelles on peut définir un problème de logarithme discret en vue d'applications cryptographiques. Ainsi, lorsqu'une variété abélienne est définie sur une extension de corps fini $\mathbb{F}_{q^n}$, on peut transférer le problème du logarithme discret défini sur cette variété à sa restriction de Weil définie sur $\mathbb{F}_q$. L'idée consiste alors à exploiter la structure de cette restriction pour en déduire des informations sur le DLP.

Le premier intérêt de cette méthode est de mettre en évidence la fragilité du DLP sur certaines courbes elliptiques ; plus précisément, lorsqu'il existe une application de transfert explicite de la courbe elliptique à la jacobienne d'une courbe de petit genre ou de petit degré préservant le DLP, on peut utiliser les méthodes de calculs d'indices disponibles sur celle-ci pour rendre plus vulnérable le DLP sur la courbe elliptique de départ. Le deuxième intérêt est d'aider à la construction de cryptosystèmes basés sur courbes hyperelliptiques. Pour s'assurer que la jacobienne de la courbe admet bien un sous-groupe d'ordre un grand nombre premier, il est nécessaire d'en connaître la cardinalité : une possibilité est d'utiliser la descente de Weil pour construire une courbe hyperelliptique dont la jacobienne est isogène à la restriction de Weil d'une courbe elliptique, puis d'utiliser des algorithmes polynomiaux de comptage de points sur courbes elliptiques pour faire le calcul.

Dans ce chapitre, on se concentre sur les nouvelles méthodes apportées par la descente de Weil, pour attaquer le problème du logarithme discret de courbes elliptiques définies sur des extensions de corps finis. Après avoir détaillé la construction de la restriction de Weil d'une variété algébrique, on explique comment construire explicitement l'application de transfert du logarithme discret avec la technique GHS, dans le cas de la caractéristique 2 [GHS02b] et de la caractéristique impaire [Die03].

6.1 Restriction de Weil et recouvrement

6.1.1 Définition et propriétés

Soit $\mathbb{K}$ une extension séparable de degré n d'un corps k. Par le procédé de *restriction des scalaires*, on peut associer à toute variété algébrique V définie sur $\mathbb{K}$ de dimension d, une variété algébrique définie sur k de dimension nd. La variété ainsi obtenue est appelée *restriction de Weil*

de V relativement à l'extension $\mathbb{K}/k$ et est notée $W_{\mathbb{K}/k}(V)$.

Lorsque la variété V est affine, on peut décrire explicitement cette construction en utilisant les coordonnées affines. Soit I l'idéal de $\mathbb{K}[X_1, \ldots, X_r]$ tel que $V = \mathbb{V}(I) \subset \mathbb{K}^r$, on note $\langle f_1, \ldots, f_s \rangle$ une famille de générateurs de cet idéal. Soit $\{u_1, \ldots, u_n\}$ une base du k-espace vectoriel $\mathbb{K}$. On introduit les variables $X_{i,j}$ telles que

$$\forall i \in [\![1; r]\!],\ X_i = X_{i,1}u_1 + \ldots + X_{i,n}u_n.$$

En remplaçant X_i par cette expression dans les polynômes $f_1, \ldots, f_s$, puis en récupérant les coordonnées des polynômes obtenus dans la base $\{u_1, \ldots, u_n\}$, on obtient un système polynomial à rn variables et sn équations défini sur k. La restriction de Weil $\mathcal{W} = W_{\mathbb{K}/k}(V)$ de V est alors définie comme le sous-ensemble fermé de Zariski défini par ces équations. En particulier, il existe une identification naturelle entre les points de $V(\mathbb{K})$ et ceux de $\mathcal{W}(k)$, mais la topologie de Zariski sur $\mathcal{W}(k)$ est plus fine que celle de $V(\mathbb{K})$. Pour obtenir la restriction de Weil d'une variété projective, il est possible de faire une construction analogue dans les cartes affines de V, puis de "recoller" les restrictions de Weil de chaque carte par les applications birationnelles de changement de cartes.

Dans le cas où l'extension $\mathbb{K}/k$ est galoisienne, on peut construire de façon plus intrinsèque la restriction de Weil. Pour tout k-automorphisme σ de $\mathbb{K}$, on définit la variété V^σ, image de $V = V(f_1, \ldots, f_s)$ par σ, par

$$V^\sigma = V(f_1^\sigma, \ldots, f_s^\sigma)$$

où f^σ est le polynôme obtenu à partir de f en appliquant σ à tous ses coefficients ; en particulier $\sigma(f(x)) = f^\sigma(\sigma(x))$ et $(V^\sigma)^\tau = V^{\tau\circ\sigma}$. Si $P = (x_1, \ldots, x_r)$ est un point de V, alors $\sigma(P) = (\sigma(x_1), \ldots, \sigma(x_r))$ est un point de V^σ. La restriction de Weil de V est alors la variété (a priori définie sur $\mathbb{K}$)

$$W_{\mathbb{K}/k}(V) = \prod_{\sigma\in G} V^\sigma = \{(P_\sigma)_{\sigma\in G} : P_\sigma \in V^\sigma\}$$

où $G = \mathrm{Gal}(\mathbb{K}/k)$, munie de l'action "tordue" du groupe de Galois : soient $P = (P_\sigma)_{\sigma\in G}$ un point de $\mathcal{W} = W_{\mathbb{K}/k}(V)$ et $\bar\tau \in \mathrm{Gal}(\overline{k}/k)$, alors $\bar\tau(P) = (Q_\sigma)_{\sigma\in G}$ où $Q_\sigma = \bar\tau(P_{\tau^{-1}\circ\sigma}) \in V^\sigma$ et $\tau \in G$ est la restriction de $\bar\tau$ à $\mathbb{K}$. En particulier, $\mathcal{W}$ étant invariante par le groupe de Galois, elle est bien définie sur k, et l'ensemble de ses points k-rationnels est

$$\mathcal{W}(k) = \{(\sigma(P))_{\sigma\in G} : P \in V(\mathbb{K})\}$$

qui s'identifie naturellement à $V(\mathbb{K})$.

Propriété 6.1.1. *Soient V une variété algébrique définie sur $\mathbb{K}$ de dimension d et $\mathcal{W} = W_{\mathbb{K}/k}(V)$ sa restriction de Weil relativement à l'extension galoisienne $\mathbb{K}/k$. Alors*

- *$\mathcal{W}(k) = V(\mathbb{K})$ et $\mathcal{W}(\ell) = V(L)$ pour toute extension algébrique ℓ de k telle que $L = \ell\mathbb{K}$ soit une extension de degré n de ℓ :*

$$\begin{array}{ccc} \mathbb{K} & \text{——} & L = \ell\mathbb{K} \\ {\scriptstyle n}\big| & & \big|{\scriptstyle n} \\ k & \text{————} & \ell \end{array}$$

- *$\mathcal{W}(\mathbb{K}) = \prod_{\sigma\in G} V^\sigma(\mathbb{K})$; on note pr la projection $\mathcal{W}(\mathbb{K}) \to V(\mathbb{K})$.*
- *(Propriété universelle) Pour toute variété V' définie sur k et tout $\mathbb{K}$-morphisme $\varphi : V'(\mathbb{K}) \to V(\mathbb{K})$, il existe un unique k-morphisme $\psi : V'(k) \to \mathcal{W}(k)$ tel que le diagramme suivant*

commute :

$$\begin{array}{ccc} V'_{|k} & \xrightarrow{\varphi} & V_{|\mathbb{K}} \\ & \searrow^{\psi} & \uparrow{\scriptstyle pr} \\ & & \mathcal{W}_{|k} \end{array}$$

- *(Fonctorialité) Pour tout morphisme $\varphi : V \to V'$ de variétés algébriques défini sur $\mathbb{K}$, il existe un morphisme $\psi = W_{\mathbb{K}/k}(\varphi) : W_{\mathbb{K}/k}(V) \to W_{\mathbb{K}/k}(V')$ défini sur k tel que le diagramme suivant commute :*

$$\begin{array}{ccc} V_{|\mathbb{K}} & \xrightarrow{\varphi} & V'_{|\mathbb{K}} \\ {\scriptstyle pr}\uparrow & & \uparrow{\scriptstyle pr} \\ W_{\mathbb{K}/k}(V) & \xrightarrow{\psi} & W_{\mathbb{K}/k}(V') \end{array}$$

et tel que $W_{\mathbb{K}/k}(\varphi \circ \varphi') = W_{\mathbb{K}/k}(\varphi) \circ W_{\mathbb{K}/k}(\varphi')$.

Démonstration. Les deux premières propriétés se lisent directement sur la description galoisienne de $\mathcal{W}$. Pour le troisième point, on remarque que comme V' est définie sur k, $V'^{\sigma} = V'$ pour tout $\sigma \in G$; en particulier, l'application φ^{σ} obtenue en appliquant σ à tous ses coefficients est un $\mathbb{K}$-morphisme de V' dans V^{σ}. On pose donc $\psi(P) = (\varphi^{\sigma}(P))_{\sigma \in G}$; c'est un $\mathbb{K}$-morphisme de V' dans $\mathcal{W}$, qui commute avec tous les éléments de G. Par conséquent, $\psi : V' \to \mathcal{W}$ est en fait défini sur k. Pour l'unicité, on note que si $P \in V'(k)$, alors nécessairement $\psi(P) = (\sigma(\varphi(P)))_{\sigma \in G}$. Pour le dernier point, à chaque $\sigma \in G$, on peut associer le morphisme $\varphi^{\sigma} : V^{\sigma} \to V'^{\sigma}$; on pose alors $\psi((P_{\sigma})_{\sigma \in G}) = (\varphi^{\sigma}(P_{\sigma}))_{\sigma \in G}$, et on vérifie que ψ est bien défini sur k. □

L'avant-dernier point est en fait une propriété universelle pouvant servir à définir la restriction de Weil.

Dans le cas où V est une variété abélienne, l'identification naturelle des points de $V(\mathbb{K})$ et de $\mathcal{W}(k)$ permet de transporter la loi de groupe. Comme la loi de groupe sur V est définie par des applications rationnelles régulières, par fonctorialité $\mathcal{W}$ hérite d'une structure de groupe algébrique.

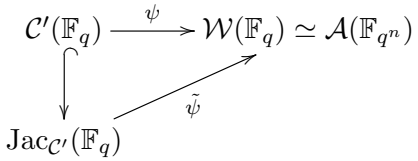
Propriété 6.1.2. *La restriction de Weil relativement à l'extension galoisienne $\mathbb{K}/k$ d'une variété abélienne définie sur $\mathbb{K}$ est une variété abélienne définie sur k.*

6.1.2 Transfert du logarithme

Soient $\mathcal{C}$ une courbe de genre $g \geq 1$ définie sur $\mathbb{F}_{q^n}$ $(n > 1)$ et $\mathcal{A} = \mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{q^n})$ sa jacobienne, qui est une variété abélienne de dimension g. On s'intéresse au problème du logarithme discret sur $\mathcal{A}$: soit $D_0 \in \mathcal{A}$ d'ordre un grand nombre premier, étant donné un diviseur $D \in \langle D_0 \rangle$, trouver d tel que $D = [d]D_0$. Dans le but d'une attaque, on souhaite transférer ce problème sur la jacobienne d'une courbe $\mathcal{C}'$ définie sur $\mathbb{F}_q$. On propose à cet effet deux méthodes : la première, basée sur une approche géométrique du problème, consiste à trouver une courbe de petit genre incluse dans la restriction de Weil de $\mathcal{A}$ puis à construire un morphisme explicite entre les variétés abéliennes données par la jacobienne de cette courbe et la restriction de $\mathcal{A}$; la deuxième utilise l'existence d'un recouvrement de la courbe $\mathcal{C}$ par une courbe $\mathcal{C}'$ (que l'on obtient avec la théorie des corps de fonctions) pour faire le transfert du DLP.

L'approche géométrique

On note $\mathcal{W} = W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\mathcal{A})$ la restriction de Weil de $\mathcal{A}$. Suivant [Fre98] et [GS99], dès que l'on se donne une courbe $\mathcal{C}'$ dans $\mathcal{W}$, ou plus généralement une application régulière $\psi : \mathcal{C}' \to \mathcal{W}$ où $\mathcal{C}'$ est une courbe définie sur $\mathbb{F}_q$, on peut transférer le DLP de la façon suivante. Quitte à composer ψ par la translation par un élément de $\mathcal{W}$, on peut supposer qu'il existe un point $P_\infty \in \mathcal{C}'(\mathbb{F}_q)$ tel que $\psi(P_\infty)$ est l'élément neutre de $\mathcal{W}$. On considère dans un premier temps le prolongement naturel de ψ au produit symétrique $\mathcal{C}'^g/\mathfrak{S}_g$, qui au g-uplet non ordonné $(P_1, \ldots, P_g)$ associe le point $\sum_{i=1}^g \psi(P_i)$, la somme étant prise pour la loi de groupe sur $\mathcal{W}$ induite par celle de $\mathcal{A}$. Le produit symétrique $\mathcal{C}'^g/\mathfrak{S}_g$ étant birationnel à la jacobienne de $\mathcal{C}'$, on en déduit une application régulière $\tilde{\psi} : \mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q) \to \mathcal{W}(\mathbb{F}_q)$ [Mil86c]. Or, une propriété fondamentale des variétés abéliennes est que toute application régulière entre deux variétés abéliennes envoyant l'élément neutre sur l'élément neutre est aussi un morphisme de groupes [Mil86b] ; par conséquent, $\tilde{\psi}$ est un morphisme de groupes de $\mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q)$ vers $\mathcal{W}(\mathbb{F}_q) \simeq \mathcal{A}(\mathbb{F}_{q^n})$. On peut alors tirer en arrière le DLP sur $\mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q)$.

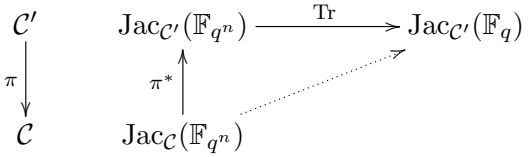


Cette approche présente deux difficultés majeures. Premièrement, pour que le DLP soit plus facile à attaquer dans $\mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q)$ que dans $\mathcal{A}(\mathbb{F}_{q^n})$, il faut que le genre de $\mathcal{C}'$ soit suffisamment petit. Une idée naturelle est alors de chercher pour $\mathcal{C}'$ une courbe dans $\mathcal{W}(\mathbb{F}_q)$ de petit degré ; une façon de faire est d'intersecter $\mathcal{W}(\mathbb{F}_q)$ par $n-1$ hyperplans (voir ci-dessous), mais même ainsi le genre obtenu est en général trop grand pour que le transfert soit intéressant. La deuxième difficulté est liée au calcul explicite d'antécédents de D et D_0 par $\tilde{\psi}$; cela revient à être capable de décomposer D et D_0 en une somme d'au plus g points de l'image $\psi(\mathcal{C}')$. Ce type de décompositions, qui sera étudié plus en détail dans le chapitre suivant, se ramène à la résolution de systèmes polynomiaux multivariés. Mais en pratique, ces systèmes deviennent rapidement trop compliqués pour pouvoir être résolus, ce qui limite la portée de cette approche.

On note que cette méthode s'applique à toute variété abélienne, et qu'il n'est pas nécessaire de se restreindre au cas où $\mathcal{A}$ est une variété jacobienne.

L'approche par recouvrement

Dans le cas où $\mathcal{C}$ est une courbe elliptique E, on a une identification naturelle entre la courbe et sa jacobienne. D'après la propriété 6.1.1, il est alors équivalent de se donner un $\mathbb{F}_q$-morphisme $\psi : \mathcal{C}'_{|\mathbb{F}_q} \to W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$ ou de se donner un $\mathbb{F}_{q^n}$-morphisme $\pi : \mathcal{C}'_{|\mathbb{F}_q}(\mathbb{F}_{q^n}) \to E(\mathbb{F}_{q^n})$; on appelle *recouvrement* de E un tel morphisme. Plus généralement, la donnée d'un recouvrement $\pi : \mathcal{C}'_{|\mathbb{F}_q} \to \mathcal{C}_{|\mathbb{F}_{q^n}}$ permet de transférer efficacement le DLP de $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{q^n})$ dans $\mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q)$. Pour cela, il suffit de composer le tiré en arrière $\pi^* : \mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{q^n}) \to \mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_{q^n})$ avec l'application trace $\mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_q} : \mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_{q^n}) \to \mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q)$, $D \mapsto \sum_{i=0}^{n-1} \sigma^i(D)$.



Cette application de transfert $\mathrm{Tr} \circ \pi^*$ appelée *conorme-norme* dans [GHS02b], est facilement calculable si le degré de π n'est pas trop gros. Pour que le transfert du DLP ait un intérêt, il faut d'une part que le genre de la courbe $\mathcal{C}'$ soit suffisamment petit, et d'autre part que l'application *conorme-norme* préserve un large sous-groupe d'ordre premier de $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{q^n})$, autrement dit que l'intersection de ce sous-groupe avec le noyau de l'application soit triviale (un simple argument de cardinalité impose alors $g(\mathcal{C}') \geq ng(\mathcal{C})$ dans la plupart des cas). Généralement, cette dernière condition ne pose pas de difficulté ; par contre, la construction d'une courbe $\mathcal{C}'$ avec de bonnes propriétés reste le principal obstacle.

6.2 Technique GHS

Dans [GHS02b], Gaudry, Hess et Smart analysent en détail le cas où $\mathcal{C}'$ est donnée par des sections hyperplanes, et proposent une technique utilisant la théorie des corps de fonctions pour expliciter ce recouvrement dans le cas d'une courbe elliptique E définie sur une extension de corps binaire. La caractéristique impaire a été ensuite reprise par Diem dans [Die03]. Certaines courbes étant plus vulnérables que d'autres à ces attaques, Galbraith, Hess et Smart [GHS02a, Hes04] proposent alors d'utiliser des marches d'isogénies pour transporter le DLP d'une courbe résistante à GHS à une courbe vulnérable, élargissant ainsi la portée de l'attaque GHS [MTW04].

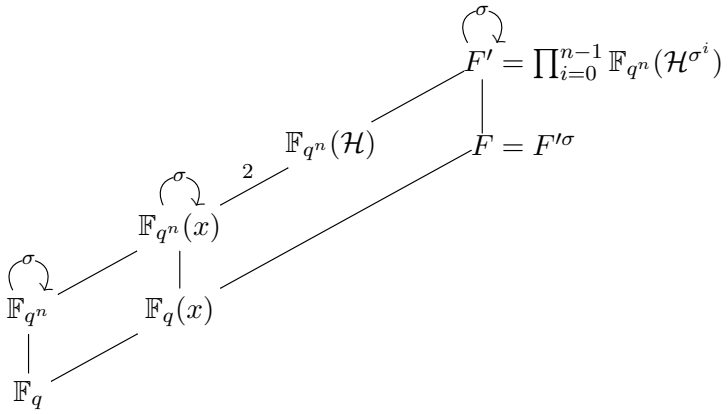
6.2.1 Cadre général

La technique GHS pour la recherche de recouvrement se formule plus simplement en termes de corps de fonctions, en utilisant la correspondance donnée en (5.1), section 5.1.1. Étant donnée une courbe $\mathcal{H}$ hyperelliptique définie sur $\mathbb{F}_{q^n}$, on cherche une extension algébrique $F' = \mathbb{F}_{q^n}(\mathcal{C}')$ de $\mathbb{F}_{q^n}(\mathcal{H})$ et un corps de fonction $F/\mathbb{F}_q (= \mathbb{F}_q(\mathcal{C}'))$ tels que $F' = \mathbb{F}_{q^n}F$. En particulier, F' doit être muni d'un automorphisme d'ordre n, prolongement naturel du Frobenius $\sigma_{\mathbb{F}_{q^n}/\mathbb{F}_q}$ encore noté σ, et F s'obtient comme l'ensemble F'^{σ} des éléments de F' invariants par σ.

Comme la courbe $\mathcal{H}$ est supposée définie sur $\mathbb{F}_{q^n}$, il n'existe a priori pas d'action du Frobenius sur $\mathbb{F}_{q^n}(\mathcal{H})$. Par contre, on peut utiliser l'application naturelle

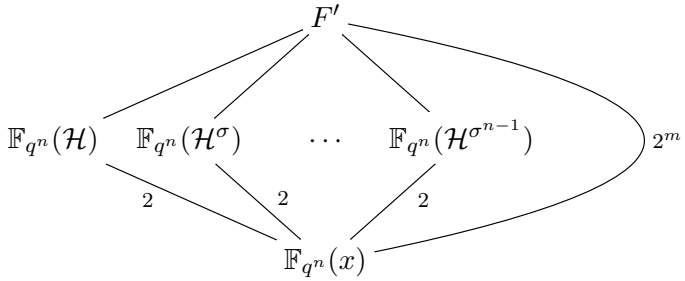
$$\sigma^* : \mathbb{F}_{q^n}(\mathcal{H}^{\sigma^{i+1}}) \to \mathbb{F}_{q^n}(\mathcal{H}^{\sigma^i})$$

pour construire F' comme compositum des corps de fonctions $\mathbb{F}_{q^n}(\mathcal{H}^{\sigma^i})$. Il faut pour cela plonger ces corps de fonctions dans un sur-corps commun ; le plus simple est de considérer les $\mathbb{F}_{q^n}(\mathcal{H}^{\sigma^i})$ comme des extensions quadratiques d'un même corps de fonctions rationnelles $\mathbb{F}_{q^n}(x)$ (ce qui est possible puisque $\mathcal{H}$ est hyperelliptique) et de prendre comme sur-corps une clôture algébrique $\overline{\mathbb{F}_{q^n}(x)}$ contenant $\mathbb{F}_{q^n}(\mathcal{H})$ et $\overline{\mathbb{F}_q}(x)$. On note que comme l'extension $\mathbb{F}_{q^n}(\mathcal{H}^{\sigma^i})/\mathbb{F}_{q^n}(x)$ est galoisienne (car quadratique), il n'y a pas d'ambiguïté sur l'image du plongement dans $\overline{\mathbb{F}_{q^n}(x)}$. On peut alors définir F' comme le compositum $\prod_{i=0}^{n-1} \mathbb{F}_{q^n}(\mathcal{H}^{\sigma^i})$ dans cette clôture. Il reste à vérifier que l'automorphisme de Frobenius peut bien s'étendre en un automorphisme d'ordre n de F' vérifiant $\sigma(x) = x$, et à déterminer le genre et une expression de la courbe $\mathcal{C}'$ ainsi construite.



On remarque que le recouvrement obtenu par cette technique dépend fortement du choix de l'extension $\mathbb{F}_{q^n}(\mathcal{H})/\mathbb{F}_{q^n}(x)$, donc d'un choix d'équation pour $\mathcal{H}$. Si le genre de $\mathcal{H}$ est plus grand que 1, il est connu que $\mathbb{F}_{q^n}(\mathcal{H})$ contient un seul sous-corps rationnel d'indice 2, cependant le choix du générateur x (tel que $\sigma(x) = x$) reste important. Celui-ci va notamment influer sur l'expression des équations de la courbe $\mathcal{C}'$ de corps de fonctions F et sur son genre.

Comme chacune des extensions intermédiaires $\mathbb{F}_{q^n}(\mathcal{H}^{\sigma^i})/\mathbb{F}_{q^n}(x)$ est quadratique, le degré de l'extension $F'/\mathbb{F}_{q^n}(x)$ est nécessairement $[F' : \mathbb{F}_{q^n}(x)] = 2^m$ où m est un entier inférieur ou égal à n. On verra que le genre de la courbe $\mathcal{C}'$ dépend essentiellement de la valeur de m, qui est parfois appelé "nombre magique" pour cette raison.



On considère dans la suite une courbe hyperelliptique $\mathcal{H}$ de genre g, donnée par une équation générale de la forme

$$y^2 + h_0(x)y = h_1(x) \text{ avec } h_0, h_1 \in \mathbb{F}_{q^n}[x] \text{ et } h_0 = 0 \text{ en caractéristique impaire.} \tag{6.1}$$

Parallèlement au nombre magique m, on introduit aussi l'entier $\overline{m}$ tel que $[\overline{\mathbb{F}_q}F' : \overline{\mathbb{F}_q}(x)] = 2^{\overline{m}}$. L'extension F' du corps de fonction de $\mathbb{F}_{q^n}(\mathcal{H})$ s'obtient en considérant $F' = \mathbb{F}_{q^n}(x, y_0, \ldots, y_{n-1})$, où $y_0, \ldots, y_{n-1}$ vérifient les équations suivantes

$$\begin{cases} y_0^2 + h_0 y_0 = h_1 \\ \quad \vdots \\ y_{n-1}^2 + h_0^{\sigma^{n-1}} y_{n-1} = h_1^{\sigma^{n-1}} \end{cases} \tag{6.2}$$

Lemme 6.2.1. *Si m est l'entier tel que $[F' : \mathbb{F}_{q^n}(x)] = 2^m$, alors $F' = \mathbb{F}_{q^n}(x, y_0, \ldots, y_{m-1})$; de la même façon, $\overline{\mathbb{F}_q}F' = \overline{\mathbb{F}_q}(x, y_0, \ldots, y_{\overline{m}-1})$.*

Démonstration. On pose $F'_i = \mathbb{F}_{q^n}(x, y_0, \ldots, y_{i-1})$. D'une part $F'_{i+1} = F'_i(y_i)$ et donc $[F'_{i+1} : F'_i] = 1$ ou 2. D'autre part, s'il existe un entier $i < n$ tel que $F'_{i+1} = F'_i$, alors $F'_j = F'_i$ pour tout $i \leq j \leq n$; en effet par récurrence, si $F'_j = F'_i$ alors $y_{j-1} \in \mathbb{F}_{q^n}(x, y_0, \ldots, y_{i-1})$ et en appliquant σ, on obtient $y_j \in \mathbb{F}_{q^n}(x, y_1, \ldots, y_i) \subset F'_{i+1} = F'_i$. Finalement, comme $[F'_n : \mathbb{F}_{q^n}(x)] = 2^m$, on a nécessairement $F'_m = F'$. La démonstration pour $\overline{m}$ est similaire. □

Dans l'attaque GHS, pour que la courbe $\mathcal{C}'$ soit bien définie sur $\mathbb{F}_q$, il faut s'assurer que le corps des constantes de F' est $\mathbb{F}_{q^n}$, ce qui sera vérifié lorsque $m = \overline{m}$:

Lemme 6.2.2. *Le corps des constantes de F' est égal à $\mathbb{F}_{q^n}$ si $m = \overline{m}$ et à $\mathbb{F}_{q^{2n}}$ sinon.*

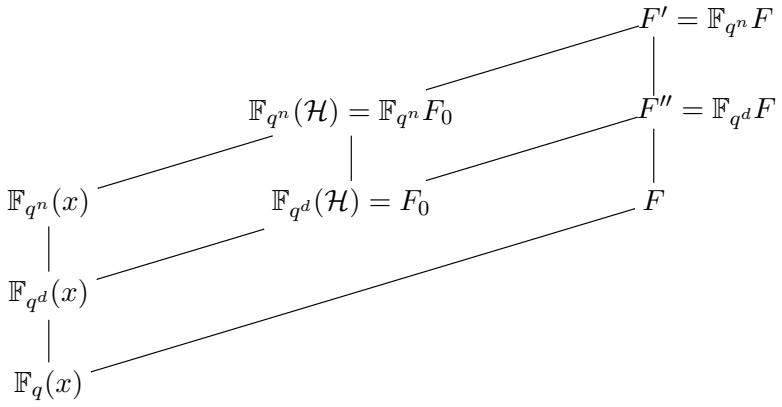
Démonstration. Soit $\widetilde{\mathbb{F}_{q^n}}$ le corps des constantes de F'. On a d'une part $\mathrm{Gal}(\widetilde{\mathbb{F}_{q^n}(x)}/\mathbb{F}_{q^n}(x)) \simeq \mathrm{Gal}(\widetilde{\mathbb{F}_{q^n}}/\mathbb{F}_{q^n})$ qui est cyclique. D'autre part, l'application $\mathrm{Gal}(F'/\mathbb{F}_{q^n}(x)) \to \mathrm{Gal}(\widetilde{\mathbb{F}_{q^n}(x)}/\mathbb{F}_{q^n}(x))$ étant surjective, $\mathrm{Gal}(\widetilde{\mathbb{F}_{q^n}(x)}/\mathbb{F}_{q^n}(x))$ est un quotient de $\mathrm{Gal}(F'/\mathbb{F}_{q^n}(x))$ qui par construction est isomorphe à $(\mathbb{Z}/2\mathbb{Z})^m$. Or les seuls quotients cycliques de $(\mathbb{Z}/2\mathbb{Z})^m$ sont $\{1\}$ et $\mathbb{Z}/2\mathbb{Z}$, on a donc soit $\widetilde{\mathbb{F}_{q^n}} = \mathbb{F}_{q^n}$, soit $\widetilde{\mathbb{F}_{q^n}} = \mathbb{F}_{q^{2n}}$.
Par ailleurs, on vérifie aisément que $[F' : \mathbb{F}_{q^n}(x)] = [\overline{\mathbb{F}_q}F' : \overline{\mathbb{F}_q}(x)]$ si et seulement si le corps des constantes de F' est bien $\mathbb{F}_{q^n}$, ce qui donne bien l'équivalence cherchée. □

On prolonge ensuite σ défini sur $\mathbb{F}_{q^n}(x)$ en un automorphisme d'ordre n défini sur $F' = \mathbb{F}_{q^n}(x, y_0, \ldots, y_{n-1})$, afin d'obtenir $F = F'^{\sigma}$. Lorsque $m = n$, ceci ne pose pas de difficulté : il suffit de poser pour $0 \leq i \leq n-2$, $\sigma(y_i) = y_{i+1}$ et $\sigma(y_{n-1}) = y_0$. Lorsque n est impair, on peut également faire ce prolongement en introduisant la suite exacte suivante :

$$0 \longrightarrow \mathrm{Gal}(F'/\mathbb{F}_{q^n}(x)) \simeq (\mathbb{Z}/2\mathbb{Z})^m \xrightarrow{\imath} \mathrm{Gal}(F'/\mathbb{F}_q(x)) \xrightarrow{pr} \mathrm{Gal}(\mathbb{F}_{q^n}(x)/\mathbb{F}_q(x)) \simeq \mathbb{Z}/n\mathbb{Z} \longrightarrow 0.$$

On considère $\tau \in \mathrm{Gal}(F'/\mathbb{F}_q(x))$ tel que $pr(\tau) = \sigma$; si τ est d'ordre n, on prolonge σ par τ. Sinon, τ^n est dans l'image de $\imath$ et d'ordre 2, donc $(\tau^n)^n = \tau^n$ (puisque n est impair). On prolonge alors σ par $\tau \circ \tau^{-n}$, et on vérifie sans difficulté que ce prolongement est bien un élément d'ordre n de $\mathrm{Gal}(F'/\mathbb{F}_q(x))$.

Si la courbe $\mathcal{H}$ est définie sur un sous-corps de $\mathbb{F}_{q^n}$, le transfert du logarithme discret peut échouer. Plus précisément, supposons qu'il existe un corps de fonctions F_0, extension quadratique de $\mathbb{F}_{q^d}(x)$ où $d|n$, dont le corps de constantes est $\mathbb{F}_{q^d}$ et tel que $\mathbb{F}_{q^n}(\mathcal{H}) = \mathbb{F}_{q^n}F_0$. Alors $\mathcal{H}$ est en fait définie sur $\mathbb{F}_{q^d}$ et $F_0 = \mathbb{F}_{q^d}(\mathcal{H})$.



Dans ce cas, l'application de transfert conorme-norme de $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$ dans $\mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q)$ se factorise via $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^d})$, et contient dans son noyau le sous-groupe de trace zéro de $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$, c'est-à-dire le noyau de l'homomorphisme $\mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_{q^d}} : \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n}) \to \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^d})$.

$$\begin{array}{ccccc}
\mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_{q^n}) & \xrightarrow{\mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_{q^d}}} & \mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_{q^d}) & \xrightarrow{\mathrm{Tr}_{\mathbb{F}_{q^d}/\mathbb{F}_q}} & \mathrm{Jac}_{\mathcal{C}'}(\mathbb{F}_q) \\
\uparrow{\scriptstyle \pi^*} & & \uparrow{\scriptstyle \pi^*} & & \\
\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n}) & \xrightarrow{\mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_{q^d}}} & \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^d}) & &
\end{array}$$

Dans le cas contraire, le noyau de l'application de transfert ne comprend que des éléments d'ordre une puissance de 2, et donc ne peut pas contenir un sous-groupe d'ordre premier grand.

Proposition 6.2.3 ([Die03, Hes04])**.** *On suppose que $m = \overline{m}$ et que σ se prolonge en un automorphisme d'ordre n de F'. Si pour tout $1 \leq i \leq n-1$, $\sigma^i(\mathbb{F}_{q^n}(\mathcal{H})) \neq \mathbb{F}_{q^n}(\mathcal{H})$, alors le noyau de l'application conorme-norme est inclus dans $Jac_{\mathbb{F}_{q^n}}(\mathcal{H})[2^{m-1}]$.*

Il est à noter toutefois que même si $\mathcal{H}$ est définie sur un sous-corps, il est possible de faire fonctionner l'attaque GHS en choisissant pour $\mathcal{H}$ une équation à coefficients dans $\mathbb{F}_{q^n}$ et non dans un sous-corps.

6.2.2 Caractéristique 2

On commence par donner un encadrement des valeurs possibles du genre de la courbe $\mathcal{C}'$ obtenue par l'attaque GHS, montrant que ce genre est nécessairement exponentiel en le "nombre magique" m :

Proposition 6.2.4 ([Hes03, Th. 2])**.** *On suppose que $m \geq 2$ et que F' admet pour corps de constantes $\mathbb{F}_{q^n}$. Alors*

$$2^{m-2}g(\mathcal{H}) + 1 \leq g(F') \leq n(2^m - 1)g(\mathcal{H}).$$

Afin d'estimer plus précisément le genre de la courbe $\mathcal{C}'$ obtenue, on va expliciter m (resp. $\overline{m}$). On utilise à cet effet la théorie des extensions d'Artin-Schreier. Quitte à faire le changement de variables $y \mapsto y/h_0(x)$ dans (6.1), on se ramène à une équation pour $\mathcal{H}$ de la forme $y^2 + y = h(x)$, où $h \in \mathbb{F}_{q^n}(x)$.

On note $\wp : f \mapsto f^2 + f$ l'opérateur d'Artin-Schreier agissant sur le corps de fonctions $\mathbb{F}_{q^n}(x)$, resp. $\overline{\mathbb{F}_q}(x)$, et on considère le $\mathbb{F}_2$-espace vectoriel défini par $\mathcal{P} = \mathbb{F}_{q^n}(x)/\wp(\mathbb{F}_{q^n}(x))$ obtenu en quotientant le corps des fonctions rationnelles par l'image de $\wp$, resp. $\overline{\mathcal{P}} = \overline{\mathbb{F}_q}(x)/\wp(\overline{\mathbb{F}_q}(x))$. La théorie d'Artin-Schreier permet de montrer (voir [GHS02b, Lemme 6]) que m est la dimension du $\mathbb{F}_2$-sous-espace vectoriel $U \subset \mathcal{P}$ engendré par les classes dans $\mathcal{P}$ de $h, h^\sigma, \ldots, h^{\sigma^{n-1}}$, resp. $\overline{m}$ est celle du sous-espace $\overline{U} \subset \overline{\mathcal{P}}$ correspondant. De plus, il existe une correspondance entre les extensions intermédiaires $\mathbb{F}_{q^n}(x) \subset H \subset F'$ avec $[H : \mathbb{F}_{q^n}(x)] = 2^d$ et les $\mathbb{F}_2$-sous-espaces vectoriels U' de U de dimension d, donnée par $H = \mathbb{F}_{q^n}(x)(\wp^{-1}(V'))$ où V' est un ensemble de représentants de U' dans $\mathbb{F}_{q^n}(x)$.

Le $\mathbb{F}_2$-espace vectoriel $\mathcal{P}$, resp. $\overline{\mathcal{P}}$, possède une structure supplémentaire de $\mathbb{F}_2[t]$-module, où l'action d'un polynôme de $\mathbb{F}_2[t]$ sur un élément de $\mathcal{P}$, resp. $\overline{\mathcal{P}}$, provient de l'action sur $\mathbb{F}_{q^n}(x)$

donnée par $(\sum a_i t^i) \cdot f(x) = \sum a_i f^{\sigma_i}(x)$. Il est alors naturel de s'intéresser à l'idéal

$$\mathcal{I}_h = \{P \in \mathbb{F}_2[t] : P \cdot h(x) \in \wp(\mathbb{F}_{q^n}(x))\}.$$

Le polynôme minimal M_h de cet idéal est de degré exactement m et vérifie $M_h | (t^n + 1)$; de plus, on a un isomorphisme entre U et $\mathbb{F}_2[t]/\langle M_h \rangle$. Similairement, on trouve que $\overline{m}$ est le degré du polynôme $\overline{M}_h$ minimal tel que $P \cdot h(x) \in \wp(\overline{\mathbb{F}_q}(x))$, et $\overline{M}_h$ divise nécessairement M_h.

Proposition 6.2.5 ([GS91]). *Sous l'hypothèse que F' admet pour corps de constantes $\mathbb{F}_{q^n}$, on a*

$$g(F') = \sum_{i=1}^{2^m - 1} g(F_i),$$

où les F_i sont les corps intermédiaires $\mathbb{F}_{q^n}(x) \subset F_i \subset F'$ tels que $[F_i : \mathbb{F}_{q^n}(x)] = 2$.

Par la théorie d'Artin-Schreier, les F_i sont en correspondance avec les éléments non nuls de $U \simeq \mathbb{F}_2[t]/\langle M_h \rangle$, et sont tous de la forme $\mathbb{F}_{q^n}(x, z)$ où $z^2 + z = P \cdot h$ et $P \in \mathbb{F}_2[t]/\langle M_h \rangle$. Connaissant h et m, on peut alors en déduire le genre de F'.

Le fait que M_h divise $t^n + 1$ limite les valeurs possibles de m, et permet dans certains cas d'obtenir une borne inférieure sur le genre de F' ne dépendant que de n. Par exemple, pour n premier impair, en introduisant la décomposition en facteurs irréductibles $\prod_i \Phi_{n,i}$ du n-ième polynôme cyclotomique Φ_n, on a

$$t^n + 1 = (t+1)\Phi_n(t) = (t+1) \prod_i \Phi_{n,i}(t).$$

Soit $d_{n,i}$ le degré de $\Phi_{n,i}$. Si ζ est une racine de $\Phi_{n,i}$, elle engendre le corps $\mathbb{F}_{2^{d_{n,i}}}$ sur $\mathbb{F}_2$ et $d_{n,i}$ est donc le plus petit entier d tel que $\sigma_{2^d}(\zeta) = \zeta^{2^d} = \zeta$. Comme ζ est une racine primitive n-ième de l'unité, on a $\zeta^{2^d} = \zeta$ si et seulement si $2^d = 1 \bmod n$; on obtient ainsi que le degré de chacun des $\Phi_{n,i}$ est l'ordre de 2 dans $(\mathbb{Z}/n\mathbb{Z})^*$, noté $\varphi_2(n)$. Si $\mathcal{H}$ n'est pas définie sur $\mathbb{F}_q$, ceci implique que m est de la forme $k\varphi_2(n)$ ou $k\varphi_2(n) + 1$, avec $k \in \mathbb{N}^*$. Or les valeurs de n pour lesquelles $\varphi_2(n)$ est petit sont relativement rares (tels les nombres de Mersenne ou les nombres de Fermat qui donnent la valeur quasi-optimale pour m). En particulier, il n'y a qu'un nombre restreint de valeurs de n pour lesquelles l'attaque GHS peut être efficace.

Il reste enfin à déterminer l'équation de $\mathcal{C}'$, autrement dit $\tilde{y}$ tel que $F = \mathbb{F}_q(x, \tilde{y})$. Ceci impose que $F' = \mathbb{F}_{q^n}(x, y_0, \ldots, y_{m-1}) = \mathbb{F}_{q^n}(x, \tilde{y})$ et donc que le polynôme minimal de $\tilde{y}$ sur $\mathbb{F}_{q^n}(x)$ (qui est aussi le polynôme minimal de $\tilde{y}$ sur $\mathbb{F}_q(x)$) soit de degré 2^m. Une possibilité est de prendre alors pour $\tilde{y}$:

$$\tilde{y} = \sum_{i=0}^{n-1} \sigma^i(\theta) y_i, \text{ avec } \{\theta, \sigma(\theta), \ldots, \sigma^{n-1}(\theta)\} \text{ base normale de } \mathbb{F}_{q^n}.$$

On vérifie alors sans difficulté que $\sigma(\tilde{y}) = \tilde{y}$ (donc $\tilde{y} \in F$). De plus $\#\{\tau(\tilde{y}) : \tau \in \mathrm{Gal}(F'/\mathbb{F}_{q^n}(x))\} = 2^m = \#\mathrm{Gal}(F'/\mathbb{F}_{q^n}(x))$; en effet $\tau(\tilde{y}) = \sum_{i=0}^{n-1} \sigma^i(\theta)\tau(y_i) = \tilde{y} + \sum_{i=0}^{n-1} \sigma^i(\theta)(\tau(y_i) + y_i)$ où $\tau(y_i) + y_i \in \mathbb{F}_2$, et $\tau(\tilde{y}) = \tilde{y}$ si et seulement si $\tau(y_i) = y_i$ pour tout i, i.e. si et seulement si $\tau = Id$. Le polynôme minimal de $\tilde{y}$ sur $\mathbb{F}_{q^n}(x)$ est ainsi égal à $\prod_{\tau \in \mathrm{Gal}(F'/\mathbb{F}_{q^n}(x))}(T - \tau(\tilde{y}))$, de degré 2^m, et fournit une équation de la courbe $\mathcal{C}'$ définie sur $\mathbb{F}_q$. Pour trouver l'application de recouvrement $\pi : \mathcal{C}'(\mathbb{F}_{q^n}) \to E(\mathbb{F}_{q^n})$, il faut exprimer $y = y_0$ en fonction de $\tilde{y}$; cela peut se faire en calculant la base de Gröbner pour un ordre d'élimination de l'idéal de $\mathbb{F}_{q^n}(x)[y_0, \ldots, y_n, \tilde{y}]$ contenant les équations des courbes E^{σ^i}, l'équation de $\mathcal{C}'$, ainsi que l'expression de $\tilde{y}$.

Cas des courbes elliptiques

Soit E une courbe elliptique ordinaire définie sur $\mathbb{F}_{q^n}$ (où $q = 2^d$), d'équation $y^2 + xy = x^3 + ax^2 + b$. En remplaçant y par $yx + \sqrt{b}$ puis en divisant par x^2, on obtient une équation de la forme Artin-Schreier $y^2+y = x+a+\sqrt{b}/x$. Une forme plus générale est obtenue par un changement de variable $x \mapsto \lambda x$, $\lambda \in \mathbb{F}_{q^n}$, on a alors

$$E: \ y^2 + y = h(x) \quad \text{où} \quad h(x) = \beta x + \alpha + \gamma/x. \tag{6.3}$$

Un argument simple sur le degré montre que si $P \in \mathbb{F}_2[t]$ est tel que $P \cdot h = \wp(f)$ pour un certain $f \in \mathbb{F}_{q^n}(x)$ ou $\overline{\mathbb{F}_q}(x)$, alors nécessairement $P \cdot \beta = P \cdot \gamma = 0$. Réciproquement, si $P \cdot \beta = P \cdot \gamma = 0$, alors $P \cdot h = P \cdot \alpha \in \wp(\overline{\mathbb{F}_q}(x))$ car $P \cdot \alpha \in \mathbb{F}_{q^n} \subset \wp(\overline{\mathbb{F}_q})$. On introduit alors les polynômes minimaux M_γ et M_β des idéaux $\{P : P \cdot \gamma = 0\}$ et $\{P : P \cdot \beta = 0\}$ de $\mathbb{F}_2[t]$; on vient de montrer que $\overline{M}_h = ppcm(M_\beta, M_\gamma)$. Pour déterminer M_h, on note que si $\mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_2}(\alpha) = 0$, alors $P \cdot \alpha$ est toujours dans $\wp(\mathbb{F}_{q^n})$; si $\mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_2}(\alpha) = 1$, alors $P \cdot \alpha \in \wp(\mathbb{F}_{q^n})$ si et seulement si $(t+1)|P$. Par conséquent,

$$M_h = \begin{cases} ppcm(M_\beta, M_\gamma) & \text{si } \mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_2}(\alpha) = 0, \\ ppcm(M_\beta, M_\gamma, t+1) & \text{sinon.} \end{cases}$$

En particulier, on a $m = \overline{m}$ si et seulement si $\mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_2}(\alpha) = 0$ ou $(t+1)|\overline{M}_h$. Si n est impair, ceci peut se reformuler en

$$m = \overline{m} \Leftrightarrow \mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_2}(\alpha) = 0 \text{ ou } \mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\beta) \neq 0 \text{ ou } \mathrm{Tr}_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\gamma) \neq 0.$$

Dans la suite, on suppose que $m = \overline{m}$. Pour calculer le genre, on applique la Proposition 6.2.5. Chacun des corps F_i apparaissant dans la formule a une équation de la forme $z^2 + z = (P_i \cdot \beta)x + P_i \cdot \alpha + (P_i \cdot \gamma)/x$ où P_i est de degré inférieur à celui M_h. Le genre de F_i vaut 1 sauf si $P_i \cdot \beta = 0$ ou $P_i \cdot \gamma = 0$, i.e. si $M_\beta | P_i$ ou $M_\gamma | P_i$ (comme $M_h = \overline{M}_h = ppcm(M_\beta, M_\gamma)$, on ne peut pas avoir les deux conditions simultanément). On trouve alors que le genre de F' est

$$g(F') = 2^m - 1 - (2^{m-\deg M_\beta} - 1 + 2^{m-\deg M_\gamma} - 1) = 2^m - 2^{m-\deg M_\beta} - 2^{m-\deg M_\gamma} + 1. \tag{6.4}$$

Lorsque $\gamma \in \mathbb{F}_q$ ou $\beta \in \mathbb{F}_q$, il est facile de démontrer que $\mathcal{C}'$ est un recouvrement hyperelliptique. Supposons par exemple que $\gamma \in \mathbb{F}_q$ (le raisonnement est identique lorsque $\beta \in \mathbb{F}_q$). Comme $F' = \mathbb{F}_{q^n}(x, y_0, \ldots, y_{m-1})$ où les fonctions y_i vérifient les équations suivantes

$$\begin{cases} y_0^2 + y_0 = \beta x + \alpha + \gamma/x \\ \qquad\qquad \vdots \\ y_{m-1}^2 + y_{m-1} = \sigma^{m-1}(\beta)x + \sigma^{m-1}(\alpha) + \gamma/x, \end{cases}$$

en posant $z_i = y_i + y_0$, on obtient le système d'équations équivalent

$$\begin{cases} y_0^2 + y_0 = \beta x + \alpha + \gamma/x \\ z_1^2 + z_1 = (\beta + \sigma(\beta))x + \alpha + \sigma(\alpha) \\ \qquad\qquad \vdots \\ z_{m-1}^2 + z_{m-1} = (\beta + \sigma^{m-1}(\beta))x + \alpha + \sigma^{m-1}(\alpha), \end{cases}$$

et $F' = \mathbb{F}_{q^n}(x, z_1, \ldots, z_{m-1})(y_0)$. Par conséquent, F' est une extension de degré 2 du corps de fonctions $\mathbb{F}_{q^n}(x, z_1, \ldots, z_{m-1})$ de genre 0 ; F' est donc un corps hyperelliptique.

Ainsi en caractéristique 2, on peut toujours se ramener, quitte à faire un changement de variable, à un recouvrement hyperelliptique. Mais bien entendu, ce recouvrement ne sera pas nécessairement celui pour lequel la valeur du nombre magique m sera minimal.

Exemple 6.2.6. *On considère la courbe elliptique définie sur* $\mathbb{F}_{2^7} \simeq \mathbb{F}_2(\theta)$ *où* $\theta^7+\theta^6+1=0$ *(de telle sorte que* θ *engendre une base normale de* $\mathbb{F}_{2^7}$*) d'équation*

$$E:\ y^2+xy=x^3+(\theta^2+1).$$

En remplaçant y *par* $yx+\sqrt{\theta^2+1}$*, on obtient l'équation sous forme d'Artin-Schreier*

$$E:\ y^2+y=x+(\theta+1)/x.$$

La décomposition en facteurs irréductibles de X^7+1 *est* $(X+1)(X^3+X^2+1)(X^3+X+1)$*, et l'ordre de* 2 *modulo* 7 *est* $\varphi_2(7)=3$*, donc les valeurs possibles pour* m *sont* $3,4,6$ *ou* 7*. Les polynômes minimaux de* $\beta=1$ *et* $\gamma=\theta+1$ *sont alors* $M_\beta=X+1$ *et* $M_\gamma=\sum_{i=0}^{6}X^i$*, en particulier* $M_h=X^7+1$*. Par conséquent le genre du recouvrement obtenu par la méthode GHS est* $g=2^7-2^6-2+1=63$*. En posant* $\tilde{y}=\theta y_0+\theta^2y_1+\cdots+\theta^{64}y_6$*, où* y_i *vérifie l'équation* $y_i^2+y_i=x+(\theta^{2^i}+1)/x$*, on trouve qu'une équation de* $\mathcal{C}'$*, donnée par le polynôme minimal* $\prod_{\tau\in Gal(F'/\mathbb{F}_{2^7}(x))}(T-\tau(\tilde{y}))$*, est*

$$x^{16}(\tilde{y}^{128}+\tilde{y})=x^{80}+x^{48}+x^{32}+x^{24}+x^{20}+x^{18}+x^{17}+x^8+1.$$

Cependant cette équation masque le fait que la courbe $\mathcal{C}'$ *est hyperelliptique puisque* $\beta\in\mathbb{F}_2$*. Le sous-corps* L' *de genre* 0 *et d'indice* 2 *de* F' *est* $\mathbb{F}_{2^7}(x,z_1,\ldots,z_6)$ *où* $z_i=y_i+y_0$ *vérifie l'équation* $z_i^2+z_i=(\theta^{2^i}+\theta)/x$*. On montre que* $L'=\mathbb{F}_{2^7}(x,w)=\mathbb{F}_{2^7}(w)$ *où* w *est une racine du polynôme* $X^{64}+X^{32}+\cdots+X^2+X+1/x\in\mathbb{F}_{2^7}(x)[X]$*, et que* $F'=\mathbb{F}_{2^7}(w,\bar{y})$*, où* $\bar{y}=y_0+\cdots+y_6$ *vérifie* $\bar{y}^2+\bar{y}=x$ *; une équation de* $\mathcal{C}'$ *est donc*

$$\bar{y}^2+\bar{y}=\left(\sum_{i=0}^{6}w^{2^i}\right)^{-1},$$

qui se ramène à un changement de variables près à l'équation équivalente

$$\bar{y}^2+(\sum_{i=0}^{6}w^{2^i})\bar{y}=\sum_{i=0}^{6}w^{2^i}.$$

En remplaçant x *par* $(\theta^5+\theta^4)x$*, on obtient une nouvelle équation d'Artin-Schreier pour* E *:*

$$E:\ y^2+y=(\theta^5+\theta^4)x+(\theta^3+\theta^2)/x.$$

Pour cette équation, les polynômes minimaux de $\beta=\theta^5+\theta^4$ *et* $\gamma=\theta^3+\theta^2$ *sont* $M_\beta=M_\gamma=X^3+X+1$*, en particulier* $\overline{M_h}=M_h=X^3+X+1$ *(puisque* $\alpha=0$*) et* $\overline{m}=m=3$*. Pour trouver l'équation de la courbe* $\mathcal{C}'$ *définie sur* $\mathbb{F}_2$ *de genre* $2^3-2^0-2^0+1=7$ *qui recouvre* E*, on considère encore* $\tilde{y}=\theta y_0+\theta^2y_1+\cdots+\theta^{64}y_6$ *; la courbe* $\mathcal{C}'$ *(qui n'est pas hyperelliptique) est alors donnée par*

$$x^2(\tilde{y}^8+\tilde{y}^4+\tilde{y})=x^6+1.$$

Sur cet exemple, on voit que le choix de l'équation de E *a des grandes répercussions sur le type de revêtement obtenu.*

6.2.3 Caractéristique impaire

On peut faire la même analyse en caractéristique impaire à quelques petites modifications près. On considère après changement de variable la courbe $\mathcal{H}$ d'équation :

$$\mathcal{H}: \ y^2 = h(x) \quad \text{avec } h \in \mathbb{F}_{q^n}[x]. \tag{6.5}$$

Pour calculer le genre de $\mathcal{C}'$, on a un résultat plus précis qui découle de la formule d'Hurwitz [Sti93, III.5.6] (et qui s'applique aussi aux extensions intermédiaires de F' sur $\mathbb{F}_{q^n}(x)$).

Proposition 6.2.7 ([Die03]). *En caractéristique impaire, le genre de la courbe $\mathcal{C}'$ est donné par*

$$g(F') = 2^{\overline{m}-2}(r-4)+1 \tag{6.6}$$

où r est le nombre de points de ramification de $\overline{\mathbb{F}_q}F'$ sur $\overline{\mathbb{F}_q}(x)$, autrement dit r est la cardinalité de l'ensemble des abscisses (dans $\mathbb{P}^1(\overline{\mathbb{F}_q})$) des points de Weierstrass sur $\overline{\mathbb{F}_q}$ des courbes $\mathcal{H}^{\sigma^i}$ définies par les équations $y^2 = h^{\sigma^i}(x)$.

On peut déterminer comme précédemment les valeurs possibles de m en faisant à appel à la théorie de Kummer (au lieu de celle d'Artin-Schreier). En particulier, on définit $\mathcal{P}$, resp. $\overline{\mathcal{P}}$, comme le $\mathbb{F}_2$-espace vectoriel obtenu en prenant $\mathbb{F}_{q^n}(x)^*/(\mathbb{F}_{q^n}(x)^*)^2$, resp. $\overline{\mathbb{F}_q}(x)^*/(\overline{\mathbb{F}_q}(x)^*)^2$, et on a $m = \dim U$ où U est engendré par les classes de $h, h^\sigma, \ldots, h^{\sigma^{n-1}}$ dans $\mathcal{P}$, resp. $\overline{m} = \dim \overline{U}$. L'action naturelle de $\mathbb{F}_2[t]$ sur $\mathcal{P}$ et $\overline{\mathcal{P}}$ est alors induite par l'action :

$$\begin{array}{rcl} \mathbb{Z}[t] \times \overline{\mathbb{F}_q}(x) & \to & \overline{\mathbb{F}_q}(x) \\ \left(\sum a_i t^i, f(x)\right) & \mapsto & (\sum a_i t^i) \cdot f(x) = \prod (f^{\sigma_i}(x))^{a_i}. \end{array}$$

Comme précédemment, on a une correspondance naturelle entre les extensions intermédiaires $\mathbb{F}_{q^n}(x) \subset H \subset F'$ avec $[H : \mathbb{F}_{q^n}(x)] = 2^d$ et les $\mathbb{F}_2$-sous-espaces vectoriels U' de U de dimension d, donnée par $H = \mathbb{F}_{q^n}(x)(\sqrt{V'})$ où V' est un ensemble de représentants de U' dans $\mathbb{F}_{q^n}(x)$. On définit encore M_h, resp. $\overline{M}_h$, comme le polynôme minimal de l'idéal $\mathcal{I}_h = \{P \in \mathbb{F}_2[t] : P \cdot h \in (\mathbb{F}_{q^n}(x)^*)^2\}$, resp. $P \cdot h \in (\overline{\mathbb{F}_q}(x)^*)^2$; son degré est exactement m, resp. $\overline{m}$. Comme $M_h | (t^n+1)$, les considérations précédentes sur les valeurs possibles de m en fonction de n restent valables.

Lorsque l'on connaît les racines dans $\overline{\mathbb{F}_q}$ de $h \in \mathbb{F}_{q^n}[x]$, il est assez facile de déterminer $\overline{M}_h$: en effet, $P \cdot h$ est un carré dans $\overline{\mathbb{F}_q}(x)$ si et seulement si toutes ses racines (qui sont des itérés par σ des racines de h) sont de multiplicité paire. Pour que $P \cdot h$ soit un carré dans $\mathbb{F}_{q^n}(x)$, il faut en plus que son coefficient dominant soit un carré dans $\mathbb{F}_{q^n}$. En particulier, si h est unitaire alors on a nécessairement $m = \overline{m}$. Plus précisément, si on note c le coefficient dominant de h, alors le coefficient dominant de $P \cdot h$ est $P \cdot c$. Si $c \in (\mathbb{F}_{q^n})^2$, alors $P \cdot c$ est un carré pour tout P ; sinon, $P \cdot c \in (\mathbb{F}_{q^n})^2$ si et seulement si $(t+1)|P$. Par conséquent,

$$M_h = \begin{cases} \overline{M}_h & \text{si } c \in (\mathbb{F}_{q^n})^2 \\ ppcm(\overline{M}_h, t+1) & \text{sinon.} \end{cases}$$

Contrairement au cas de la caractéristique 2, il est rare qu'il existe une extension intermédiaire $F' - L - \mathbb{F}_{q^n}(x)$ telle que L soit de genre 0 et d'indice 2 dans F'. En effet, la formule (6.6) montre que le genre de L est impair dès que $[\overline{\mathbb{F}_q}L : \overline{\mathbb{F}_q}(x)] \geq 2^3$, donc dès que $\overline{m} \geq 4$.

Déterminer une équation de $\mathcal{C}'$ est plus délicat qu'en caractéristique 2, sauf lorsque M_h est un polynôme irréductible sur $\mathbb{F}_2$ (on suppose encore que $m = \overline{m}$ et que σ se prolonge en un

automorphisme d'ordre n). Dans ce cas particulier, il n'existe pas de corps intermédiaire entre $\mathbb{F}_q(x)$ et F. En effet, si une telle extension $F - H - \mathbb{F}_q(x)$ existe, on en déduit une tour correspondante $F' - \mathbb{F}_{q^n} H - \mathbb{F}_{q^n}(x)$. Le corps $H' = \mathbb{F}_{q^n} H$ est donc stable par σ, et le sous-espace vectoriel U' de U correspondant doit être lui aussi stable par σ ; mais si M_h est irréductible, il n'existe pas de sous-espaces stables non triviaux de U. Tout élément $\tilde{y}$ de $F \setminus \mathbb{F}_q(x)$ engendre donc F sur $\mathbb{F}_q(x)$; dans la plupart des cas, $\tilde{y} = \sum_{i=0}^{n-1} y_i$ convient.

Exemple 6.2.8. *On considère la courbe elliptique définie sur* $\mathbb{F}_{5^4} \simeq \mathbb{F}_5(\theta)$ *où* $\theta^4+4\theta^3+\theta^2+4\theta+3 = 0$*, d'équation*

$$E : \ y^2 = \theta^{-1}x(x-\theta)(x-\theta^5).$$

La décomposition en facteurs irréductibles de X^4+1 *est* $(X+1)^4$*, donc nécessairement* $X+1|\overline{M}_h$*, ce qui implique que* $M_h = \overline{M}_h$ *et* $m = \overline{m}$*. En observant les racines de* h *et de ses images par* σ*, on trouve facilement que* $\overline{M}_h = (X+1)^3$ *et* $\overline{m} = 3$*. Par ailleurs l'extension* $F'/\mathbb{F}_{2^4}(x)$ *est ramifiée au-dessus des éléments de l'ensemble* $\{\infty, 0, \theta, \sigma(\theta), \sigma^2(\theta), \sigma^3(\theta)\}$ *donc* $r = 6$ *et le genre de* F' *vaut* $2^{3-2}(6-4)+1 = 5$*.*

Le degré de l'extension $n = 4$ *étant pair, il n'est pas certain que* σ *puisse se relever en un automorphisme d'ordre* 4 *; de fait, sur cet exemple on va voir que* σ *n'admet que des relevés d'ordre* 8*. Pour* $i = 0, \ldots, 3$*, on note comme précédemment* y_i *un élément tel que* $y_i^2 = h^{\sigma^i}$*, de telle sorte que* $F' = \mathbb{F}_{5^4}(x, y_0, y_1, y_2, y_3)$*. Comme* $m = 3$*, on a vu que* $F' = \mathbb{F}_{5^4}(x, y_0, y_1, y_2)$*, il existe donc une relation entre les* y_i *:*

$$\begin{aligned}(y_0y_1y_2y_3)^2 &= N_{\mathbb{F}_{5^4}/\mathbb{F}_5}(\theta^{-1})x^4(x-\theta)^2(x-\sigma(\theta))^2(x-\sigma^2(\theta))^2(x-\sigma^3(\theta))^2 \\ &= ((3\theta^2+\theta+3)x^2(x^4+4x^3+x^2+4x+3))^2.\end{aligned}$$

Sans perte de généralité, on peut poser $y_3 = (3\theta^2+\theta+3)x^2(x^4+4x^3+x^2+4x+3))/(y_0y_1y_2)$*,* $y_1 = \sigma(y_0)$*, et* $y_2 = \sigma(y_1)$*. Comme* $\sigma(y_2)^2 = h^{\sigma^3}$*, on a* $\sigma(y_2) = \pm y_3$*. Si on pose* $\sigma(y_2) = y_3$*, alors* $\sigma(y_3) = \sigma(3\theta^2+\theta+3)x^2(x^4+4x^3+x^2+4x+3)/(y_1y_2y_3) = y_0\,\sigma(3\theta^2+\theta+3)/(3\theta^2+\theta+3) = -y_0$ *; même chose si on pose* $\sigma(y_2) = -y_3$*. Dans tous les cas, on trouve* $\sigma^4(y_0) = -y_0$ *; tout relevé de* σ *est donc d'ordre* 8 *et l'attaque GHS ne fonctionne pas sur cette équation.*

Si l'on fait un changement de variables en remplaçant x *par* θx *et* y *par* θy*, on obtient pour* E *la nouvelle équation*

$$E : \ y^2 = x(x-1)(x-\theta^4) = x(x-1)(x-(\theta^3+4\theta^2+\theta+2)).$$

On voit facilement que pour cette équation $m = 4$ *et* $r = 7$*, l'ensemble des points de ramification contenant* ∞*,* 0*,* 1 *et les conjugués par* σ *de* θ^4*. Le genre de la courbe* $\mathcal{C}'$ *est donc* $2^{4-2}(7-4)+1 = 13$*, et puisque* $m = 4$*, cette courbe n'est a priori pas hyperelliptique. En posant* $\tilde{y} = y_0+y_1+y_2+y_3$*, on trouve qu'une équation de* $\mathcal{C}'$ *est donnée par*

$$\begin{aligned}&\tilde{y}^{16} + (3x^3+2x)\tilde{y}^{14} + (2x^6+4x^4+4x^3)\tilde{y}^{12} + (3x^9+2x^7+2x^6+2x^5+4x^4+2x^3)\tilde{y}^{10} \\ &+(2x^{12}+x^{10}+x^9+4x^8+2x^7+4x^6+3x^5+3x^4)\tilde{y}^8+(4x^{15}+2x^{13}+3x^{12}+3x^{11}+3x^{10}+3x^8+2x^7+2x^6+3x^5)\tilde{y}^6 \\ &+(x^{16}+3x^{14}+x^{13}+x^{12}+3x^{11}+2x^9+4x^8+4x^7+x^6)\tilde{y}^4+(2x^{16}+2x^{15}+x^{14}+4x^{13}+x^{12}+3x^{11}+3x^{10}+4x^9+x^8+4x^7)\tilde{y}^2 \\ &+4x^{18}+x^{17}+2x^{16}+3x^{15}+4x^{14}+2x^{13}+4x^{12}+3x^{11}+2x^{10}+x^9+4x^8 = 0.\end{aligned}$$

Si l'on considère maintenant la tordue quadratique de E *d'équation*

$$E' : \ y^2 = x(x-\theta)(x-\theta^5)$$

on a comme dans le premier cas $m = 3$, $r = 6$ et donc $g = 5$, et cette fois σ se relève bien en un automorphisme d'ordre 4 de F'. Bien que $m < 4$, la courbe $\mathcal{C}'$ n'est pas hyperelliptique (tous les sous-corps d'indice 2, donc avec $m = 2$, vérifient $r \geq 4$). En posant encore $\tilde{y} = y_0 + y_1 + y_2 + y_3$, on trouve qu'une équation de $\mathcal{C}'$ est donnée par

$$\tilde{y}^8 + (4x^3 + 3x^2 + 4x)\tilde{y}^6 + x^2\tilde{y}^4 + (3x^5 + x^4 + 3x^3)\tilde{y}^2 + 4x^4 = 0.$$

Sur ces exemples, il serait aussi bien entendu possible d'appliquer la technique GHS avec l'extension $\mathbb{F}_{5^4}/\mathbb{F}_{5^2}$ plutôt que $\mathbb{F}_{5^4}/\mathbb{F}_5$.

6.2.4 Marche d'isogénies

L'attaque GHS est d'autant plus efficace que le genre de la courbe $\mathcal{C}'$ obtenue est petit (idéalement égal à n). Lorsque l'on obtient un recouvrement d'une courbe elliptique E par une courbe de genre trop grand, il peut être parfois intéressant d'utiliser une suite d'isogénies de petits degrés, aussi appelée "marche d'isogénies", pour transférer le DLP défini sur E à une courbe E' qui serait plus vulnérable à l'attaque GHS, i.e. admettant un recouvrement par une courbe de genre plus petit.

Il existe essentiellement deux stratégies pour trouver une telle courbe E' lorsqu'elle existe [Hes05, section VIII.3] : soit on considère toutes les isogénies de petits degrés partant de E jusqu'à ce qu'une courbe faible soit trouvée, soit on parcourt l'ensemble des courbes faibles jusqu'à en trouver une qui soit isogène à E. La stratégie optimale dépend alors de la taille de la classe d'isogénies de E et du nombre de courbes faibles. On donnera un exemple d'utilisation de marche d'isogénies en section 8.2.1.

6.3 Le cas des extensions cubiques

L'objectif de cette section est de déterminer les courbes elliptiques définies sur $\mathbb{F}_{q^3}$ admettant un recouvrement de petit genre, en caractéristique impaire comme paire sous l'hypothèse que $m = \overline{m}$. On retrouve ainsi les classifications données par Thériault dans [Thé03] et Momose et Chao dans [MC05]. Dans le cas où le revêtement est hyperelliptique, on montre que l'application de recouvrement donnée par GHS se factorise par un quotient bi-elliptique, donnant une expression simple du revêtement.

6.3.1 Courbes vulnérables à la méthode GHS

Caractéristique impaire

Soit $\mathbb{F}_{q^3}$ un corps fini de caractéristique impaire. Le genre de la courbe définie sur $\mathbb{F}_q$, obtenue par la technique GHS appliquée à une courbe elliptique $E_{|\mathbb{F}_{q^3}}$, est donné par la formule (6.6) et dépend de deux paramètres m et r (on se placera toujours dans le cas où $\overline{m} = m$). La décomposition en facteurs irréductibles du polynôme $X^3 + 1$ sur $\mathbb{F}_2$ montre que les valeurs possibles de m sont $1, 2$ ou 3, le polynôme M_h valant respectivement $X + 1$, $X^2 + X + 1$ ou $X^3 + 1$. Dans le cas où $m = 1$, la courbe E est en fait définie sur $\mathbb{F}_q$, ce qui implique que le noyau de l'application conorme-norme contient le grand sous-groupe des points de trace nulle de $E(\mathbb{F}_{q^3})$; on exclut donc cette possibilité.

On s'intéresse maintenant aux valeurs admissibles pour r, et à la forme des équations des courbes correspondantes. On rappelle que l'entier r est le cardinal de l'ensemble R des abscisses dans $\mathbb{P}^1(\overline{\mathbb{F}_q}) \simeq \overline{\mathbb{F}_q} \cup \{\infty\}$ des points de 2-torsion des courbes E^{σ^i}. On pose comme précédemment $h \in \mathbb{F}_{q^3}[x]$ le polynôme tel que l'équation de E est donnée par $y^2 = h$, et on note R_0 l'ensemble des racines de h dans $\overline{\mathbb{F}_q}$, union $\{\infty\}$ si $\deg h = 3$. On a alors $R = R_0 \cup \sigma(R_0) \cup \sigma^2(R_0)$, et $\sigma^3(R_0) = R_0$. Comme $\#R_0 = 4$, le paramètre $r = \#R$ est nécessairement compris entre 4 et 12. Par ailleurs, on a $r = 4$ si et seulement si $\sigma(R_0) = R_0$, ce qui implique que h et h^σ ont les mêmes racines et donc que $m = 1$; ce cas a été exclu.

- Si $r = 5$, alors $R = R_0 \cup \{\alpha\}$. Comme $\sigma^3(R_0) = R_0$, l'élément $\alpha \in \overline{\mathbb{F}_q}$ vérifie $\sigma^3(\alpha) = \alpha$, et $\sigma(\alpha) \neq \alpha$ (sinon on aurait $\sigma(R_0) = R_0$). Par suite α appartient à $\mathbb{F}_{q^3} \setminus \mathbb{F}_q$, et l'ensemble R contient $\alpha, \sigma(\alpha), \sigma^2(\alpha)$ ainsi que deux autres éléments u et v globalement invariant par σ. L'équation de E est alors de la forme

$$y^2 = g(x)(x - \sigma(\alpha))(x - \sigma^2(\alpha)), \quad \alpha \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q,\ g \in \mathbb{F}_q[x],\ \deg(g) = 1 \text{ ou } 2. \tag{6.7}$$

  On vérifie aisément que pour une telle équation $m = 3$ et donc $g = 3$.
- Si $r = 6$, alors $R = R_0 \cup \{\alpha, \beta\}$. Comme $\sigma^3(R_0) = R_0$, l'ensemble $\{\alpha, \beta\}$ est aussi invariant par σ^3, mais pas par σ. Par conséquent soit σ^3 fixe α et β, qui sont donc des éléments de $\mathbb{F}_{q^3} \setminus \mathbb{F}_q$, soit σ^3 échange α et β, qui sont dans ce cas des éléments de $\mathbb{F}_{q^6} \setminus (\mathbb{F}_{q^3} \cup \mathbb{F}_{q^2})$. Par ailleurs R contient l'ensemble des conjugués de α et β par σ, et il y a deux possibilités.
  - L'ensemble $\{\sigma^i(\alpha)\} \cup \{\sigma^i(\beta)\}$ est de cardinal 6, et donc égal à R. Cela correspond au cas où $\alpha, \beta \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q$, $\beta \notin \{\alpha, \sigma(\alpha), \sigma^2(\alpha)\}$, et au cas où $\alpha \in \mathbb{F}_{q^6} \setminus (\mathbb{F}_{q^3} \cup \mathbb{F}_{q^2})$, $\beta = \sigma^3(\alpha)$. L'équation de E est alors d'une des deux formes suivantes, du type I

$$y^2 = (x - \sigma(\alpha))(x - \sigma^2(\alpha))(x - \sigma(\beta))(x - \sigma^2(\beta)), \quad \alpha, \beta \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q, \tag{6.8}$$

    ou du type II

$$y^2 = (x - \sigma(\alpha))(x - \sigma^2(\alpha))(x - \sigma^4(\alpha))(x - \sigma^5(\alpha)), \quad \alpha \in \mathbb{F}_{q^6} \setminus (\mathbb{F}_{q^3} \cup \mathbb{F}_{q^2}). \tag{6.9}$$

    Pour ces deux types le paramètre m vaut 2, et donc $g = 3$.
  - L'ensemble R contient strictement $\{\sigma^i(\alpha)\} \cup \{\sigma^i(\beta)\}$. Cela correspond au cas où $\alpha, \beta \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q$ et $\beta \in \{\sigma(\alpha), \sigma^2(\alpha)\}$; quitte à échanger α et β, on peut supposer $\beta = \sigma(\alpha)$. L'ensemble R est alors de la forme $\{\alpha, \sigma(\alpha), \sigma^2(\alpha), u, v, w\}$ où u, v et w sont trois autres éléments globalement invariants par σ, et l'équation de E est de la forme

$$y^2 = g(x)(x - \sigma^2(\alpha)), \quad \alpha \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q,\ g \in \mathbb{F}_q[x],\ \deg(g) = 2 \text{ ou } 3.$$

    Pour cette équation on vérifie que $m = 3$ et donc $g = 5$.

Les valeurs de r supérieures ou égales à 7 donnent des genres $g \geq 4$; les équations (6.7), (6.8) et (6.9) sont donc les seules pour lesquelles l'attaque GHS donne un revêtement de genre optimal $g = 3$. On va donner les équations des courbes obtenues, ainsi que des applications de recouvrement.

Soit $E_{|\mathbb{F}_{q^3}} : y^2 = h(x)$ une courbe elliptique de la forme (6.8) ou (6.9). Le polynôme $M_h = X^2 + X + 1$ est irréductible, ce qui implique que le $\mathbb{F}_2$-espace vectoriel U engendré par les classes d'équivalence de h, h^σ, h^{σ^2} n'admet pas de sous-espaces non triviaux stables par σ. En particulier l'extension $F/\mathbb{F}_q(x)$ n'admet pas de corps intermédiaires, et a priori F n'a pas de sous-corps rationnel d'indice 2 ; la courbe $\mathcal{C}'$ n'est pas hyperelliptique. En posant $\tilde{y} = y_0 + y_1 + y_2$, où $y_i^2 = h^{\sigma^i}$, on trouve qu'une équation de $\mathcal{C}'$ est

$$\tilde{y}^4 - 2(h + h^\sigma + h^{\sigma^2})\tilde{y}^2 - 8f\,\tilde{y} + (h + h^\sigma + h^{\sigma^2})^2 - 4(h\,h^\sigma + h^\sigma h^{\sigma^2} + h\,h^{\sigma^2}) = 0,$$

avec $f(x) = (x-\alpha)(x-\sigma(\alpha))(x-\sigma^2(\alpha))(x-\beta)(x-\sigma(\beta))(x-\sigma^2(\beta))$. Pour trouver sans trop de difficultés l'application de recouvrement $\pi : \mathcal{C}'(\mathbb{F}_{q^3}) \to E(\mathbb{F}_{q^3})$, on peut faire une élimination des variables y_1 et y_2 dans l'équation de $\mathcal{C}'$ en utilisant les relations qui lient $y_0, y_1, y_2, h, h^\sigma, h^{\sigma^2}$ et f. On obtient alors l'expression de y_0 en fonction de $\tilde{y}$, ce qui donne

$$\pi(x, \tilde{y}) = \left(x,\ \frac{h(h - h^\sigma - h^{\sigma^2} + \tilde{y}^2)}{2(\tilde{y}h + f)} \right).$$

Soit maintenant $E_{|\mathbb{F}_{q^3}}$ une courbe de la forme (6.7), d'équation

$$y^2 = h(x) = g(x)(x - \sigma(\alpha))(x - \sigma^2(\alpha))$$

où $\alpha \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q$ et $g \in \mathbb{F}_q[x]$, $\deg(g) = 1$ ou 2. Comme le polynôme $M_h = X^3 + 1$ n'est pas irréductible, l'extension $F/\mathbb{F}_q(x)$ admet des corps intermédiaires, donnés par les sous-espaces vectoriels non triviaux stables par σ de $U = \langle h, h^\sigma, h^{\sigma^2} \rangle \subset \mathbb{F}_{q^3}(x)^*/(\mathbb{F}_{q^3}(x)^*)^2$. On considère alors le corps $L' = \mathbb{F}_{q^3}(x, z_1, z_2)$, où $z_1 = y_1/y_0$ et $z_2 = y_2/y_1$ vérifient les équations

$$z_1^2 = \frac{x-\alpha}{x-\sigma(\alpha)}, \qquad z_2^2 = \frac{x-\sigma(\alpha)}{x-\sigma^2(\alpha)}.$$

Pour ce corps, on a $m = 2$ et $r = 3$, par conséquent son genre est 0, et L' est un sous-corps rationnel d'indice 2 de F'. La courbe $\mathcal{C}'$ est donc hyperelliptique. Pour trouver une équation simple de $\mathcal{C}'$, on cherche un générateur invariant par σ de L' sur $\mathbb{F}_{q^3}$. Suivant [MC05], on introduit

$$\phi : x \mapsto \frac{D}{x-\alpha} + \alpha \tag{6.10}$$

l'unique involution de $\mathbb{P}^1(\overline{\mathbb{F}_q})$ qui envoie α sur l'infini et $\sigma(\alpha)$ sur $\sigma^2(\alpha)$, autrement dit telle que $D = \left(\alpha - \sigma^2(\alpha)\right)(\alpha - \sigma(\alpha))$. On note $N(X) = (X-\alpha)(X-\sigma(\alpha))(X-\sigma^2(\alpha))$ le polynôme minimal de α sur $\mathbb{F}_q$, et $F \in \mathbb{F}_q[X]$ le polynôme tel que

$$F(X) = N(X)\left(X + \phi(X) + \phi^\sigma(X) + \phi^{\sigma^2}(X)\right).$$

Soit $w \in \overline{\mathbb{F}_q(x)}$ une racine du polynôme $F(X) - 4xN(X) \in \mathbb{F}_{q^3}(x)[X]$, de telle sorte que le corps $\mathbb{F}_{q^3}(x, w)$ est une extension de degré 4 de $\mathbb{F}_{q^3}(x)$. Comme $x = F(w)/4N(w)$, le corps $\mathbb{F}_{q^3}(x, w)$ est rationnel, égal à $\mathbb{F}_{q^3}(w)$. Par ailleurs z_1 et z_2 appartiennent à $\mathbb{F}_{q^3}(w)$: en effet, on vérifie que

$$z_1^2 = \frac{x-\alpha}{x-\sigma(\alpha)} = \frac{F(w) - 4\alpha N(w)}{F(w) - 4\sigma(\alpha)N(w)} = \left(\frac{(w-\alpha)(w-\phi(w))}{(w-\sigma(\alpha))(w-\phi^\sigma(w))} \right)^2$$

et z_2^2 a une expression similaire. Donc $L' \subset \mathbb{F}_{q^3}(w)$, et les deux corps étant des extensions de même degré de $\mathbb{F}_{q^3}(x)$, ils sont égaux : $L' = \mathbb{F}_{q^3}(w)$. En posant ensuite $\tilde{y} = y_0 y_1 y_2$, de telle sorte que $F' = \mathbb{F}_{q^3}(x, z_1, z_2, \tilde{y}) = \mathbb{F}_{q^3}(w, \tilde{y})$, on trouve qu'une équation de la courbe $\mathcal{C}'$ est $\tilde{y}^2 = g(x)^3 N(x)^2 = g\left(\frac{F(w)}{4N(w)}\right)^3 N\left(\frac{F(w)}{4N(w)}\right)^2$. Si l'on pose finalement $\bar{y} = \frac{2N(w)}{N(x)g(x)}\tilde{y}$, on obtient l'équation suivante :

$$\mathcal{C}' :\ \bar{y}^2 = 4N(w)^2 g\left(\frac{F(w)}{4N(w)}\right) = aF(w)^2/4 + bF(w)N(w) + 4cN(w)^2$$

où $g(X) = aX^2 + bX + c \in \mathbb{F}_q[X]$. Pour l'application de recouvrement $\pi : \mathcal{C}'(\mathbb{F}_{q^3}) \to E(\mathbb{F}_{q^3})$, on connaît déjà l'expression de x en fonction de w, et il faut trouver celle de y_0 en fonction de $\bar{y}$ et w. Comme $y_0 = \dfrac{y_0y_1y_2}{y_1y_2} = \dfrac{\tilde{y}}{y_2^2}\dfrac{y_2}{y_1} = \dfrac{\tilde{y}\, z_2}{h^{\sigma^2}}$, on obtient finalement

$$\pi : (w, \bar{y}) \to \left(\frac{F(w)}{4N(w)}, \bar{y}\frac{(w - \phi^{\sigma}(w))(w - \phi^{\sigma^2}(w))}{8N(w)(w - \alpha)} \right).$$

Caractéristique 2

Il est possible de mener la même analyse en caractéristique 2. Soit donc $\mathbb{F}_{q^3}$ un corps fini de caractéristique 2, et $E_{|\mathbb{F}_{q^3}}$ une courbe elliptique mise sous forme d'Artin-Schreier $y^2 + y = \beta x + \alpha + \gamma/x$; on demande encore que $m = \overline{m} \neq 1$. La formule (6.4) montre que le genre du revêtement dépend essentiellement des deux polynômes M_β et M_γ, qui peuvent prendre comme valeurs $X + 1$, $X^2 + X + 1$ ou $X^3 + 1$. Neuf cas sont donc possibles (six seulement si on observe que les rôles de β et γ sont interchangeables), et les valeurs correspondantes de m et du genre sont résumées dans la table 6.1. On note de plus que $M_\beta = X + 1$ si et seulement si $\beta \in \mathbb{F}_q$, et $M_\beta = X^2 + X + 1$ si et seulement si $\mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\beta) = 0$; la même chose est bien sûr vraie pour γ.

M_γ \ M_β	$X + 1$	$X^2 + X + 1$	$X^3 + 1$
$X + 1$	$m = 1$ exclu	$m = 3$ $g = 3$	$m = 3$ $g = 4$
$X^2 + X + 1$	$m = 3$ $g = 3$	$m = 2$ $g = 3$	$m = 3$ $g = 6$
$X^3 + 1$	$m = 3$ $g = 4$	$m = 3$ $g = 6$	$m = 3$ $g = 7$

TABLE 6.1 – Valeurs possibles du genre pour l'attque GHS sur $E_{|\mathbb{F}_{q^3}}$ en caractéristique paire.

Le genre optimal $g = 3$ est atteint pour seulement deux combinaisons : si β et γ sont tous deux de trace nulle, ou si l'un des deux est de trace nulle et l'autre dans $\mathbb{F}_q$. Dans le premier cas, pour avoir $m = \overline{m}$ il faut que $\mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_2}(\alpha) = 0$; quitte à remplacer y par $y + s$ où $s^2 + s = \alpha$, on peut supposer $\alpha = 0$, et l'équation de E est de la forme

$$y^2 + y = \beta x + \gamma/x, \quad \beta, \gamma \in \mathbb{F}_{q^3},\ \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\beta) = \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\gamma) = 0. \tag{6.11}$$

Le polynôme $M_h = X^2 + X + 1$ est irréductible, donc l'extension $F/\mathbb{F}_q(x)$ n'admet pas de sous-corps d'indice 2 et $\mathcal{C}'$ n'est en général pas hyperelliptique. Pour déterminer une équation du relèvement, plutôt que de travailler avec un élément dépendant du choix d'une base normale on peut considérer $\tilde{y} = y_0y_1y_2$, avec $y_i^2 + y_i = \beta^{\sigma^i} x + \gamma^{\sigma^i}/x$ et $y_0 + y_1 + y_2 = 0$. On obtient alors l'équation suivante où, pour simplifier, Tr et N désignent la norme et la trace relativement à l'extension $\mathbb{F}_{q^3}/\mathbb{F}_q$:

$$\begin{aligned} x^6\tilde{y}^4 + \left(\mathrm{Tr}(\beta\sigma(\beta))x^8 + \mathrm{Tr}(\beta\gamma)x^6 + \mathrm{Tr}(\gamma\sigma(\gamma))\right)\tilde{y}^2 \\ + \left(N(\beta)x^9 + \mathrm{Tr}(\beta^2\gamma)x^7 + \mathrm{Tr}(\beta\gamma^2)x^5 + N(\gamma)x^3\right)\tilde{y} \\ + \left(N(\beta)x^6 + \mathrm{Tr}(\beta^2\gamma)x^4 + \mathrm{Tr}(\beta\gamma^2)x^2 + N(\gamma)\right)^2 = 0. \end{aligned}$$

Dans le deuxième cas, quitte à échanger β et γ et à remplacer x par x/β, on peut supposer que $\beta = 1$ et l'équation de E est donc de la forme

$$y^2 + y = x + \alpha + \gamma/x, \quad \alpha, \gamma \in \mathbb{F}_{q^3},\ \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\gamma) = 0. \tag{6.12}$$

On a vu que le revêtement obtenu par GHS d'une telle courbe est hyperelliptique, et que le sous-corps rationnel d'indice 2 de F' est donné par $L' = \mathbb{F}_{q^3}(x, z_1, z_2)$ où $z_1 = y_1 + y_0$, $z_2 = y_2 + y_1$, et $y_i^2 + y_i = x + \alpha^{\sigma^i} + \gamma^{\sigma^i}/x$, de telle sorte que

$$z_1^2 + z_1 = \sigma(\alpha) + \alpha + \sigma^2(\gamma)/x, \qquad z_2^2 + z_2 = \sigma^2(\alpha) + \sigma(\alpha) + \gamma/x.$$

La difficulté est de trouver un générateur de L' sur $\mathbb{F}_{q^3}$, invariant par σ. Comme dans le cas de la caractéristique impaire, on introduit

$$\phi : x \mapsto \frac{\sigma(\gamma)\sigma^2(\gamma)}{x + \gamma} + \gamma \tag{6.13}$$

l'unique involution de $\mathbb{P}^1(\overline{\mathbb{F}_q})$ qui envoie γ sur l'infini et $\sigma(\gamma)$ sur $\sigma^2(\gamma)$. On note encore $N(X) = (X+\gamma)(X+\sigma(\gamma))(X+\sigma^2(\gamma))$ le polynôme minimal de γ sur $\mathbb{F}_q$, et $F \in \mathbb{F}_q[X]$ le polynôme tel que

$$F(X) = N(X)\left(X + \phi(X) + \phi^{\sigma}(X) + \phi^{\sigma^2}(X)\right) = \left(X^2 + \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\gamma\sigma(\gamma))\right)^2.$$

Soit $w \in \overline{\mathbb{F}_q(x)}$ une racine du polynôme $F(X) + xN(X) \in \mathbb{F}_{q^3}(x)[X]$, de telle sorte que le corps $\mathbb{F}_{q^3}(x, w)$ est une extension de degré 4 de $\mathbb{F}_{q^3}(x)$. Comme $x = F(w)/N(w)$, le corps $\mathbb{F}_{q^3}(x, w)$ est rationnel, égal à $\mathbb{F}_{q^3}(w)$. Par ailleurs z_1 et z_2 appartiennnent à $\mathbb{F}_{q^3}(w)$: en effet, on vérifie que

$$z_1^2 + z_1 = \sigma(\alpha) + \alpha + \frac{\sigma^2(\gamma)}{x} = \sigma(\alpha) + \alpha + \sigma^2(\gamma)\frac{N(w)}{F(w)} = \wp\left(\delta + \frac{\sigma^2(\gamma)(w + \sigma^2(\gamma))}{w^2 + \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\gamma\sigma(\gamma))}\right)$$

où $\delta \in \mathbb{F}_{q^3}$ est tel que $\delta^2 + \delta = \sigma(\alpha) + \alpha$, et z_2 a une expression similaire. Donc $L' \subset \mathbb{F}_{q^3}(w)$, et les deux corps étant des extensions de même degré de $\mathbb{F}_{q^3}(x)$, ils sont égaux : $L' = \mathbb{F}_{q^3}(w)$. En posant ensuite $\tilde{y} = y_0 + y_1 + y_2$, de telle sorte que $F' = \mathbb{F}_{q^3}(x, z_1, z_2, \tilde{y}) = \mathbb{F}_{q^3}(w, \tilde{y})$, on trouve qu'une équation de la courbe $\mathcal{C}'$ est $\tilde{y}^2 + \tilde{y} = x + \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\alpha) = F(w)/N(w) + \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\alpha)$, ce qui en notant $\bar{y} = N(w)\,\tilde{y}$ se réécrit

$$\mathcal{C}' :\ \bar{y}^2 + N(w)\bar{y} = F(w)N(w) + \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\alpha)N(w)^2.$$

Pour l'application de recouvrement $\pi : \mathcal{C}'(\mathbb{F}_{q^3}) \to E(\mathbb{F}_{q^3})$, on connaît déjà l'expression de x en fonction de w, et il faut trouver celle de $y = y_0$ en fonction de $\bar{y}$ et w. Comme $y_0 = \tilde{y} + y_1 + y_2 = \bar{y}/N(w) + z_2$, on obtient finalement

$$\pi : (w, \bar{y}) \mapsto \left(\frac{F(w)}{N(w)}, \frac{\bar{y}}{N(w)} + \sigma(\delta) + \frac{\gamma(w+\gamma)}{w^2 + \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\gamma\sigma(\gamma))}\right).$$

6.3.2 Quotients bi-elliptiques

Les recouvrements hyperelliptiques de genre 3 obtenus ci-dessus présentent la particularité de se factoriser par une courbe elliptique intermédiaire : en caractéristique 2 comme impaire, il existe une courbe elliptique $E'_{/\mathbb{F}_{q^3}}$ telle que l'application de recouvrement $\pi : \mathcal{C}'(\mathbb{F}_{q^3}) \to E(\mathbb{F}_{q^3})$, de degré

4, soit la composée des deux morphismes $\pi_1 : \mathcal{C}'(\mathbb{F}_{q^3}) \to E'(\mathbb{F}_{q^3})$ et $\pi_2 : E'(\mathbb{F}_{q^3}) \to E(\mathbb{F}_{q^3})$ de degré 2. Autrement dit, l'extension $F'/\mathbb{F}_{q^3}(E)$ contient un corps intermédiaire $\mathbb{F}_{q^3}(E')$ de genre 1. Cette observation montre qu'il est possible d'obtenir des recouvrements de petit genre qui sont différents et de plus petit degré que ceux obtenus directement par GHS.

En caractéristique impaire, pour E de la forme (6.7), le corps $\mathbb{F}_{q^3}(E')$ est égal à $\mathbb{F}_{q^3}(x, y, z)$ où $z = y_2/y_1$ vérifie $z^2 = \frac{x-\sigma(\alpha)}{x-\sigma^2(\alpha)}$ (de telle sorte que $m = 2$ et $r = 4$). On a donc $x = \frac{\sigma^2(\alpha)z^2-\sigma(\alpha)}{z^2-1}$, et $\mathbb{F}_{q^3}(x, y, z) = \mathbb{F}_{q^3}(y, z)$, où y et z sont reliés par la relation

$$\left(\frac{(z^2-1)^2}{z(\sigma^2(\alpha)-\sigma(\alpha))}y\right)^2 = (z^2-1)^2\, g\left(\frac{\sigma^2(\alpha)z^2-\sigma(\alpha)}{z^2-1}\right).$$

En posant $z' = \frac{2\sigma^2(\alpha)z-2\sigma(\alpha)}{z-1}$ et $y' = 4y\frac{z'-\sigma(\alpha)-\sigma^2(\alpha)}{z'-2\sigma(\alpha)}$, une équation de E' est donnée par

$$E' :\ y'^2 = 4\left(z' - \sigma(\alpha) - \sigma^2(\alpha)\right)^2\, g\left(\frac{z'^2 - 4\sigma(\alpha)\sigma^2(\alpha)}{4(z' - \sigma(\alpha) - \sigma^2(\alpha))}\right).$$

L'application de recouvrement de $\mathcal{C}'$ dans E' correspondante est alors

$$\pi : (w, \bar{y}) \mapsto \left(w + \phi(w), \frac{\bar{y}}{(w-\alpha)^2}\right)$$

où ϕ est l'involution donnée en (6.10). La courbe E' est ainsi obtenue comme le quotient de $\mathcal{C}'$ par l'unique involution $\varphi \in \mathrm{Aut}(\mathcal{C}')$ relevant $\phi \in \mathrm{Aut}(\mathbb{P}^1)$, ce qu'on note $E' = \mathcal{H}/\langle\varphi\rangle$; une telle involution est appelée *bi-elliptique*. Plus généralement, une courbe est appelée bi-elliptique si elle possède une involution telle que le quotient par cette involution est une courbe elliptique. Les conjugués de φ par σ sont encore des involutions bi-elliptiques de $\mathcal{C}'$, et l'ensemble $\{Id, \varphi, \varphi^\sigma, \varphi^{\sigma^2}\}$ forme un sous-groupe du groupe des automorphismes de $\mathcal{C}'$; le revêtement $\pi : \mathcal{C}' \to E$ correspond alors au quotient par ce sous-groupe, ce qui justifie a posteriori l'introduction de ϕ et des polynômes F et N à la section précédente.

En caractéristique paire, la courbe intermédiaire E' est encore un quotient de $\mathcal{C}'$ par l'involution bi-elliptique qui relève l'homographie ϕ donnée en (6.13). L'application π_2 de E' dans E est une isogénie de degré 2, duale du morphisme de Frobenius σ_2 relatif à l'extension $\mathbb{F}_{q^3}/\mathbb{F}_2$ (voir [Sil86]), et E' est isomorphe à $E^{\sigma_2^{-1}}$.

Plutôt que d'utiliser la techique GHS pour obtenir des revêtements de genre 3, il est donc possible de chercher directement quelles sont les courbes hyperelliptiques $\mathcal{H}_{|\mathbb{F}_q}$ de genre 3 admettant une involution biellipticque ϕ qui soit définie sur $\mathbb{F}_{q^3}$ (et pas sur $\mathbb{F}_q$), de telle sorte que la courbe elliptique quotient E soit définie sur $\mathbb{F}_{q^3}$. Cette recherche exhaustive étant assez technique, les détails sont reportés en annexe. La conclusion de celle-ci est que, en dehors du cas présenté ci-dessus, les seuls courbes elliptiques $E_{|\mathbb{F}_{q^3}}$ obtenues ainsi sont toutes $\mathbb{F}_{q^3}$-isomorphes à des courbes définies sur $\mathbb{F}_q$. Ces revêtements sont donc moins pertinents pour la cryptographie, mais restent intéressants pour l'attaque du DLP dans le groupe des points de trace nulle d'une telle courbe (cf. [DS03, section 4]).

Chapitre 7

Attaques par décomposition

En 2004, Semaev propose pour la première fois d'attaquer directement le problème du logarithme discret sur les courbes elliptiques par des méthodes de calcul d'indices, sans passer par un transfert du problème à un groupe où la complexité du DLP serait plus simple que celle donnée par les attaques génériques. Le principe est d'exploiter la loi de groupe existante sur l'ensemble des points rationnels $E(\mathbb{F}_q)$ d'une courbe elliptique donnée, afin de donner une condition polynomiale sur les abscisses de points $(P_i)_{i=1\ldots m}$ intervenant dans une relation de la forme $P_1 + \ldots + P_m = \mathcal{O}_E$. Malheureusement, il semble difficile dans cette approche de convenir à la fois une base de factorisation composée de points privilégiés (ayant par exemple des "petites" abscisses) ainsi que d'un algorithme permettant de calculer efficacement des solutions de "petites" valeurs du système polynomial associé à la recherche d'une relation.

Suite à cette tentative, Gaudry et Diem [Gau08, Die11] ont indépendamment proposé d'appliquer la méthode de Semaev à des courbes elliptiques définies sur des extensions de corps finis. Étant donnée une courbe elliptique E définie sur une extension de corps $\mathbb{F}_{q^n}$, l'idée consiste à choisir la base de factorisation comme l'ensemble des points de $E(\mathbb{F}_{q^n})$ dont l'abscisse est dans $\mathbb{F}_q$; on peut alors ramener la recherche de relations sur les points de cette base de factorisation à la résolution de systèmes polynomiaux définis sur $\mathbb{F}_q$, obtenus grâce aux polynômes de sommation de Semaev et à la restriction de Weil de $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$.

On détaille dans la première section de ce chapitre la construction explicite des polynômes de sommation de Semaev permettant de caractériser simplement que la somme de m points d'une courbe elliptique est égale au point à l'infini. Ces polynômes étant par construction symétriques, on montre comment il est possible d'en calculer directement une expression en terme des polynômes symétriques élémentaires. On analyse ensuite la méthode de décomposition de Gaudry et Diem permettant d'attaquer le problème du logarithme discret d'une courbe elliptique définie sur une extension, ou plus généralement une variété abélienne de petite dimension. On montre alors comment l'approche de Nagao permet de généraliser l'idée de Semaev au cas des courbes de genre supérieur pour la recherche de relations. Dans la dernière section, on présente de nouveaux résultats sur les attaques par décomposition. On introduit notamment une variante de la méthode de décomposition de Gaudry donnant une meilleure complexité asymptotique pour l'attaque de courbes elliptiques définies sur $\mathbb{F}_{q^n}$ dès que $\log_2 q \leq \frac{3-\omega}{2} n^3$, où ω est l'exposant intervenant dans la complexité effective du produit matriciel (cf. section 2.2.2). En particulier, notre approche permet d'obtenir des relations sur $E(\mathbb{F}_{q^5})$, ce qui n'est pas le cas avec l'approche originale de [Gau08, Die11]. On donne ensuite un exemple pratique d'application sur $\mathbb{F}_{2^{155}}$, permettant notamment la résolution du problème Diffie-Hellman statique assisté d'un oracle (SDHP) sur la courbe d'Oakley 'Well Known

Group 3' proposée dans les standards IPSEC. Enfin, on propose une variante de l'approche de Nagao permettant d'accélérer en pratique la recherche de relations en genre $g > 1$; cette variante s'accompagne d'une technique de crible, compatible notamment avec les méthodes de variations "larges primes" présentées en section 4.3.2. En pratique, on montrera comment cette méthode est particulièrement bien adaptée au cas des courbes hyperelliptiques définies sur des extensions quadratiques.

7.1 Polynômes de sommation de Semaev

Dans [Sem04], Semaev tente de définir un calcul d'indices sur courbe elliptique. L'idée essentielle est de ne considérer que des relations faisant intervenir un nombre fixe de points de la courbe ; on peut ainsi utiliser la loi de groupe définie sur les points rationnels de la courbe pour transcrire la recherche de relations en la résolution de systèmes polynomiaux multivariés, dont les inconnues sont précisément les coordonnées des points de la relation cherchée. Semaev simplifie alors les conditions polynomiales obtenues en introduisant les *polynômes de sommation*, qui n'ont comme variables que les abscisses des points intervenant dans la relation. Ceux-ci étant par nature symétriques, il est possible de les exprimer en fonction des polynômes symétriques élémentaires afin d'en diminuer le degré total et d'en faciliter la résolution. Il est à noter cependant que la taille de ces polynômes croît exponentiellement avec le nombre de variables m, et que leur calcul devient donc rapidement prohibitif lorsque m augmente. On détaille dans cette section la construction originale proposée dans [Sem04], ainsi que deux autres méthodes efficaces permettant de construire directement ces polynômes sous forme symétrisée. En particulier, pour une courbe elliptique définie sur un corps de taille 160 bits, la première méthode apporte une réduction du coût mémoire par un facteur 25 et une amélioration du temps de calcul par un facteur au moins égal à 3, tandis que la deuxième permet de gagner un facteur 10 sur le temps de calcul du cinquième polynôme de Semaev symétrisé.

Dans toute cette section, E désigne une courbe elliptique définie sur un corps K, donnée sous forme de Weierstrass, et $\mathcal{O}$ son point à l'infini.

7.1.1 Définition, propriétés

Proposition 7.1.1. *Pour tout $m \geq 2$, il existe un polynôme $f_m \in K[X_1, \ldots, X_m]$ appelé* m-ième polynôme de sommation de Semaev*, qui est irréductible, symétrique, de degré 2^{m-2} en chaque variable, et pour lequel étant donnés $P_1 = (x_{P_1}, y_{P_1}), \ldots, P_m = (x_{P_m}, y_{P_m}) \in E(\overline{K}) \setminus \{\mathcal{O}\}$, on a*

$$f_m(x_{P_1}, \ldots, x_{P_m}) = 0 \Leftrightarrow \exists \epsilon_1, \ldots, \epsilon_m \in \{1, -1\},\ \epsilon_1 P_1 + \ldots + \epsilon_m P_m = \mathcal{O}. \tag{7.1}$$

La preuve de l'existence de ces polynômes de sommation est constructive et s'obtient par récurrence en utilisant la propriété 2.2.2 du résultant. En effet, si l'on suppose la propriété (7.1) vraie jusqu'à un certain rang $m-1$, alors

$$\begin{aligned}
P_1 \pm P_2 \pm \ldots \pm P_m = \mathcal{O} \quad &\Leftrightarrow \quad \forall j \in [\![1; m-3]\!], \exists R \in E(\overline{K}),\ \begin{cases} P_1 \pm \ldots \pm P_j + R = \mathcal{O} \\ R \mp P_{j+1} \mp \ldots \mp P_m = \mathcal{O} \end{cases} \\
&\Leftrightarrow \quad \forall j \in [\![1; m-3]\!], \quad f_{j+1}(x_{P_1}, \ldots, x_{P_j}, X) \\
&\qquad \text{et } f_{m-j+1}(X, x_{P_{j+1}}, \ldots, x_{P_m}) \text{ ont une racine commune}^{\,1} \\
&\Leftrightarrow \quad \forall j \in [\![1; m-3]\!], \quad \mathrm{Res}_X \left(f_{j+1}(x_{P_1}, \ldots, x_{P_j}, X)\ , \right. \\
&\qquad\qquad \left. f_{m-j+1}(X, x_{P_{j+1}}, \ldots, x_{P_m}) \right) = 0
\end{aligned}$$

En particulier, le m-ième polynôme de Semaev s'obtient en calculant

$$f_m(X_1, \ldots, X_m) = \mathrm{Res}_X\left(f_{j+1}(X_1, \ldots, X_j, X), f_{m-j+1}(X, X_{j+1}, \ldots, X_m)\right)$$

pour n'importe quel $j \in [\![1; m-3]\!]$. On renvoie à l'article [Sem04] pour les détails du reste de la preuve.

Pour les petites valeurs de m, il est facile de donner une expression de $f_m(X)$. Par exemple, si l'on considère deux points $P_1, P_2 \in E(\overline{K})$ où E est donnée par une équation de Weierstrass générale de la forme :

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

alors $P_1 \pm P_2 = \mathcal{O}$ si et seulement si $x(P_1) = x(P_2)$. Autrement dit, le deuxième polynôme de Semaev est

$$f_2(X_1, X_2) = X_1 - X_2.$$

De la même façon, on peut déterminer le troisième polynôme de Semaev en considérant l'équation de la droite $\ell:\ y - \lambda x - \mu = 0$ (normalisée au point $\mathcal{O}$) passant par P_1, P_2 et $P_3 \in E(\overline{K})$ tels que $P_1 \pm P_2 \pm P_3 = \mathcal{O}$. En utilisant l'équation de la courbe, il est alors possible de se débarrasser de la variable y en multipliant par l'expression conjuguée afin d'obtenir un polynôme en x dont les trois racines sont exactement les abscisses x_1, x_2 et x_3 des trois points considérés :

$$\begin{aligned}(y - (\lambda x + \mu))(y + a_1x + a_3 + \lambda x + \mu) &= y(y + a_1x + a_3) - (\lambda x + \mu)^2 - (a_1x + a_3)(\lambda x + \mu)\\ &= x^3 - (a_1\lambda + \lambda^2 - a_2)x^2 + (a_4 - a_3\lambda - a_1\mu - 2\lambda\mu)x\\ &\quad -(a_3\mu + \mu^2 - a_6).\end{aligned}$$

En éliminant les variables λ et μ dans le système obtenu en identifiant les coefficients de ce dernier polynôme avec les polynômes symétriques de x_1, x_2 et x_3, on obtient le troisième polynôme de Semaev :

$$\begin{aligned}f_3(X_1, X_2, X_3) = X_1^2X_2^2 + X_1^2X_3^2 + X_2^2X_3^2 - 2X_1^2X_2X_3 - 2X_1X_2^2X_3 - 2X_1X_2X_3^2 - (a_1^2 + 4a_2)X_1X_2X_3\\ - (a_1a_3 + 2a_4)(X_1X_2 + X_1X_3 + X_2X_3) - (a_3^2 + 4a_6)(X_1 + X_2 + X_3) - a_1^2a_6 + a_1a_3a_4 - a_2a_3^2 - 4a_2a_6 + a_4^2.\end{aligned}$$

En particulier, en caractéristique différente de 2 et 3, l'expression du troisième polynôme de Semaev de la courbe E d'équation réduite $y^2 = x^3 + ax + b$ devient

$$f_3(X_1, X_2, X_3) = (X_1 - X_2)^2X_3^2 - 2\left((X_1 + X_2)(X_1X_2 + a) + 2b\right)X_3 + (X_1X_2 - a)^2 - 4b(X_1 + X_2),$$

et si l'on considère une courbe ordinaire définie sur $\mathbb{F}_{2^d}$ d'équation réduite $y^2 + xy = x^3 + ax^2 + b$, le troisième polynôme de Semaev se simplifie en

$$f_3(X_1, X_2, X_3) = (X_1X_2 + X_1X_3 + X_2X_3)^2 + X_1X_2X_3 + b.$$

7.1.2 Calculs des polynômes de sommations symétrisés

Première méthode

On note que cette approche géométrique pour la construction de f_3 permet en fait d'en calculer directement une expression en fonction des polynômes symétriques élémentaires. Cette méthode se généralise sans difficulté pour la construction de f_m et permet de remplacer le calcul de résultant

1. Il faut faire attention à distinguer les cas où $R = \mathcal{O}$ ou pas.

ainsi que sa symétrisation par un calcul de base de Gröbner pour un certain ordre d'élimination. En effet, soit $D = (P_1) + \ldots + (P_m) - m(\mathcal{O}) \in Div^0_{\overline{K}}(E)$ le diviseur principal associé aux points $P_1, \ldots, P_m \in E(\overline{K})$ tels que $P_1 + \ldots + P_m = \mathcal{O}$. On peut définir à constante près une fonction $g_m \in \overline{K}(E)$ telle que $D = div(g_m)$ et celle-ci peut être calculée explicitement à l'aide de techniques similaires à celles utilisées dans l'algorithme de Miller [Mil04] :

Soient $l_i(X, Y) = 0$ $(1 \leq i \leq m-1)$ les équations des droites passant par $P_1 + \ldots + P_i$ et P_{i+1}, et $v_i(X, Y) = 0$ $(1 \leq i \leq m-2)$ les équations des droites verticales passant par $P_1 + \ldots + P_{i+1}$, alors

$$g_m(X, Y) = \frac{l_1 \ldots l_{m-1}}{v_1 \ldots v_{m-2}}(X, Y).$$

Une simple récurrence permet alors de montrer que

$$g_m(X, Y) = g_{m,1}(X) + Y g_{m,2}(X) \tag{7.2}$$

où $g_{m,1}$ et $g_{m,2}$ sont deux polynômes de degrés respectifs $d_{m,1}$ et $d_{m,2}$ tels que

$$d_{m,1} = \begin{cases} m/2 \text{ si } m \text{ est pair} \\ (m-1)/2 \text{ si } m \text{ est impair} \end{cases} \quad \text{et } d_{m,2} = \begin{cases} (m-4)/2 \text{ si } m \text{ est pair} \\ (m-3)/2 \text{ si } m \text{ est impair.} \end{cases}$$

On note que si les droites l_i et v_i sont normalisées au point à l'infini, alors la fonction g_m est uniquement déterminée. L'intersection de la courbe d'équation $g_m = 0$ avec E est alors exactement l'ensemble des points $\{P_1, \ldots, P_m\}$; en particulier on a la proposition suivante

Proposition 7.1.2. *Soit E une courbe elliptique donnée par une équation de Weierstrass de la forme E : $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ sur un corps K et soient $P_1, \ldots, P_m \in E(\overline{K})$ tels que $P_1 + \ldots + P_m = \mathcal{O}$.*
On considère les polynômes $g_{m,1}$ et $g_{m,2}$ donnés par l'équation (7.2)*. Alors,*

$$g_{m,1}(x)^2 - (a_1x + a_3)(g_{m,1}\, g_{m,2})(x) - (x^3 + a_2x^2 + a_4x + a_6)g_{m,2}(x)^2 = 0 \Leftrightarrow x \in \{x_{P_1}, \ldots, x_{P_m}\}.$$

Réciproquement, si $h_{m,1}$ et $h_{m,2}$ sont deux polynômes arbitraires de $K[X]$ de degrés respectifs $d_{m,1}$ et $d_{m,2}$, alors les racines de $F_m(X) = h_{m,1}(X)^2 - (a_1X + a_3)(h_{m,1}\, h_{m,2})(X) - (X^3 + a_2X^2 + a_4X + a_6)\, h_{m,2}(X)^2$ dans $\overline{K}$ comptées avec multiplicité sont les abscisses de points $Q_1, \ldots, Q_m \in E(\overline{K})$ tels que $Q_1 + \ldots + Q_m = \mathcal{O}$.

Démonstration. La réciproque repose sur le fait que dans $\overline{K}(E)$, on a

$$F_m(X) = (g_{m,1}(X) + Y\, g_{m,2}(X))\, (g_{m,1}(X) - (Y + a_1X + a_3)\, g_{m,2}(X))\,.$$

Le degré de F_m étant exactement m, les racines de F_m sont exactement les abscisses des points $(\pm)Q_i \in E(\overline{K})$. À un changement de signe près, on peut supposer que Q_i est un zéro de $g_{m,1} + Y g_{m,2}$ (et donc que $-Q_i$ est un zéro du deuxième facteur $g_{m,1} - Y(Y + a_1X + a_3)g_{m,2}$). En particulier le diviseur principal $div(g_{m,1} + Y g_{m,2})$ est égal à $(Q_1) + \ldots + (Q_m) - m(\mathcal{O})$ et donc $Q_1 + \ldots + Q_m = \mathcal{O}$. □

Grâce à ce résultat, on peut maintenant construire les polynômes de sommation symétrisés. On considère les polynômes suivants dans $A[X] = \overline{K}[\alpha_0, \ldots, \alpha_{d_{m,1}}, \beta_0, \ldots, \beta_{d_{m,2}}][X]$:

$$h_{m,1}(X) = \sum_{i=0}^{d_{m,1}} \alpha_i X^i, \quad h_{m,2}(X) = X^{d_{m,2}} + \sum_{i=0}^{d_{m,2}-1} \beta_i X^i,$$

$$F_m(X) = h_{m,1}(X)^2 - (a_1X + a_3)(h_{m,1}\, h_{m,2})(X) - (X^3 + a_2X^2 + a_4X + a_6)\, h_{m,2}(X)^2.$$

Suivant la même démarche que pour le calcul du troisième polynôme de Semaev, on peut identifier les coefficients de F_m avec les polynômes symétriques élémentaires $e_1, \ldots, e_m$ de $x_{P_1}, \ldots, x_{P_m}$ et obtenir ainsi un idéal, noté J, de $\overline{K}[\alpha_0, \ldots, \alpha_{d_{m,1}}, \beta_0, \ldots, \beta_{d_{m,2}}, e_1, \ldots, e_m]$. Le calcul de la base de Gröbner de J pour le m-ième ordre d'élimination donne alors une base de générateurs de l'idéal $J' = J \cap K[e_1, \ldots, e_m]$. La proposition 7.1.2 permet alors d'affirmer qu'un m-uplets $(e_1, \ldots, e_m)$ appartient à l'ensemble algébrique $\mathbb{V}(J')$ si et seulement les racines du polynôme $T^m + \sum_{i=1}^{m}(-1)^i e_i T^{m-i}$ sont les abscisses de points de $E(\overline{K})$ dont la somme est égale au point à l'infini $\mathcal{O}$. En particulier, les résultats d'existence et d'unicité des polynômes de Semaev permettent de voir que l'idéal J' ainsi obtenu est principal et engendré par $f_m(e_1, \ldots, e_m)$; on obtient donc avec cette élimination le calcul du m-ième polynôme de Semaev directement symétrisé. Cette méthode est à rapprocher de l'idée de Nagao pour le calcul d'indices sur les jacobiennes de courbes de genre plus grand que 1, voir section 7.2.3.

Exemple 7.1.3. *Soit E une courbe elliptique définie sur un corps K de caractéristique différente de 2 et 3, ayant pour équation de Weierstrass E : $y^2 = x^3 + ax + b$. Le calcul du cinquième polynôme de Semaev f_5 s'obtient en considérant le polynôme*

$$F_5(X) = (\alpha_2X^2 + \alpha_1X + \alpha_0)^2 - (X^3 + aX + b)(X + \beta_0)^2.$$

En identifiant les coefficients de ce polynôme avec les polynômes symétriques élémentaires $e_1, \ldots, e_5$ en les variables $x_{P_1}, \ldots, x_{P_4}, x_{P_5}$, on déduit le système polynomial

$$\begin{cases} e_1 = \alpha_2^2 - 2\beta_0 \\ e_2 = \beta_0^2 + a - 2\alpha_1\alpha_2 \\ e_3 = 2\alpha_0\alpha_2 + \alpha_1^2 \\ e_4 = a\beta_0^2 + 2b\beta_0 - 2\alpha_0\alpha_1 \\ e_5 = \alpha_0^2 - b\beta_0^2. \end{cases}$$

Le calcul d'une base de Gröbner pour le quatrième ordre d'élimination de l'idéal J de $\overline{K}[\alpha_0, \alpha_1, \alpha_2, \beta_0, e_1, \ldots, e_5]$ correspondant permet alors d'obtenir directement l'expression de $f_5(e_1, \ldots, e_5)$.

Deuxième méthode

On présente ici une autre méthode, plus efficace, permettant de calculer les polynômes de Semaev directement en fonction des polynômes symétriques élémentaires. L'idée consiste à symétriser partiellement les résultants calculés à chaque étape de la récurrence : ceci permet alors de réduire à la fois les tailles des polynômes intermédiaires ainsi que le coût de la symétrisation finale en la distribuant sur les différentes étapes de la récurrence. La proposition suivante résume cette approche :

Proposition 7.1.4. *Soit E une courbe elliptique donnée par une équation de Weierstrass de la forme E : $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ sur un corps K. Les polynômes de sommation symétrisés sont déterminés par la récurrence suivante :*

$$\begin{aligned}\tilde{f}_3(e_{1,2}, e_{2,2}, X_3) = \big(e_{1,2}^2 - 4e_{2,2}\big)X_3^2 - \big(2e_{1,2}e_{2,2} + (a_1a_3 + 2a_4)e_{1,2} + (a_1^2 + 4a_2)e_{2,2} + a_3^2 + 4a_6\big)X_3 \\ - a_3^2e_{1,2} - 4a_6e_{1,2} + e_{2,2}^2 - a_1a_3e_{2,2} - 2a_4e_{2,2} - a_1^2a_6 + a_1a_3a_4 - a_2a_3^2 - 4a_2a_6 + a_4^2,\end{aligned}$$

et pour $m \geq 3$,

$$\tilde{f}_{m+1}(e_{1,m}\,,\,\ldots,e_{m,m}\,,\,X_{m+1}) = Sym_m\left(Res_Y\left(\tilde{f}_m(e_{1,m-1},\ldots,e_{m-1,m-1},Y), f_3(e_{1,1},X_{m+1},Y)\right)\right),$$

où

- $e_{r,n}$ *est le* r*-ième polynôme symétrique élémentaire en les variables* $X_1,\ldots,X_n$ *;*
- $f_3(X_1,X_2,X_3)$ *est le troisième polynôme de sommation de Semaev;*
- Sym_m *représente l'opération consistant à réécrire un polynôme partiellement symétrisé en fonction des polynômes symétriques élémentaires*

$$\begin{cases} e_{1,m} = e_{1,1} + e_{1,m-1} \\ e_{2,m} = e_{1,1}\,e_{1,m-1} + e_{2,m-1} \\ \qquad\vdots \\ e_{m-1,m} = e_{1,1}\,e_{m-2,m} + e_{m-1,m-1} \\ e_{m,m} = e_{1,1}\,e_{m-1,m-1}. \end{cases}$$

On remarque alors que le degré total de $\tilde{f}_{m+1}(e_{1,m},\ldots,e_{m,m},X_{m+1})$ en les variables de $e_{1,m},\ldots,e_{m,m}$ est 2^{m-1}, qui correspond aussi au degré en X_{m+1}. Il est bien sûr également possible de définir $\tilde{f}_m$ à partir des résultants de $\tilde{f}_{m-j}$ et $\tilde{f}_{j+2}$ (pour tout $j \in [\![1; m-3]\!]$) comme c'est le cas dans l'approche originale de Semaev. Cela permet de réduire le nombre de calculs de résultants, au coût d'une symétrisation plus compliquée à chaque étape. Dans le contexte présenté en section suivante, où l'on ne considérera que les m-ièmes polynômes de sommation pour $m \leq 6$, l'approche donnée en proposition 7.1.4 est expérimentalement la plus rapide.

Un exemple de calcul

On donne ici un exemple de calcul du cinquième polynôme de sommation de Semaev symétrisé pour une courbe elliptique définie $\mathbb{F}_{p^5}$ où p est premier, réalisé avec le logiciel Magma V2.17-5 sur un ordinateur Intel Core 2 Duo à 2.6 GHz. On compare les temps obtenus pour le calcul classique utilisant les résultants suivi d'une symétrisation finale, et pour le calcul avec les deux précédentes méthodes. On donne également l'usage mémoire pour l'exemple de calcul dans le cas d'une courbe elliptique de 160 bits.

$\log_2(p)$	résultant et symétrisation	1ère méthode	2ème méthode
8	$1.54 + 10.45 = 11.99$ s	1.75 s	1.04 s
16	$1.58 + 10.63 = 12.21$ s	1.77 s	1.04 s
32	$10.23 + 23.16 = 33.39$ s	11.12 s	3.57 s
usage mémoire	510 Mo	22 Mo	66 Mo

FIGURE 7.1 – Comparaison des calculs du cinquième polynôme de sommation symétrisé avec la méthode classique et les deux méthodes proposées

Autant le cinquième polynôme de sommation est très rapide à calculer, autant le sixième est difficile à obtenir. En particulier, aucune des trois méthodes données ne permet d'en faire le calcul : l'usage mémoire atteint rapidement la capacité maximale de notre machine personnelle (environ 3 Go). Par contre, il reste possible avec notre méthode de symétrisation partielle des résultants de calculer le sixième polynôme de sommation pour une courbe définie sur $\mathbb{F}_{p^6}$, en fonction de

$e_{1,5}, \ldots, e_{5,5}$ lorsque x_6 est fixé : par exemple, pour une taille de courbe de 162 bits, on obtient le calcul en environ 10 min avec un usage mémoire de 60 Mo.

Cas de la caractéristique 2

Ainsi que l'explique Granger dans [Gra10], les polynômes de sommation sont beaucoup plus creux en caractéristique 2, et sont donc plus rapides à calculer. Par exemple, le cinquième polynôme de sommation partiellement symétrisé a seulement 100 termes et son calcul nécessite à peine 50 ms sur $\mathbb{F}_{(2^{31})^5}$, alors qu'en caractéristique impaire, le même polynôme possède 3 972 termes et s'obtient en un peu moins de 4 s sur $\mathbb{F}_{p^5}$ avec p premier de taille similaire.

7.2 La méthode de Gaudry et Diem

La principale difficulté dans l'approche de Semaev réside dans le choix d'une base de factorisation adaptée à la recherche de relations. Gaudry [Gau08] et Diem [Die11] (indépendamment) pallient ce problème en proposant une nouvelle méthode de calculs d'indices, permettant d'attaquer des courbes définies sur une extension d'un corps fini $\mathbb{F}_{q^n}$ de petit degré. L'idée consiste à utiliser les outils de la restriction de Weil pour obtenir la base adéquate et ramener la recherche de relations à la résolution d'un système polynomial multivarié sur $\mathbb{F}_q$. Bien que cette méthode de calcul d'indices s'applique théoriquement à toutes les variétés abéliennes de petite dimension, on présente la méthode uniquement dans le contexte d'attaques du DLP défini sur des jacobiennes de courbes hyperelliptiques de petit genre. Un intérêt particulier est porté au cas des courbes elliptiques, dans lequel les polynômes de sommation de Semaev jouent un rôle essentiel. On expose ensuite l'approche de Nagao permettant de généraliser l'idée de Semaev dans le cas des courbes de genre plus grand que 1. La complexité théorique est analysée pour les deux approches, que l'on compare finalement à la technique GHS.

Dans ce qui suit, $\mathcal{H}$ désigne une courbe hyperelliptique définie sur $\mathbb{F}_{q^n}$ de genre g admettant un modèle imaginaire

$$\mathcal{H} : \; y^2 + h_0(x)y = h_1(x) \quad \text{où } \deg(h_1) = 2g+1, \; \deg(h_0) \leq g. \tag{7.3}$$

On note $\mathcal{O}_{\mathcal{H}}$ son point à l'infini et $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$ sa jacobienne.

7.2.1 Description générale de l'attaque

Pour définir la base de factorisation, on considère l'ensemble des éléments de $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$ de la forme $D_Q \sim (Q) - (\mathcal{O}_{\mathcal{H}})$ où Q est un point de $\mathcal{H}(\mathbb{F}_{q^n})$ d'abscisse dans $\mathbb{F}_q$. Étant donnée l'involution ι agissant naturellement sur $\mathcal{H}$, il est possible de diminuer de moitié la taille de cet ensemble et de prendre pour base de factorisation $\mathcal{F}$ un ensemble de représentants de

$$\{D_Q \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n}) \; : \; D_Q \sim (Q) - (\mathcal{O}_{\mathcal{H}}), \; Q \in \mathcal{H}(\mathbb{F}_{q^n}), \; x(Q) \in \mathbb{F}_q\}/\iota.$$

En particulier, pour une courbe elliptique E définie sur $\mathbb{F}_{q^n}$ admettant une équation de Weierstrass réduite, on introduit l'ensemble des points $\mathbb{F}_{q^n}$-rationnels dont l'abscisse est dans $\mathbb{F}_q$; un choix naturel de base de factorisation lorsque $q = 3 \bmod 4$ est par exemple

$$\mathcal{F} = \{P \in E(\mathbb{F}_{q^n}) \; : \; P = (x_P, y_P), x_P \in \mathbb{F}_q, y_P \in S\},$$

où S est l'ensemble constitué des carrés de $\mathbb{F}_{q^n}$.

Pour calculer le logarithme discret d'un diviseur $D \in \langle D_0 \rangle$ où $D_0 \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$ est d'ordre grand premier, on doit d'abord trouver des relations en décomposant des combinaisons de la forme $R = [a]D_0$ ou $R = [a]D_0 + [b]D$, où a et b sont des entiers aléatoires, en sommes de points de $\mathcal{F}$. Suivant l'idée de Semaev, Gaudry suggère de considérer seulement les relations de la forme

$$R \sim \sum_{i=1}^{ng} ((Q_i) - (\mathcal{O}_{\mathcal{H}})) \text{ avec } x_{Q_i} \in \mathbb{F}_q \text{ pour tout } i \in [\![1; n]\!], \tag{7.4}$$

où ng est la dimension de la variété abélienne obtenue en considérant la restriction de Weil $\mathcal{W} = W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n}))$. En utilisant la représentation de Mumford des éléments de $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$, l'équation 7.4 peut s'écrire polynomialement sous la forme :

$$R \sim \sum_{i=1}^{ng}(x - x_i, y_i) \sim \left(x^g + \sum_{k=0}^{g-1} f_k(x_1, \ldots, x_{ng}, y_1, \ldots, y_{ng}) x^k,\ \sum_{k=0}^{g-1} g_k(x_1, \ldots, x_{ng}, y_1, \ldots, y_{ng}) x^k \right),$$

où f_k, g_k sont des fractions rationnelles en les coordonnées (x_i, y_i) des points Q_i pour $1 \leq i \leq ng$. En identifiant les coefficients dans la représentation de Mumford de R avec ceux de $\sum_{i=1}^{ng}(x - x_i, y_i)$, on obtient alors un système défini sur $\mathbb{F}_{q^n}$ en $2ng$ variables composé de $2g$ équations provenant de l'identification et des ng équations liant x_i et y_i. On procède ensuite à la restriction de Weil : on considère $\mathbb{F}_{q^n}$ comme $\mathbb{F}_q[t]/(f(t))$ où f est un polynôme irréductible de degré n sur $\mathbb{F}_q$, de façon à représenter chaque point $P \in \mathcal{H}(\mathbb{F}_{q^n})$ par $2n$ coordonnées $x_{j,P}, y_{j,P} \in \mathbb{F}_q$ telles que $x_P = x_{0,P} + x_{1,P}t + \ldots + x_{n-1,P}t^{n-1}$ et $y_P = y_{0,P} + y_{1,P}t + \ldots + y_{n-1,P}t^{n-1}$. Si l'on prend en compte le fait que les points Q_i ont tous leurs abscisses dans $\mathbb{F}_q$, on obtient ainsi un nouveau système défini sur $\mathbb{F}_q$ en $(n+1)ng$ variables et $(n+2)ng$ équations dont les solutions donnent précisément la décomposition de R. Il est à noter cependant qu'en pratique, le système ainsi construit est bien souvent trop compliqué pour pouvoir être résolu. On verra néanmoins qu'il est possible dans le cas particulier des courbes elliptiques de simplifier considérablement (grâce aux polynômes de Semaev) les expressions polynomiales obtenues, rendant possible le calcul de décompositions pour des petites valeurs de n. De même, dans le cas des courbes de genre $g > 1$, l'approche de Nagao permettra d'obtenir des décompositions lorsque le genre reste petit.

Enfin, une fois que l'on a collecté suffisamment de relations indépendantes de la forme (7.4) (au moins autant que la cardinalité de $\mathcal{F}$), on procède à la phase d'algèbre linéaire classique décrite en section 4.3 pour en déduire le logarithme de D en base D_0.

On note que cette méthode de calcul d'indices par décomposition est un cas particulier dans le contexte plus général des variétés abéliennes de petite dimension [Gau08]. Soit $\mathcal{A}$ une variété abélienne définie sur $\mathbb{F}_q$ de dimension n ; dans ce qui précède, $\mathcal{A}$ est égale à la restriction de Weil $W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n}))$. On choisit d'abord une "bonne" représentation de $\mathcal{A}$, i.e une carte affine $U \subset \mathcal{A}$ (autrement dit un ouvert dense) dans laquelle on dispose d'un système de coordonnées $(x_1, \ldots, x_n, y_1, \ldots, y_m)$ tel que le corps de fonctions $\mathbb{F}_q(\mathcal{A})$ soit une extension algébrique du corps des fractions rationnelles $\mathbb{F}_q(x_1, \ldots, x_n)$. On définit ensuite la base de factorisation de façon similaire aux attaques par descente de Weil, en considérant l'intersection de $\mathcal{A}$ avec les $n-1$ hyperplans suivants : $\mathcal{F} = \{P \in U : x_2(P) = \ldots = x_n(P) = 0\}$. Cette variété de dimension 1 est généralement irréductible et possède de l'ordre de q points (voir [Gau08] pour plus de détails). On cherche alors des décompositions de points de $\mathcal{A}$ comme somme de n points de $\mathcal{F}$ en utilisant la loi de groupe définie sur $\mathcal{A}$: on ramène ainsi chaque test de décomposition à la recherche de racines $\mathbb{F}_q$-rationnelles d'un système polynomial multivarié. Comme précédemment, on déduit enfin des relations obtenues les logarithmes discrets cherchés à l'aide de techniques d'algèbre linéaire creuse.

Remarque 7.2.1. *Avec cette présentation générale dans le contexte des variétés abéliennes de petite dimension, on retrouve à la fois la technique de calculs d'indice par décomposition présentée ci-dessus pour les courbes (hyper-)elliptiques définies sur $\mathbb{F}_{q^n}$, mais également la méthode de calcul d'indices classique existant sur les courbes hyperelliptiques de petit genre $g \geq 3$ proposée dans [Gau00] et détaillée en section 5.4.2. En effet, étant donnée la variété abélienne égale à la jacobienne d'une courbe $\mathcal{H}$ hyperelliptique définie sur $\mathbb{F}_q$ de genre g, on considère la carte affine obtenue en prenant $u_0 = 1$ dans le système de coordonnées $(u_g, \ldots, u_0, v_{g-1}, \ldots, v_0)$ donné par les coefficients des polynômes (u, v) dans la représentation de Mumford des diviseurs. La base de factorisation définie par $\mathcal{F} = \{(u_g x^g + \ldots + u_1 x + 1, v_{g-1} x^{g-1} + \ldots, v_1 x + v_0) : u_2 = \ldots = u_g = 0\}$ correspond alors aux diviseurs admettant un seul point dans leur support. Ainsi, tester si un diviseur (u, v) se décompose en somme de n diviseurs de $\mathcal{F}$ revient à tester si u est scindé sur $\mathbb{F}_q$, ce qui est précisément la méthode originale de [Gau00].*

7.2.2 Cas particulier des courbes elliptiques

Lorsque l'on recherche des relations sur une courbe elliptique définie sur $\mathbb{F}_{q^n}$, on a tout intérêt à utiliser les polynômes de sommation de Semaev pour simplifier les expressions polynomiales obtenues. Ainsi lorsque l'on cherche une relation du type (7.4), autrement dit de la forme

$$R = \pm P_1 \pm P_2 \pm \ldots \pm P_n \quad \text{avec } P_i \in \mathcal{F}, \tag{7.5}$$

on se ramène à l'équation équivalente

$$\tilde{f}_{n+1}(e_1, \ldots, e_n, x_R) = 0 \tag{7.6}$$

où $\tilde{f}_{n+1} \in \mathbb{F}_{q^n}[X_1, \ldots, X_{n+1}]$ est le $(n+1)$-ième polynôme de sommation symétrisé, de degré total 2^{n-1} en les fonctions élémentaires symétriques $e_1, \ldots, e_n$ des variables $x_{P_1}, \ldots, x_{P_n}$. Les inconnues $e_1, \ldots, e_n$ étant dans $\mathbb{F}_q$, on obtient en triant (7.6) suivant les puissances de t une expression de la forme

$$\sum_{i=0}^{n} \varphi_i(e_1, \ldots, e_n)\, t^i = 0,$$

où chacun des polynômes φ_i est à coefficients dans $\mathbb{F}_q$ (dépendants paramétriquement des composantes de x_R). De cette restriction des scalaires, on déduit un système polynomial de degré total 2^{n-1} en n équations et n inconnues

$$\varphi_0(e_1, \ldots, e_n) = 0, \quad \varphi_1(e_1, \ldots, e_n) = 0, \quad \ldots \quad \varphi_{n-1}(e_1, \ldots, e_n) = 0. \tag{7.7}$$

Une fois le $(n+1)$-ième polynôme de Semaev calculé, ce système est donc facile à écrire mais reste par contre difficile à résoudre. Bien que le coût de la résolution soit délicat à estimer, on sait toutefois que la complexité du calcul est au moins polynomiale en le degré de l'idéal zéro-dimensionnel correspondant ; or suivant l'analyse faite dans [Die11], on peut montrer que ce degré est génériquement égal à la borne de Bézout $2^{n(n-1)}$. En particulier, la recherche de relations devient rapidement infaisable lorsque n croît, notamment dès que $n \geq 5$.

Pour passer d'une solution de l'équation (7.6) à une relation de la forme (7.5), on doit déterminer les solutions $\mathbb{F}_q$-rationnelles du polynôme univarié $F(x) = x^n - e_1 x^{n-1} + \ldots + (-1)^n e_n$, ce qui peut se faire aisément en utilisant par exemple l'algorithme présenté en section 1.2.2. Lorsque F est scindé sur $\mathbb{F}_q$, il reste encore à déterminer les points de $\mathcal{F}$ dont l'abscisse est solution de F, ainsi que le signe à apposer devant chacun de ces points pour obtenir une décomposition de R. Le

coût de cette "désymétrisation" reste négligeable comparé au coût de la résolution des systèmes polynomiaux considérés. On note cependant que lorsque F est scindé sur $\mathbb{F}_q$, il est possible a priori que certaines racines ne correspondent pas aux abscisses de points de $\mathcal{F}$: c'est le cas notamment lorsque l'ordonnée y est dans une extension quadratique $\mathbb{F}_{(q^n)^2}$ mais pas dans $\mathbb{F}_{q^n}$. Néanmoins, tous ces points sont dans le sous-groupe $G' = \psi(E'(\mathbb{F}_{q^n})) \subset E(\mathbb{F}_{q^{2n}})$, où $E'_{|\mathbb{F}_{q^n}}$ est la tordue quadratique de E et ψ est un $\mathbb{F}_{q^{2n}}$-isomorphisme entre $E'(\mathbb{F}_{q^{2n}})$ et $E(\mathbb{F}_{q^{2n}})$ (cf. section 5.1.4). La décomposition de R est donc de la forme $R = \pm P_1 \pm \ldots \pm P_k \pm P_{k+1} \pm \ldots \pm P_n$ avec $P_i \in \mathcal{F}$ si $i \leq k$ et $P_i \in G'$ si $i > k$; autrement dit on a $R \mp P_1 \mp \ldots \mp P_k = \pm P_{k+1} \pm \ldots \pm P_n$ où le terme de gauche est dans $E(\mathbb{F}_{q^n})$ et le terme de droite est dans G'. Comme l'intersection de ces deux groupes est réduite à $E(\mathbb{F}_{q^n})[2]$, on obtient en particulier une décomposition en seulement $n-k$ points d'un point de 2-torsion de $E(\mathbb{F}_{q^n})$. Bien que l'existence d'une telle décomposition soit possible, elle reste néanmoins très improbable ; de fait, lorsque F est scindé sur $\mathbb{F}_q$, on obtiendra en pratique toujours une décomposition dans $\mathcal{F}$. On note que cette analyse est valable aussi bien en caractéristique impaire qu'en caractéristique 2.

Analyse et complexité

Avant d'analyser la méthode, on s'assure que la base de factorisation $\mathcal{F}$ contient suffisamment de points. D'un point de vue heuristique, il est assez clair que sa cardinalité est approximativement $q/2$, on justifie ici rigoureusement cette analyse. L'objet géométrique correspondant à $\mathcal{F}$ est la variété projective $\mathcal{C} = \{P \in E(\mathbb{F}_{q^n}) : P = (x_P, y_P),\ x_P \in \mathbb{F}_q\} \cup \{\mathcal{O}_E\}$, contenue dans la restriction de Weil $\mathcal{W} = W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$ de E relativement à l'extension $\mathbb{F}_{q^n}/\mathbb{F}_q$; il s'agit précisément de l'intersection de $\mathcal{W}$ par des hyperplans que l'on considère dans l'attaque GHS (voir section 6.2.1). Il est relativement aisé de déterminer que $\mathcal{C}$ est une courbe (i.e. est irréductible) quand le nombre magique m est égal à n, et de borner son genre grâce à la formule (6.4) ou (6.6) ; on pourra alors donner une estimée de sa cardinalité grâce à la borne de Hasse. Suivant l'approche de Diem, on va supposer que la courbe elliptique E satisfait la condition suivante, qui est une reformulation de la condition 2.7 de [Die11] et implique directement que $m = n$:

Condition 7.2.2. *Soit σ la fonction d'exponentiation par q. Il existe un point $P \in E(\overline{\mathbb{F}_{q^n}})$ de 2-torsion tel que pour tout $1 \leq i \leq n$, $\sigma^i(x_P) \neq x_P$ et $\sigma^i(x_P)$ n'est pas l'abscisse d'un point de 2-torsion de E.*

Rigoureusement parlant, cette condition porte sur l'équation de Weierstrass de E et non sur la courbe elle-même. Diem montre qu'il est toujours possible de trouver une équation de E pour laquelle cette condition est satisfaite, et on suppose dans la suite que l'équation de E choisie vérifie la condition 7.2.2. Le nombre de points de $\mathcal{C}$ est alors plus grand que $q + 1 - n2^{n+2}(\sqrt{q} - 1)$; dès que $n \leq c \log_2 q$ où $c < 1/2$, ce nombre est plus grand que $q/2$ pour q suffisamment grand. On a donc la proposition suivante :

Proposition 7.2.3. *Pour tout $\epsilon > 0$, il existe une constante $C > 0$ telle que pour tout $q > C$ et tout $n < (1/2 - \epsilon) \log_2 q$, la base de factorisation $\mathcal{F}$ associée à une courbe elliptique définie sur $\mathbb{F}_{q^n}$ satisfaisant la condition 7.2.2 contient plus de $q/4$ éléments.*

On note cependant que même lorsque $n > \log_2(q)/2$, la base de factorisation $\mathcal{F}$ peut avoir un nombre de points de l'ordre de q. En effet, les valeurs possibles pour la cardinalité d'une courbe irréductible définie sur $\mathbb{F}_q$ étant principalement concentrées autour de $q+1$, il paraît peu probable qu'aucune des équations représentant une courbe elliptique $E_{|\mathbb{F}_{q^n}}$ donnée ne procure une base de factorisation admettant suffisamment d'éléments.

Durant la première étape de la méthode de Gaudry et Diem, on doit collecter de l'ordre de $\#\mathcal{F} \simeq q/2$ relations de la forme (7.5). Le $(n+1)$-ième polynôme de sommation peut être calculé une fois pour toute en $\mathrm{Poly}(e^{(n+1)^2} \log_2 q)$ opérations [Die11], et sa symétrisation s'obtient facilement en calculant des bases de Gröbner pour un ordre d'élimination (voir section 7.1.2). La probabilité de décomposition d'un point $R \in E(\mathbb{F}_{q^n})$ est heuristiquement en

$$\frac{\#\mathcal{C}^n/\mathfrak{S}_n}{\#E(\mathbb{F}_{q^n})} \simeq \frac{q^n}{n!}\frac{1}{q^n} = \frac{1}{n!};$$

cette approximation peut être justifiée rigoureusement en utilisant la théorie de l'intersection, voir encore [Die11]. Le coût du test de décomposition d'un point R en somme de n points de la base de factorisation, noté $c(n,q)$, correspond au coût de la résolution d'un système polynomial en n équations et n variables, défini sur $\mathbb{F}_q$ et de degré total 2^{n-1}. Comme on a besoin d'au moins $\#\mathcal{F}$ relations, le coût total de la phase de recherche de relations est

$$n!\, c(n,q) \#\mathcal{F} \simeq n!\, c(n,q)\, q/2.$$

L'estimation de $c(n,q)$ n'est pas évidente dans la mesure où le coût de la résolution du système polynomial dépend de la méthode utilisée. La technique standard consiste à calculer une base de Gröbner pour l'ordre lexicographique de l'idéal correspondant. On rappelle (voir section 1.2.1) que la base de Gröbner attendue pour l'ordre lexicographique est de la forme :

$$\{X_1 - g_1(X_n), \ldots, X_{n-1} - g_{n-1}(X_n), g_n(X_n)\},$$

où g_n est un polynôme univarié de degré égal au degré de l'idéal, et $g_1, \ldots, g_{n-1}$ sont des polynômes univariés de degré strictement inférieur. Il devient alors facile de déterminer les solutions du système correspondant en utilisant par exemple l'algorithme donné en section 1.2.2. Comme expliqué en section 2.5, plutôt que de faire un calcul direct très coûteux de la base de Gröbner pour l'ordre lexicographique, une stratégie efficace consiste à d'abord calculer une base de Gröbner pour l'ordre grevlex puis à faire un changement d'ordre. Dans la mesure où le nombre de systèmes à résoudre de la forme (7.7) est important (de l'ordre de $n!\, q/2$) et où les idéaux correspondants sont génériquement zéro-dimensionnels, on a tout intérêt à utiliser la variante de F4 présentée en chapitre 3 pour faire le calcul pour l'ordre grevlex, puis à appliquer l'algorithme de changement d'ordre FGLM. Pour déterminer une borne supérieure sur la complexité du calcul de la base de Gröbner pour l'ordre grevlex du système zéro-dimensionnel $\{\varphi_0, \ldots, \varphi_{n-1}\}$, on peut utiliser les estimées données en section 3.2.6. Si l'on fait l'hypothèse raisonnable que le système est régulier, le degré de régularité du système homogénéisé est plus petit que la borne de Macaulay $d = \sum_{i=0}^{n-1}(\deg \varphi_i - 1) + 1 = n2^{n-1} - n + 1$, les polynômes φ_i étant tous de degré 2^{n-1}. La complexité du calcul avec la variante F4 est ainsi majorée par $\tilde{O}\left(\binom{n2^{n-1}+1}{n}^{\omega}\right)$, où ω est l'exposant intervenant dans la complexité effective du produit matriciel. Comme n est négligeable comparé à $n2^{n-1}+1$, en utilisant la formule de Stirling, on obtient :

$$\binom{n2^{n-1}+1}{n} \sim \frac{(n2^{n-1})^n}{n!} \sim 2^{n(n-1)} e^n\, (2\pi n)^{-1/2}.$$

La complexité du calcul de la base de Gröbner pour l'ordre grevlex est donc $\tilde{O}\left(\left(2^{n(n-1)} e^n\, n^{-1/2}\right)^{\omega}\right)$. En utilisant les estimées précises de la complexité de FGLM données en section 2.5.2 et le fait que les idéaux des systèmes considérés sont génériquement de degré $2^{n(n-1)}$, on obtient une complexité pour le changement d'ordre en $\tilde{O}(n2^{3n(n-1)})$, qui reste le coût dominant du calcul de la base de Gröbner pour l'ordre lex lorsque $\omega < 3$. En particulier, on a l'estimation suivante

$$c(n,q) = \tilde{O}(n2^{3n(n-1)}).$$

La deuxième étape de résolution des logarithmes discrets peut se faire avec des techniques d'algèbre linéaire creuse (voir section 4.3.1) ; la complexité du calcul est alors en $\tilde{O}(nq^2)$ opérations dans $\mathbb{Z}/r\mathbb{Z}$, où r est l'ordre du point P base du logarithme discret. Cette phase est donc dominante d'un point de vue temps de calcul à n fixé et $q \to \infty$.

Afin d'améliorer la complexité asymptotique de l'algorithme, Gaudry propose d'équilibrer le coût de la construction de la matrice avec celui de l'algèbre linéaire, en utilisant les techniques "large primes" présentées en section 4.3.2 et 5.4.2 : la taille asymptotiquement optimale pour la base des petits diviseurs est en $q^{1-1/n}$. On doit alors obtenir de l'ordre de $q^{2-2/n}$ relations au lieu de q, et le coût de la première étape de l'algorithme devient $n!\, c(n,q)\, q^{2-2/n}$. Par opposition, le coût de l'algèbre linéaire se réduit ainsi à $\tilde{O}(n\, q^{2-2/n})$, ce qui devient négligeable par rapport au coût de l'étape précédente. Finalement, le coût total de la résolution du DLP sur une courbe elliptique définie sur $\mathbb{F}_{q^n}$, à n fixé, est en $\tilde{O}(q^{2-2/n})$ lorsque $q \to \infty$. Dès que $n > 2$, cette attaque est donc asymptotiquement plus performante que les attaques génériques dont la complexité est en $\tilde{O}(q^{n/2})$. On note cependant que la constante cachée croît de façon super-exponentielle avec n. Une estimée complète de la complexité obtenue est donnée par

$$\tilde{O}(n!\, 2^{3n(n-1)} q^{2-2/n}).$$

On remarque aussi que bien que la parallélisation de la phase de recherche de relations soit immédiate, elle est beaucoup plus difficile à mettre en place pour l'algèbre linéaire. En particulier, le choix optimal de la taille de la base de factorisation devrait dépendre en pratique non seulement de l'implantation choisie mais aussi de la puissance de calcul disponible.

7.2.3 L'approche de Nagao en genre $g > 1$

Il n'existe pas d'équivalent des polynômes de Semaev lorsque l'on souhaite obtenir des décompositions en un nombre fixe de points sur une courbe de genre $g > 1$. On peut néanmoins simplifier considérablement les expressions algébriques obtenues dans la méthode de Gaudry et Diem à partir de la loi de groupe, grâce à l'approche de Nagao [Nag10]. Par exemple dans le cas hyperelliptique, la technique de Nagao permet de ramener la recherche d'une décomposition à la résolution d'un système polynomial multivarié quadratique. Bien que moins efficace que la méthode de Semaev dans le cas elliptique, cette approche sera néanmoins la plus simple dans les autres cas.

On considère comme précédemment la courbe hyperelliptique $\mathcal{H}$, les deux diviseurs $D, D_0 \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$ intervenant dans le DLP considéré, ainsi que la base de factorisation $\mathcal{F}$ définie par Gaudry. L'objectif est de décomposer un diviseur R donné (provenant typiquement de combinaisons aléatoires de D_0 et D) en une somme de ng diviseurs de $\mathcal{F}$. On considère pour cela le $\mathbb{F}_{q^n}$-espace vectoriel de Riemann-Roch :

$$\mathcal{L}\left(ng(\mathcal{O}_{\mathcal{H}}) - R\right) = \{f \in \mathbb{F}_{q^n}(\mathcal{H})^* : div(f) \geq R - ng(\mathcal{O}_{\mathcal{H}})\} \cup \{0\},$$

voir section 5.1.2. On a alors l'équivalence

$$R + \sum_{i=1}^{ng} ((Q_i) - (\mathcal{O}_{\mathcal{H}})) = div(f) \Leftrightarrow f \in \mathcal{L}\left(ng(\mathcal{O}_{\mathcal{H}}) - R\right).$$

Trouver les ng diviseurs intervenant dans la décomposition de R revient donc à déterminer une fonction f de $\mathcal{L}\left(ng(\mathcal{O}_{\mathcal{H}}) - R\right)$ ainsi que les ng points de $\mathcal{H}$ en dehors du support de R en lesquels f s'annule. On introduit à cet effet une base du $\mathbb{F}_{q^n}$-espace vectoriel $\mathcal{L}\left(ng(\mathcal{O}_{\mathcal{H}}) - R\right)$ de dimension

$\ell+1$ où $\ell=(n-1)g$. Si le diviseur R admet une représentation de Mumford (u,v) où $\deg u = g$, une $\mathbb{F}_{q^n}$-base est donnée par

$$\mathcal{B} = \{u, xu, \ldots, x^{m_1}u, y-v, x(y-v), \ldots, x^{m_2}(y-v)\},$$

où[2] $m_1 = \lfloor (n-1)g/2 \rfloor$ et $m_2 = \lfloor ((n-1)g-1)/2 \rfloor$. En particulier, une fonction $f \in \mathbb{F}_{q^n}(\mathcal{H})^*$ de la forme

$$f(x,y) = u\sum_{i=0}^{m_1} \lambda_{2i+1}x^i + (y-v)\sum_{i=0}^{m_2}\lambda_{2i+2}x^i$$

va s'annuler en le support de R et exactement ng autres points $Q_1,\ldots,Q_{ng}$ de $\mathcal{H}(\overline{\mathbb{F}_q})$ si son coefficient dominant n'est pas zéro. On cherche une condition sur les coordonnées $\lambda_i \in \mathbb{F}_{q^n}$ de f dans $\mathcal{B}$, assurant que tous ces points sont à abscisses dans le corps de base $\mathbb{F}_q$. Pour cela, on introduit le polynôme F (déduit de f) qui admet pour racines exactement les abscisses des Q_i. Comme $\mathcal{H}$ est hyperelliptique, $F(x)$ s'obtient simplement à partir de $f(x,y)\cdot f(x,-y-h_0(x))/u(x)$ où l'on a remplacé $y(y+h_0(x))$ par $h_1(x)$ (avec h_0 et h_1 donnés par l'équation (7.3) de $\mathcal{H}$). Quitte à normaliser F au point à l'infini (i.e. à poser $\lambda_{\ell+1}=1$), on obtient une expression de la forme

$$F_{\lambda_1,\ldots,\lambda_\ell}(x) = x^{ng} + \sum_{i=0}^{ng-1} c_i(\lambda_1,\ldots,\lambda_\ell)x^i,$$

où les c_i sont des polynômes quadratiques en les variables $\lambda_1,\ldots,\lambda_\ell$. Ses ng racines sont alors exactement les abscisses des zéros de f qui ne sont pas dans le support de R. L'objectif est donc de déterminer les valeurs de $\lambda_1,\ldots,\lambda_\ell$ pour lesquelles F est scindé sur $\mathbb{F}_q$.

Une première condition évidente est que les coefficients c_i de F soient tous dans $\mathbb{F}_q$. Après une restriction des scalaires sur ces coefficients :

$$c_i(\lambda_1,\ldots,\lambda_\ell) = \sum_{j=0}^{n-1} c_{i,j}(\lambda_{1,0},\ldots,\lambda_{\ell,n-1})t^j,$$

on déduit un système quadratique polynomial

$$c_{i,j}(\lambda_{1,0},\ldots,\lambda_{\ell,n-1}) = 0,\ 0 \le i \le ng-1,\ 1 \le j \le n-1$$

composé de $(n-1)ng$ d'équations et $(n-1)ng$ variables provenant des composantes $\lambda_{1,0},\ldots,\lambda_{\ell,n-1}$ de $\lambda_1,\ldots,\lambda_\ell$ sur $\mathbb{F}_q$. Heuristiquement, l'idéal correspondant est alors génériquement de dimension 0, et on peut résoudre le système en calculant comme précédemment une base de Gröbner pour l'ordre lexicographique.

Une fois les solutions en $\lambda_1,\ldots,\lambda_\ell$ obtenues, il reste encore à voir si le polynôme $F \in \mathbb{F}_q[x]$ est bien scindé sur $\mathbb{F}_q$; lorsque c'est le cas, on obtient les abscisses des points dans la décomposition de R. Contrairement au cas des décompositions sur courbes elliptiques avec Semaev, on note que les points correspondant à ces abscisses sont nécessairement dans $\mathcal{H}(\mathbb{F}_{q^n})$ et non dans une extension quadratique ; on peut effet retrouver directement la valeur de y correspondante en résolvant l'équation linéaire $f(x,y)=0$ où x a été évalué.

Exemple 7.2.4. *Soit $\mathcal{H}$ la courbe hyperelliptique de genre 2 définie sur $\mathbb{F}_{67^2} = \mathbb{F}_{67}[t]/(t^2-2)$ par l'équation :*

$$\mathcal{H} : y^2 = h_1(x) = x^5 + (50t+66)x^4 + (40t+22)x^3 + (65t+23)x^2 + (61t+3)x + 43t + 6.$$

2. Dans le cas plus général où $1 \le \deg u \le g$, prendre $m_1 = \lfloor (ng - \deg u)/2 \rfloor$ et $m_2 = \lfloor ((n-2)g + \deg u - 1)/2 \rfloor$.

Afin de décomposer le diviseur

$$R = \big(u(x), v(x)\big) = \big(x^2 + (52t+3)x + 21t + 2,\ (22t+41)x + 25t + 42\big) \in Jac_{\mathcal{H}}(\mathbb{F}_{67^2})$$

en somme de $ng = 4$ *diviseurs de la forme* $(Q) - (\mathcal{O}_{\mathcal{H}})$ *où* $x_Q \in \mathbb{F}_{67}$*, on considère les fonctions* $f \in \mathcal{L}(4(\mathcal{O}_{\mathcal{H}}) - R) = \langle u(x), y - v(x), x\,u(x)\rangle$ *et* $F_{\lambda_1,\lambda_2}(x) = f(x,y)f(x,-y)/u(x)$*. Le polynôme* F_{λ_1,λ_2} *est alors de la forme*

$$F_{\lambda_1,\lambda_2}(x) = (x+\lambda_2)^2 u(x) - 2\lambda_1(x+\lambda_2)v(x) + \frac{v^2(x) - h_1(x)}{u(x)}\lambda_1^2 = x^4 + (-\lambda_1^2 + 2\lambda_2 + 52t + 3)\,x^3 + \dots$$

donc unitaire de degré 4 *en* x *et quadratique en* λ_1, λ_2*. On cherche les valeurs de* $\lambda_1, \lambda_2 \in \mathbb{F}_{67^2}$ *telles que* $F_{\lambda_1,\lambda_2}(x)$ *soit à coefficients dans le corps de base* $\mathbb{F}_{67}$*, autrement dit telles que*

$$\begin{cases} (-\lambda_1^2 + 2\lambda_2 + 52t + 3) \in \mathbb{F}_{67} \\ (2t+4)\lambda_1^2 + (23t+52)\lambda_1 + \lambda_2^2 + (37t+6)\lambda_2 + 21t + 2 \in \mathbb{F}_{67} \\ (35t+28)\lambda_1^2 + (23t+52)\lambda_1\lambda_2 + (17t+50)\lambda_1 + (52t+3)\lambda_2^2 + (42t+4)\lambda_2 \in \mathbb{F}_{67} \\ (23t+16)\lambda_1^2 + (17t+50)\lambda_1\lambda_2 + (21t+2)\lambda_2^2 \in \mathbb{F}_{67} \end{cases}$$

Pour trouver les solutions en λ_1 *et* λ_2 *de ce système, on fait une restriction de Weil en introduisant les variables* $\lambda_1 = \lambda_{1,0} + t\lambda_{1,1}$ *et* $\lambda_2 = \lambda_{2,0} + t\lambda_{2,1}$*, on obtient alors le système quadratique composé de* 4 *équations et* 4 *variables :*

$$\begin{cases} -2\lambda_{1,0}\lambda_{1,1} + 2\lambda_{2,1} - 15 = 0 \\ 2\lambda_{1,0}^2 + 8\lambda_{1,0}\lambda_{1,1} + 23\lambda_{1,0} + 4\lambda_{1,1}^2 - 15\lambda_{1,1} + 2\lambda_{2,0}\lambda_{2,1} + 37\lambda_{2,0} + 6\lambda_{2,1} + 21 = 0 \\ 35\lambda_{1,0}^2 - 11\lambda_{1,0}\lambda_{1,1} + 23\lambda_{1,0}\lambda_{2,0} - 15\lambda_{1,0}\lambda_{2,1} + 17\lambda_{1,0} + 3\lambda_{1,1}^2 - 15\lambda_{1,1}\lambda_{2,0} - 21\lambda_{1,1}\lambda_{2,1} - 17\lambda_{1,1} - 15\lambda_{2,0}^2 \\ \qquad\qquad +6\lambda_{2,0}\lambda_{2,1} - 25\lambda_{2,0} + 37\lambda_{2,1}^2 + 4\lambda_{2,1} = 0 \\ 23\lambda_{1,0}^2 + 32\lambda_{1,0}\lambda_{1,1} + 17\lambda_{1,0}\lambda_{2,0} - 17\lambda_{1,0}\lambda_{2,1} - 21\lambda_{1,1}^2 - 17\lambda_{1,1}\lambda_{2,0} + 34\lambda_{1,1}\lambda_{2,1} + 21\lambda_{2,0}^2 + 4\lambda_{2,0}\lambda_{2,1} - 25\lambda_{2,1}^2 = 0 \end{cases}$$

Avec un calcul de base de Gröbner pour l'ordre lex, on déduit les deux couples de solutions $(\lambda_1, \lambda_2) = (7+40t, 8+53t)$ *et* $(\lambda_1, \lambda_2) = (55+37t, 52-t)$*. Pour chacune de ces solutions, il faut encore vérifier si* $F_{\lambda_1,\lambda_2}(x)$ *est scindé dans* $\mathbb{F}_{67}$ *:*

- *pour* $(\lambda_1, \lambda_2) = (7 + 40t, 8 + 53t)$*, on trouve* $F_{\lambda_1,\lambda_2}(x) = x^4 + 53x^3 + 26x^2 + 44x + 12 = (x^2 + 54x + 1)(x^2 - x + 12)$*, qui ne donne pas de décomposition ;*
- *pour* $(\lambda_1, \lambda_2) = (55 + 37t, 52 - t)$*, on trouve* $F_{\lambda_1,\lambda_2}(x) = (x-23)(x-34)(x-51)(x-54)$ *qui donne la décomposition* $R = (Q_1) + (Q_2) + (Q_3) + (Q_4) - 4(\mathcal{O}_{\mathcal{H}})$ *où*

$$Q_1 = \left|\begin{matrix} 23 \\ 23t+12 \end{matrix}\right| \quad Q_2 = \left|\begin{matrix} 34 \\ 10t+43 \end{matrix}\right| \quad Q_3 = \left|\begin{matrix} 51 \\ 17t+3 \end{matrix}\right| \quad Q_4 = \left|\begin{matrix} 54 \\ 23t+15 \end{matrix}\right| .$$

Dans le cas où la courbe considérée est non-hyperelliptique, on peut utiliser de la même façon l'approche de Nagao pour trouver des relations. Étant donné un diviseur D à décomposer et un point distingué $\mathcal{O}$ sur la courbe $\mathcal{C}$ non-hyperelliptique définie sur $\mathbb{F}_{q^n}$ par l'équation $g(x,y) = 0$:

1. on définit la base de factorisation par $\mathcal{F} = \{(Q) - (\mathcal{O}) \ :\ x(Q) \in \mathbb{F}_q\}$;
2. on calcule une base de $\mathcal{L}(ng(\mathcal{O}) - D))$ (en utilisant par exemple l'algorithme proposé dans [Hes02]), puis on exprime $f(x,y)$ (telle que $div(f) = D + \sum_{i=1}^{ng}((Q_i) - (\mathcal{O}))$) dans cette base ;
3. on calcule le résultant $F(x) = Res_y(f(x,y), g(x,y))$;
4. avec une restriction des scalaires, on ramène le calcul de la décomposition de D à un système polynomial carré en $(n-1)ng$ équations et variables défini sur $\mathbb{F}_q$ (qui n'est plus quadratique en les λ_i).

Le système polynomial obtenu est alors de degré total strictement supérieur à 2, ce qui rend les calculs de bases de Gröbner plus compliqués. Par exemple, pour une courbe non-hyperelliptique de genre 3, l'équation de la courbe étant celle d'une quartique, on obtient des systèmes polynomiaux de degré total 4, ce qui rend le calcul impraticable dès que $n \geq 2$.

Analyse et complexité

Pour estimer la complexité de cette méthode, il faut comme dans le cas elliptique commencer par borner la cardinalité de la base de factorisation ainsi que la probabilité qu'un diviseur aléatoire se décompose dans la base. L'analyse présentée dans la section précédente peut s'adapter sans trop de difficultés au cas hyperelliptique et permet de montrer que sous certaines hypothèses (cf. condition 7.2.2 et proposition 7.2.3), la base de factorisation va contenir un nombre de points de l'ordre de $q/2$ dès que q est suffisamment grand. La probabilité de décomposition est plus difficile à justifier, et on se contentera de l'analyse heuristique (voir [Nag10, Assumptions 1 & 2]) selon laquelle cette probabilité est de l'ordre de

$$\frac{\#\mathcal{C}^{ng}/\mathfrak{S}_{ng}}{\#\mathcal{H}} \simeq \frac{1}{(ng)!}.$$

Ainsi pour obtenir de l'ordre de $q/2$ relations, on doit résoudre en moyenne $(ng)! \cdot q/2$ systèmes polynomiaux quadratiques en $(n-1)ng$ variables et équations. Comme dans le cas elliptique précédemment traité, on peut estimer que le coût du calcul de la base de Gröbner d'un système zéro-dimensionnel est au moins polynomial en le degré $2^{(n-1)ng}$ de l'idéal correspondant ; la complexité de la phase de recherche de relations est donc a priori en $\tilde{O}(q)$ à n et g fixés. Le coût de la phase d'algèbre linéaire utilisant des techniques "creuses" est donné par les $O(ngq^2)$ opérations dans $\mathbb{Z}/r\mathbb{Z}$ où r est l'ordre du diviseur $D_0 \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n})$ servant de point base dans le problème du logarithme discret considéré. Pour rééquilibrer les deux phases et accélérer les calculs des logarithmes, on peut à nouveau utiliser la technique "double large prime" qui donne une complexité asymptotique à n et g fixés en $\tilde{O}(q^{2-2/ng})$, meilleure que celle des attaques génériques dès que $ng > 2$. Comme dans le cas des courbes elliptiques, la recherche de relations est beaucoup plus facile à paralléliser que l'algèbre linéaire, ce qui doit être pris en compte pour déterminer la taille optimale de la base de factorisation.

Comparaison avec les polynômes de sommation de Semaev dans le cas elliptique

Il est bien sûr possible d'appliquer la méthode de Nagao aux courbes elliptiques. On peut alors comparer les performances de celle-ci avec la méthode de Gaudry utilisant les polynômes de Semaev ; les caractéristiques des systèmes obtenus lorsque l'on applique les deux méthodes à une courbe elliptique E définie sur $\mathbb{F}_{q^n}$ sont les suivantes :

	degré total	nb éq./var.	$\deg(I)$	complexité de résolution
Gaudry + Semaev	2^{n-1}	n	$2^{n(n-1)}$	$\tilde{O}(2^{3n(n-1)})$
Nagao	2	$n(n-1)$	$2^{n(n-1)}$	$\tilde{O}(2^{2\omega n(n-1)})$

Bien que les idéaux zéro-dimensionnels correspondants à ces systèmes soient génériquement de degré $2^{n(n-1)}$ dans les deux cas, le calcul de la résolution des systèmes issus de l'approche de Nagao est plus coûteux : en faisant une analyse de complexité similaire à celle donnée dans la section 7.2.2, on peut estimer cette complexité à $\tilde{O}\left(\binom{2n(n-1)}{n(n-1)}^{\omega}\right) = \tilde{O}\left(2^{2\omega n(n-1)}\right)$.

Ces valeurs ne sont que des bornes supérieures mais reflètent bien la différence de performances entre les deux approches. En pratique, les calculs menés avec Magma (V2.16-2) donnent pour $n = 4$ et $p = 2^{16} + 1$ les temps suivants pour le calcul de la base de Gröbner lexicographique avec les deux méthodes :

	F4 grevlex	FGLM	Total
Gaudry + Semaev	4 s	280 s	284 s
Nagao	211 s	922 s	1133 s

Le fait que le nombre de variables soit trois fois plus grand dans l'approche de Nagao explique que l'algorithme FGLM soit à peu près trois fois plus lent.

La méthode de Nagao n'en reste pas moins intéressante dans le cas elliptique, puisqu'elle permet de calculer efficacement les polynômes de sommation de Semaev symétrisés. En effet, on peut (de façon très similaire à la première méthode proposée en section 7.1.2) calculer le polynôme $F_{\lambda_1,\ldots,\lambda_\ell}(x)$ et identifier ses coefficients avec les polynômes élémentaires symétriques $e_1,\ldots,e_n$ en les variables $x_1,\ldots,x_n$ appartenant à $\mathbb{F}_q$. Avec une restriction des scalaires, on obtient un système composé de n^2 variables $\lambda_{1,0},\ldots,\lambda_{\ell,n-1},e_1,\ldots,e_n$ et $(n-1)n$ équations ; une élimination des variables $\lambda_{1,0},\ldots,\lambda_{\ell,n-1}$ permet alors de déduire le n-ième polynôme de sommation symétrisé. On renvoie à la section 7.1.2 pour plus détails.

7.2.4 Comparaison avec l'attaque GHS

Les attaques par décomposition et GHS utilisent toutes les deux comme point de départ et comme base de factorisation la variété $\mathcal{C}$ de dimension 1 obtenue en intersectant la restriction de Weil $\mathcal{W} = W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$ d'une courbe elliptique E par des sections hyperplanes ; cependant l'utilisation qui en est faite est assez différente. Dans la méthode de Gaudry et Diem, il n'y a pas de transfert et le calcul d'indices s'effectue directement dans $\mathcal{W}$. L'attaque s'applique de façon similaire à toutes les courbes, et sa difficulté dépend essentiellement de n à cause du coût élevé du test de décomposition. Dans la technique GHS, le calcul d'indice a lieu au contraire dans la jacobienne d'une composante $\mathcal{C}'$ irréductible de $\mathcal{C}$. Le paramètre le plus important est alors le genre g de $\mathcal{C}'$, à cause de la probabilité de décomposition qui est en $1/g!$; le test de décomposition est en revanche assez simple. Le genre g dépend très fortement de la courbe de départ E, et l'efficacité de l'attaque GHS varie énormément d'une courbe à l'autre.

Pour une courbe elliptique arbitraire, le genre de $\mathcal{C}'$ est grand (de l'ordre de 2^n) et l'attaque de Gaudry et Diem est plus pertinente. Mais dès que ce genre est petit (de l'ordre de n), quitte à effectuer préalablement une marche d'isogénies, le faible coût des tests de décomposition rend la technique GHS plus performante. Il faut cependant noter qu'en pratique, les méthodes génériques restent souvent les meilleures pour attaquer le DLP sur des groupes de taille accessible aux calculs.

7.3 Contributions

Le facteur sur-exponentiel en n qui apparaît dans la complexité de l'algorithme de Gaudry et Diem est un sérieux handicap pour les applications pratiques. Notamment, dès lors que le degré n de l'extension du corps fini $\mathbb{F}_{q^n}$ sur lequel est définie la courbe elliptique est plus grand que 4, il n'est plus possible d'obtenir des décompositions avec cette approche. Afin d'améliorer cette situation, on propose une variante permettant de diminuer la complexité de la phase de recherche de relations ; on montre en particulier que la méthode de calcul d'indices correspondante est meilleure que les attaques génériques et que la méthode de Gaudry et Diem pour toute une plage de valeurs de q et n. Dans la section suivante, on donne une application directe des méthodes de calcul d'indices données dans ce chapitre à la résolution du problème SDHP assisté d'un oracle. En particulier, on

montre qu'avec notre approche, il est possible de réaliser une attaque complète de ce problème pour la courbe Oakley 'Well Known Group' 3 proposée dans les standards de sécurité IPSEC [IET98]. Enfin dans la dernière partie, on propose une variante de l'approche de Nagao en genre $g > 1$ permettant d'obtenir dans certains cas des relations plus rapidement qu'avec la méthode présentée en section 7.2.3.

7.3.1 Variante $n-1$

Le coût de la recherche d'une décomposition d'un point R obtenu par des combinaisons de la forme $[a]D_0+[b]D$ est directement relié au coût de la résolution d'un système polynomial multivarié, dont le degré dépend du polynôme de sommation utilisé. Ainsi, plutôt que d'essayer de décomposer en somme de n éléments de $\mathcal{F}$ comme dans [Gau08, Die11], on peut diminuer sensiblement la complexité du calcul en ne considérant que des décompositions en somme de $(n-1)$ éléments. De façon évidente, la probabilité d'obtenir une telle décomposition va diminuer proportionnellement à la taille de q, augmentant ainsi le nombre de systèmes à résoudre avant d'obtenir une relation. Cependant, le gain apporté au niveau de la complexité du calcul sera suffisamment important pour rendre l'approche réaliste pour $n = 5$ et $g = 1$. Cette variante présentant surtout un intérêt lorsqu'elle est appliquée aux courbes elliptiques, on se contente de la présenter dans ce contexte.

Pour trouver des relations de la forme

$$[a]P + [b]Q = \pm P_1 \pm \ldots \pm P_{n-1}, \tag{7.8}$$

où a et b sont des entiers aléatoires et les points P_i sont dans $\mathcal{F} = \{P \in E(\mathbb{F}_{q^n}) \,:\, P = (x_P, y_P), x_P \in \mathbb{F}_q\}/\iota$, on peut utiliser les techniques présentées précédemment. Ainsi, à la différence de [Gau08, Die11], on obtient à partir du n-ième polynôme de sommation un système polynomial composé de n polynômes en $(n-1)$ variables, de degré total 2^{n-2} :

$$\varphi_0(e_1, \ldots, e_{n-1}) = 0, \quad \varphi_1(e_1, \ldots, e_{n-1}) = 0, \quad \ldots \quad \varphi_{n-1}(e_1, \ldots, e_{n-1}) = 0, \tag{7.9}$$

où $e_1, \ldots, e_{n-1}$ sont les polynômes élémentaires symétriques en les variables $x_{P_1}, \ldots, x_{P_{n-1}}$. Comme ce système est surdéterminé, on peut le résoudre beaucoup plus rapidement.

Exemple 7.3.1. *On considère un exemple simple d'application de la variante $n-1$ à la courbe elliptique E définie sur $\mathbb{F}_{101^3} \simeq \mathbb{F}_{101}[t]/(t^3+t+1)$ d'équation*

$$E : y^2 = x^3 + ax + b, \quad a = 60t^2 + 52t + 44,\ b = 74t^2 + 87t + 58.$$

Cette courbe est de cardinalité $\#E = 1029583$ qui est un nombre premier, on prend alors le point $P = (84t^2 + 24t + 75, 92t^2 + 18t + 61)$ comme générateur du groupe des points rationnels de E.

Pour trouver des relations, on tire un point aléatoire $R = [606158]P = (37t^2 + 84t + 85, 86t^2 + 3t + 15)$ que l'on essaye de décomposer comme somme de deux points de $\mathcal{F}$. On considère pour cela le troisième polynôme de sommation partiellement symétrisé

$$\tilde{f}_3(e_1, e_2, x_R) = (e_1^2 - 4e_2)x_R^2 - 2(e_1(e_2 + a) + 2b)x_R + (e_2 - a)^2 - 4be_1.$$

En remplaçant a et b par leurs valeurs respectives données dans E, on obtient l'équation

$$(59t^2+29t+100)e_1^2+(27t^2+34t+32)e_1e_2+(31t^2+71t+55)e_1+e_2^2+(48t^2+83t+17)e_2+32t^2+16t+81=0$$

dont la restriction de Weil donne

$$\begin{cases} 100e_1^2 + 32e_1e_2 + 55e_1 + e_2^2 + 17e_2 + 81 = 0 \\ 29e_1^2 + 34e_1e_2 + 71e_1 + 83e_2 + 16 = 0 \\ 59e_1^2 + 27e_1e_2 + 31e_1 + 48e_2 + 32 = 0. \end{cases}$$

Ce système n'ayant pas de solution, le point R (comme la plupart des points de la courbe) n'est pas décomposable.

Lorsqu'on essaye de décomposer un autre multiple aléatoire $R = [580657]P = (16t^2 + 94t + 21, 80t^2 + 34t + 41)$, on obtient l'équation suivante

$$(61t^2+78t+59)e_1^2+(69t^2+14t+59)e_1e_2+(40t^2+20t+57)e_1+e_2^2+(40t^2+89t+80)e_2+12t^2+11t+77 = 0$$

qui produit le système

$$\begin{cases} 59e_1^2 + 59e_1e_2 + 57e_1 + e_2^2 + 80e_2 + 77 = 0 \\ 78e_1^2 + 14e_1e_2 + 20e_1 + 89e_2 + 11 = 0 \\ 61e_1^2 + 69e_1e_2 + 40e_1 + 40e_2 + 12 = 0. \end{cases}$$

Cette fois, on trouve que le système a une unique solution $(e_1, e_2) = (69, 75)$. Pour déterminer la décomposition on considère le trinôme du second degré $X^2-69X+75$, qui admet pour solutions 6 et $63 \in \mathbb{F}_{101}$ correspondant respectivement aux points $P_1 = (6, 77t^2+93t+35)$ et $P_2 = (63, t^2+66t+2) \in \mathcal{F} \cup -\mathcal{F}$. En testant les 4 choix possibles de signes, on trouve alors que $R = P_1 + P_2$. Comme $\#(\mathcal{F} \cup -\mathcal{F}) = 108$, il reste encore 54 relations de cette forme à trouver avant de compléter la phase de recherche de relations.

7.3.2 Analyse et comparaison avec la méthode de Gaudry et Diem

Pour pouvoir estimer la complexité de la variante $n-1$, il faut donner une borne sur la probabilité qu'un point aléatoire se décompose en une somme de $n-1$ points de $\mathcal{F}$ et déterminer le coût de la résolution du système polynomial correspondant.

Dans la suite, on suppose vérifiée l'hypothèse suivante :

Hypothèse 7.3.2. *Le nombre de points de $E(\mathbb{F}_{q^n})$ admettant une décomposition en somme de $n-1$ points de la base de factorisation $\mathcal{F}$ est en $\Omega(q^{n-1}/(n-1)!)$.*

Soit la courbe $\mathcal{C} = \{P \in E(\mathbb{F}_{q^n}) \ : \ P = (x_P, y_P),\ x_P \in \mathbb{F}_q\} \cup \{\mathcal{O}_E\}$ déjà introduite précédemment. Comme la cardinalité du quotient $\mathcal{C}^{(n-1)} = \mathcal{C}^{n-1}/\mathfrak{S}_{n-1}$ est approximativement $q^{n-1}/(n-1)!$, cette hypothèse signifie que la plupart des points de E ont pour préimages par l'application somme $\varsigma : \mathcal{C}^{(n-1)} \to E(\mathbb{F}_{q^n}), (P_1, \ldots, P_{n-1}) \mapsto \sum P_i$, un nombre fini, borné indépendamment de n et q, de $(n-1)$-uplets non ordonnés de $\mathcal{C}^{(n-1)}$.

En fait, on peut montrer de façon plus précise que cette hypothèse est simplement un corollaire de la conjecture suivante, à condition que la cardinalité de $\mathcal{F}$ reste bien en $\Omega(q)$ (on a vu que ceci est toujours vrai, du moment que l'équation de la courbe de E satisfait la condition 7.2.2 et que $n \leq c \log_2 q$ où $c < 1/2$, voir section 7.2.2).

Conjecture. *Soient $\mathcal{C}^{(n-1)} = \mathcal{C}^{n-1}/\mathfrak{S}_{n-1}$ le $(n-1)$-ième produit symétrique de $\mathcal{C}$, et $W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$ la restriction de Weil de E relativement à l'extension $\mathbb{F}_{q^n}/\mathbb{F}_q$. Soit $\varsigma : (P_1, \ldots, P_{n-1}) \mapsto \sum_i P_i$ le morphisme de sommation de $\mathcal{C}^{(n-1)} \to W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$, défini sur $\mathbb{F}_q$. Alors*

1. *l'application* ς *est un morphisme de degré* 1 *de* $\mathcal{C}^{(n-1)}$ *dans* $\varsigma(\mathcal{C}^{(n-1)})$
2. *la fibre* $\varsigma^{-1}(R)$ *d'un point* $R \in \varsigma(\mathcal{C}^{(n-1)})$ *a une dimension positive si et seulement si il existe des points* $P_1, \ldots, P_{n-3} \in \mathcal{C}$ *tels que* $R = \varsigma(P_1, \ldots, P_{n-3}, \mathcal{O}_E, \mathcal{O}_E)$.

On note que comme $\mathcal{C}^{(n-1)}$ est une variété projective, son image par le morphisme ς est fermée dans $W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$. Le premier point de cette conjecture a été vérifié formellement pour $n = 3$ avec le logiciel Magma, et a toujours été satisfait dans les exemples pour les autres degrés d'extension considérés. En pratique, pour tester si cette propriété est vérifiée pour une courbe donnée, il est suffisant de trouver un point $R \in \varsigma(\mathcal{C}^{(n-1)})$ tel que la fibre $\varsigma^{-1}(R)$ est une variété zéro-dimensionnelle de degré 1. Le deuxième point est plus délicat à vérifier. Il est clair que si le point R est de la forme $R = \varsigma(P_1, \ldots, P_{n-3}, \mathcal{O}_E, \mathcal{O}_E)$, autrement dit tel que $R = \sum_{i=1}^{n-3} P_i$, alors la fibre correspondante $\varsigma^{-1}(R)$ est de dimension au moins 1 : elle contient en effet tout les $(n-1)$-uplets de la forme $(P_1, \ldots, P_{n-3}, Q, -Q)$. La réciproque est plus compliquée, mais a été vérifiée expérimentalement par recherche exhaustive pour des petites courbes lorsque $n = 5$. Il est néanmoins raisonnable de penser que, même si ce n'était pas le cas pour toutes les courbes elliptiques, une grande proportion d'entre elles vont satisfaire cette conjecture (probablement à un changement d'équation près). Dans tous les cas, sous l'hypothèse 7.3.2, le nombre de tests de décompositions à faire en moyenne avant d'obtenir une relation est en $O((n-1)!\, q^2)$.

Comme expliqué précédemment, le coût d'un test de décomposition, noté $\tilde{c}(n, q)$, se ramène au coût de la résolution d'un système surdéterminé de degré total 2^{n-2} composé de n équations et $(n-1)$ variables. La complexité de la phase de recherche de relations est donc en

$$O((n-1)!\, \tilde{c}(n, q)\, q^2).$$

Pour donner une borne supérieure effective de $\tilde{c}(n, q)$, on fait une analyse similaire à celle donnée dans la section 7.2.2. Génériquement, i.e. lorsque le point R que l'on décompose n'est pas dans l'image par ς de $\mathcal{C}^{(n-1)}$, le système surdéterminé correspondant n'admet pas de solution. La base de Gröbner de l'idéal associée est alors $\{1\}$ pour n'importe quel ordre, y compris grevlex. Dans les autres cas, i.e. lorsqu'une décomposition existe, le système admet généralement un petit nombre de points, voir exactement un point si la conjecture précédente est correcte. L'idéal correspondant est alors maximal et sa base de Gröbner contient seulement des polynômes linéaires ; trouver les solutions correspondantes est donc immédiat. On note toutefois que même si l'idéal est zéro-dimensionnel de degré strictement plus grand que 1, on s'attend à ce que ce degré reste suffisamment petit et donc que la résolution puisse encore se faire aisément. Exceptionnellement, la dimension de la fibre est supérieure à 1 (ce qui peut se voir aisément sur la base de Gröbner pour l'ordre grevlex, voir section 1.1.5) ; il est bien sûr toujours possible de déduire de ces points des relations utiles, mais le plus simple est encore de ne pas comptabiliser ces rares décompositions.

Remarque 7.3.3. *On note que cette situation est en fort contraste avec celle de l'algorithme de Gaudry et Diem, où le degré de l'idéal est génériquement égal à la borne de Bézout* $2^{n(n-1)}$. *Ceci signifie que l'ensemble des solutions contient génériquement* $2^{n(n-1)}$, *mais que la plupart d'entre elles sont dans la clôture algébrique de* $\mathbb{F}_q$, *vu que la probabilité de trouver une décomposition est seulement en* $1/n!$. *Dans leur approche, le calcul d'une base de Gröbner pour l'ordre grevlex n'est donc pas suffisant pour résoudre le système et l'algorithme FGLM devient nécessaire ; le degré doublement exponentiel en* n *de l'idéal devient alors particulièrement bloquant.*

Pour obtenir une borne supérieure sur la complexité du calcul de la base de Gröbner pour l'ordre grevlex du système (7.9), on fait l'hypothèse suivante

Hypothèse 7.3.4. *Le degré maximal des polynômes qui apparaît durant le calcul de la base de Gröbner pour l'ordre grevlex du système* (7.9) $\varphi_0, \ldots, \varphi_{n-1}$ *est plus petit que la borne de Macaulay* $d = \sum_{i=0}^{n-1}(\deg \varphi_i - 1) + 1$.

Cette hypothèse est en particulier vérifiée dès que le système est semi-régulier (cas générique, cf. section 2.4.3), et a été satisfaite pour tous les systèmes testés en pratique.

Sachant que les φ_i sont tous de degré 2^{n-2} en $n-1$ variables, on obtient que $d = n2^{n-1} - n + 1$. Pour estimer le coût du calcul de la base de Gröbner, on peut alors comme précédemment faire appel aux résultats de la section 3.2.6 où l'on majore le coût du calcul par celui de la réduction de la matrice de Macaulay jusqu'au degré d. La taille de cette matrice est au plus le nombre de monômes en $n-1$ variables de degré inférieur ou égal à d, soit $\binom{n2^{n-2}}{n-1}$. On obtient avec des techniques de réduction rapide la borne supérieure suivante :

$$\tilde{c}(n, q) = \tilde{O}\left(\binom{n2^{n-2}}{n-1}^{\omega}\right), \tag{7.10}$$

où $\omega \leq 3$ est l'exposant de complexité effective de multiplication matricielle déjà introduit en section 2.2.2 (en pratique, $\omega = \log_2(7)$ lorsque l'on utilise la multiplication de Strassen [Str69]). Comme le nombre d'essais de décompositions à faire est de l'ordre de $(n-1)!\, q^2$, la complexité de la phase de recherche de relations est donc en

$$\tilde{O}\left((n-1)!\left(2^{(n-1)(n-2)} e^n\, n^{-1/2}\right)^{\omega} q^2\right).$$

La complexité de l'étape d'algèbre linéaire est toujours en $\tilde{O}(nq^2)$, donc négligeable par rapport à celle de l'étape précédente. Ceci implique le résultat suivant :

Théorème 7.3.5. *Soit E une courbe elliptique définie sur $\mathbb{F}_{q^n}$ et G un sous-groupe du groupe des points rationnels de la courbe. Lorsque les hypothèses 7.3.2 et 7.3.4 sont vérifiées, il existe un algorithme permettant de résoudre le DLP défini sur G avec une complexité en*

$$\tilde{O}\left((n-1)!\left(2^{(n-1)(n-2)} e^n\, n^{-1/2}\right)^{\omega} q^2\right). \tag{7.11}$$

Il est clair qu'à cause de ce facteur en q^2 dans la complexité, les méthodes génériques telles que Pollard-rho vont rester plus rapides que la variante $n-1$ lorsque $n \leq 4$. En revanche, pour $n > 4$ il existe une importante plage de valeurs de q pour lesquelles la variante $n-1$ est la plus efficace. En effet, avec l'estimée (7.11) on peut déjà voir qu'il existe une constante c pour laquelle cette variante est asymptotiquement plus rapide que les méthodes génériques dès que $5 \leq n \leq c \log_2 q$. Plus précisément, on peut voir que la variante $n-1$ est asymptotiquement plus rapide que l'algorithme de Pollard lorsque $n \leq \left(\frac{1}{2\omega} - \epsilon\right) \log_2 q$. De même, avec l'analyse de complexité faite en 7.2.2, on voit que la variante $n-1$ est plus rapide que l'approche de Gaudry et Diem dès que $n \geq \sqrt[3]{\left(\frac{2}{3-\omega} + \epsilon\right) \log_2 q}$. On note néanmoins que ces comparaisons n'ont de sens que du point de vue asymptotique (la variante $n-1$ n'étant pas pertinente lorsque $n \leq 4$).

Remarque 7.3.6.
La borne supérieure de $\tilde{c}(n, q)$ donnée en (7.10) *(et donc l'estimée du théorème 7.3.5) est pratique mais reste très large (de fait, obtenir des bornes précises pour le coût du calcul de bases de Gröbner est encore un problème ouvert). En effet, et comme déjà remarqué, on ne prend pas en compte le fait que la matrice de Macaulay est une matrice creuse et fortement structurée. On constate ce manque de précision en pratique. Par exemple pour $n = 5$, l'estimation* (7.10) *prédit de l'ordre de 10^{14} multiplications dans $\mathbb{F}_q$ pour faire le calcul de la base de Gröbner, alors que le calcul réel avec F4 ne requiert que 10^{10} multiplications (et 3 fois moins avec la variante F4Remake).*

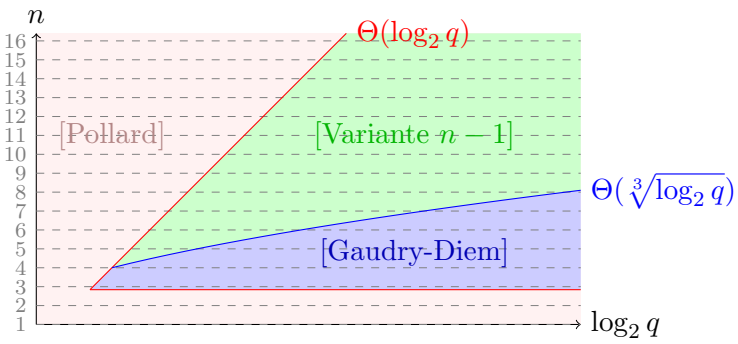


FIGURE 7.2 – Comparaison entre les méthodes de Pollard-rho, Gaudry and Diem et variante $n-1$.

Comparaison avec l'approche hybride

Pour pallier la difficulté de la résolution des systèmes intervenant dans la méthode de Gaudry et Diem, il peut être intéressant d'utiliser l'approche hybride présentée en section 3.3.2. On rappelle que l'idée de base dans cette approche est de chercher une solution du système en faisant un compromis entre le coût de la recherche exhaustive sur certaines variables et le coût du calcul de la base de Gröbner ; autrement dit, on calcule les bases de Gröbner des systèmes modifiés où certaines variables sont spécialisées. Dans ce contexte, le choix naturel est de spécialiser une variable ; la recherche exhaustive nécessite alors de multiplier par q le nombre de systèmes à résoudre, mais ces systèmes sont maintenant composés de n équations et $n-1$ variables. À première vue, cette approche semble très similaire à la variante $n-1$ proposée ; néanmoins le degré total des équations dans l'approche hybride est égal à 2^{n-1} alors qu'il est de 2^{n-2} pour la variante $n-1$. Le tableau 7.1 résume le nombre de systèmes à résoudre et leur caractéristiques lorsque l'on souhaite trouver une relation sur $E(\mathbb{F}_{q^n})$. Il met en évidence le fait que la variante $n-1$ offre un meilleur compromis que l'approche hybride en terme de nombre de systèmes à résoudre et de complexité de résolution.

Method	nombre moyen de systèmes	nombre de équations	nombre de variables	degré total
Gaudry-Diem	$n!$	n	n	2^{n-1}
Gaudry-Diem avec approche hybride	$n!\,q$	n	$n-1$	2^{n-1}
variante $n-1$	$(n-1)!\,q$	n	$n-1$	2^{n-2}

TABLE 7.1 – Comparaison entre l'approche hybride et la variante $n-1$ pour la recherche de relations sur $E(\mathbb{F}_{q^n})$.

7.3.3 Application au DLP sur $\mathbb{F}_{q^5}$

Bien que pertinente du point de vue théorique, l'approche de Gaudry-Diem reste difficile à mettre en pratique sur $\mathbb{F}_{q^n}$ dès que $n \geq 5$. Non seulement le calcul du sixième polynôme de sommation est problématique, mais le nombre très important de solutions attendues (de l'ordre de $2^{5(5-1)} \simeq 10^6$) dans $\overline{\mathbb{F}_{q^5}}$ rend la résolution très complexe ; on a vu en effet dans la section précédente que la complexité de la résolution dépend du degré $2^{n(n-1)}$ de l'idéal généré par les équations. Une idée naturelle pour diminuer ce degré serait d'ajouter les équations de corps $e_i^q - e_i$, malheureusement la valeur de q est trop grande pour que cela ait un intérêt. En particulier, on ne parvient pas en

pratique à faire une résolution complète d'un seul système dans l'approche de Gaudry-Diem, la mémoire requise excédant la capacité d'un ordinateur personnel standard. Par contre, en utilisant la variante $n-1$ ainsi que la variante F4 pour la résolution de système (cf. chapitre 3), on peut tester (et si nécessaire calculer) une décomposition sur $\mathbb{F}_{p^5}$ avec p premier de 32 bits en environ 8.5 s sur un processeur Intel Core 2 à 2.6 GHz.

En caractéristique 2, les calculs sont nettement plus rapides : les polynômes de Semaev sont en effet beaucoup plus creux qu'en caractéristique impaire, et les systèmes correspondants sont donc plus faciles à résoudre (voir section 7.1.2). On note cependant que les bornes données sur le degré de régularité et le degré des idéaux restent les mêmes, ainsi que les estimées de complexité. Les temps pour tester les décompositions avec l'approche de Gaudry-Diem sur $E(\mathbb{F}_{q^n})$ où $n \leq 4$ (q pair) sont donnés dans [Gra10] et montrent que, malgré le caractère très creux des polynômes, il est encore impossible en caractéristique 2 d'obtenir une relation sur $E(\mathbb{F}_{q^5})$. Il est intéressant de remarquer par contre que pour la variante $n-1$, on est capable (sur le même type de machine) de tester une décomposition en 4 points sur $\mathbb{F}_{2^{160}} = \mathbb{F}_{(2^{32})^5}$ en seulement 30 ms (au lieu des 8.5 s obtenus en grande caractéristique).

Malheureusement, cette variante reste encore trop lente pour pouvoir résoudre en temps raisonnable le problème ECDLP sur des corps de tailles compatibles avec les niveaux de sécurité exigés actuellement. Plus précisément, on peut estimer à partir de quelle taille de corps cette méthode est plus rapide que Pollard-rho, sachant qu'un seul test de décomposition requiert environ $3 \cdot 10^9$ multiplications sur $\mathbb{F}_p$ pour p impair et environ $2 \cdot 10^7$ multiplications dans $\mathbb{F}_{2^d}$ pour le cas binaire [3]. La variante $n-1$ est alors plus rapide dès que la cardinalité du corps de base est plus grande que 2^{60} dans le cas de la caractéristique impaire, ou 2^{45} dans le cas de la caractéristique 2.

On note cependant que cette variante permet une attaque efficace des problèmes non standards présentés en section 4.4, tel que le problème Diffie-Hellman statique assisté d'un oracle. Cette attaque est décrite complètement dans la section suivante.

7.3.4 Application au problème SDHP sur $E(\mathbb{F}_{q^5})$

De l'idée de Semaev consistant à décomposer des points de $E(\mathbb{F}_{q^n})$ dans une base de factorisation $\mathcal{F}$ bien choisie, on déduit sans difficulté une attaque sur le problème SDHP naturellement défini sur $E(\mathbb{F}_{q^n})$ (cf. section 4.4.1). Néanmoins, lorsque l'on considère la base de factorisation $\mathcal{F}$ telle que $\mathcal{F} \cup (-\mathcal{F}) = \{P \in E(\mathbb{F}_{q^n}) : x(P) \in \mathbb{F}_q\}$, seulement une petite proportion de points peuvent être décomposés (1 parmi $n!$ ou parmi $(n-1)!\,q$ selon la méthode utilisée). Pour pouvoir décomposer un challenge donné X, il faut donc faire une sorte de variation de la phase de descente des méthodes classiques de calcul d'indices, afin de rendre plus "aléatoire" le challenge.

L'algorithme assisté d'un oracle proposé pour attaquer le problème SDHP est le suivant :

1. Durant la phase d'apprentissage, demander à l'oracle de calculer $Q = [d]P$ pour tous les points P de la base $\mathcal{F}$.
2. Étant donné un challenge X, choisir un entier aléatoire premier à l'ordre de G et calculer $X_r = [r]X$.
3. Tester si X_r peut être décomposé en une somme de m points de $\mathcal{F}$: $X_r = \sum_{i=1}^{m} \epsilon_i P_i$, avec $\epsilon_i \in \{-1; 1\}$.

3. Comme la recherche de relations est la phase dominante de l'algorithme, on peut négliger la phase d'algèbre linéaire.

4. Si X_r n'est pas décomposable, recommencer au point 2 ; sinon calculer $Y = [s]\,(\sum_{i=1}^{m} \epsilon_i Q_i)$ où $s = r^{-1} \mod |G|$.

Bien entendu il est possible d'appliquer des techniques similaires pour résoudre d'autres variantes de SDHP, telles que les problèmes "Delayed Target" Discrete Logarithm ou Diffie-Hellman (voir [KM08]). On note qu'une approche similaire à celle présentée ici a été donnée indépendemment par Granger dans [Gra10].

En pratique, l'oracle est souvent limité à un seul appareil électronique (comme une carte à puce par exemple), alors que les tests de décomposition peuvent être facilement distribués sur plusieurs ordinateurs munis d'une bonne puissance de calcul. Les appels à l'oracle sont donc clairement le facteur limitant de cette attaque de SDHP. Comme expliqué dans [KM08], il est cependant possible d'utiliser une méthode à la Harley permettant de rééquilibrer l'étape d'apprentissage et l'étape de calcul de décomposition du challenge. L'idée consiste à réduire artificiellement la base de factorisation à un sous-ensemble $\mathcal{F}'$ de cardinalité $\#\mathcal{F}/l$ où $l > 1$, de façon à diminuer le nombre d'appels à l'oracle d'un facteur l. En contrepartie, on augmente le nombre de tests de décomposition avant d'obtenir une relation ; plus précisément, si l'on considère des décompositions en m points, on augmente le coût de la phase de décomposition du challenge d'un facteur l^m. Le choix optimal du paramètre l dépend alors du coût d'un appel à l'oracle et de la puissance de calcul disponible pour faire les tests de décomposition. On note toutefois que puisqu'il n'y a pas de phase d'algèbre linéaire dans l'algorithme proposé, le rééquilibrage fait est beaucoup plus simple que celui proposé dans l'approche "double large prime variation" de Gaudry.

En utilisant les estimées données dans la section précédente, on peut comparer notre méthode de décomposition avec celle de Gaudry et Diem pour l'attaque du problème SDHP sur $E(\mathbb{F}_{q^n})$. Pour simplifier, on considère le même facteur l pour les deux approches, de façon à avoir le même nombre d'appels à l'oracle $q/2l$ dans les deux cas. La complexité de la recherche d'une relation est $n!\,l^n c(n,q)$ avec la méthode de Gaudry-Diem, contre $(n-1)!\,l^{n-1} q\tilde{c}(n,q)$ pour la variante $n-1$. On voit alors que celle-ci est meilleure asymptotiquement dès que $n \geq \sqrt{\frac{1}{3-\omega}\log_2(q/l)}$. À l fixé, cette plage de valeurs pour q et n est donc moins intéressante que celle trouvée pour le problème ECDLP, par contre lorsque l augmente (et donc lorsque l'on diminue le nombre d'appels à l'oracle), la plage de valeurs pour lesquelles notre variante est plus performante augmente.

On a vu cependant qu'en pratique, sur les ordinateurs standards actuels, on arrive seulement à obtenir des décompositions d'un point donné en somme d'au plus 4 points de la base de factorisation (voir section 7.3.3). Si l'on considère $m = 4$ sur $E(\mathbb{F}_Q)$, la complexité asymptotique optimale de l'attaque de SDHP si $Q = q^4$ est obtenue en réduisant la base de factorisation à une taille de $q^{4/5}$, pour un coût total en $\tilde{O}(Q^{1/5})$ qui correspond à la complexité de la variante $n-1$ lorsque $Q = \tilde{q}^5$. On note cependant que les constantes intervenant dans la complexité de la variante sont beaucoup plus petites. En effet, les détails des calculs dans les deux cas sont résumés dans la table suivante :

	$Q = q^4$ [Gaudry-Diem]	$Q = \tilde{q}^5$ [variante $n-1$]
nb d'appels à l'oracle	$\frac{Q^{1/4}}{2l}$	$\frac{Q^{1/5}}{2\tilde{l}}$
coût de la décomposition	$4!\,l^4\,c(4, Q^{1/4})$	$4!\,\tilde{l}^4\,Q^{1/5}\,\tilde{c}(4, Q^{1/5})$

TABLE 7.2 – Comparaison entre la méthode de Gaudry-Diem et la variante $n-1$ pour attaquer SDHP sur $E(\mathbb{F}_Q)$, selon que le type de l'extension $\mathbb{F}_Q$.

Pour faire une comparaison équitable, on égalise le nombre d'appels à l'oracle dans les deux

cas, i.e. on choisit $\tilde{l} = lQ^{1/5}/Q^{1/4} = lQ^{-1/20}$. Avec ce choix, le coût d'une décomposition avec la variante $n-1$ devient $4!\,l^4\tilde{c}(4, Q^{1/5})$. Par ailleurs, comme il est plus facile de résoudre un système surdéterminé composé de 4 équations et 3 variables qu'un système composé de 3 équations et 3 variables [4], on a $\tilde{c}(4, Q^{1/5}) < c(3, Q^{1/5}) \ll c(4, Q^{1/4})$; à titre d'exemple, pour $Q = 257^{5\times4}$, on trouve avec Magma (V2.15-15) $\tilde{c}(4, Q^{1/5}) = 768\,\text{s}$ et $c(4, Q^{1/4}) = 15\,476\,\text{s}$. De façon similaire, en caractéristique 2, si l'on prend $Q = 2^{160}$, on trouve que $\tilde{c}(4, Q^{1/5}) = 0.67\,\text{s}$ et $c(4, Q^{1/4}) = 272\,\text{s}$.

Quelle que soit la caractéristique, on voit donc que pour une taille de corps donnée, le problème SDHP assisté d'un oracle sur courbe elliptique est moins sûr lorsqu'il est défini sur des extensions de degré 5 que sur celles de degré 4.

Un exemple pratique d'application

On s'intéresse à une attaque du problème Diffie-Hellman statique pour la courbe 'Well Known Group' 3 donnée dans le protocole de détermination de clé d'Oakley du standard IPSEC [IET98]. Cette courbe est définie sur $\mathbb{F}_{2^{155}} = \mathbb{F}_2[u]/(u^{155} + u^{62} + 1)$ par l'équation

$$E \ : \ y^2 + xy = x^3 + (u^{18} + u^{17} + u^{16} + u^{13} + u^{12} + u^9 + u^8 + u^7 + u^3 + u^2 + u + 1)$$

et $\#E(\mathbb{F}_{2^{155}}) = 12 \cdot 3805993847215893016155463826195386266397436443$. Bien que l'extension de $\mathbb{F}_2$ considérée soit de degré composé, la courbe est considérée comme sûre puisque résistante à toute attaque connue du DLP, notamment l'attaque GHS (voir [JMS01, Sma01] et section 6.2.2).

Soit $\mathcal{F}$ la base de factorisation définie par un ensemble de représentants de $\{(x, y) \in E(\mathbb{F}_{2^{155}}) : x \in \mathbb{F}_{2^{31}}\}/\sim$, où $(x, y) \sim (x, y + x)$. Comme expliqué précédemment, il n'est pas possible d'obtenir des décompositions d'un point donné en somme de 5 points de cette base ; on utilise donc la variante $n-1$ pour attaquer le SDHP.

Soit d l'entier secret intervenant dans le problème Diffie-Hellman statique considéré. Durant la phase d'apprentissage, on demande à l'oracle de calculer $[d]P$ pour tous les points $P \in \mathcal{F}$, soit environ 2^{31} appels au total. Pour un challenge X donné, on essaie de décomposer des multiples aléatoires $[r]X$ (où r est premier avec 12) comme une somme de 4 points de $\mathcal{F}$:

$$[r]X = \pm P_1 \pm P_2 \pm P_3 \pm P_4.$$

Dès que l'on trouve r pour lequel la décomposition réussit, on déduit du calcul de $[r^{-1}](\pm[d]P_1 \pm [d]P_2 \pm [d]P_3 \pm [d]P_4)$ la valeur de $[d]X$. En moyenne, il faut de l'ordre de $4!\,2^{31}$ essais avant d'obtenir une telle décomposition. La principale difficulté est donc de détecter rapidement si un point $[r]X$ se décompose en somme de 4 ; en utilisant l'algorithme variante de F4 présentée en chapitre 3, on arrive à tester une décomposition en seulement 22.95 ms sur un processeur Intel Xeon à 2.93 Ghz. Il est bien sûr tout à fait possible de paralléliser la phase de recherche d'une relation : on peut ainsi par exemple diminuer le temps de calcul à 2 semaines moyennant un accès 1000 processeurs équivalents à celui utilisé.

La seule autre méthode connue pour attaquer le SDHP sur une courbe elliptique générale consiste à d'abord résoudre le problème du logarithme discret afin de retrouver directement l'entier secret d. Bien qu'aucune attaque directe du DLP de la courbe d'Oakley n'ait été envisagée, on peut essayer d'extrapoler, à titre de comparaison, les temps obtenus pour des attaques lancées sur des

4. On peut par exemple commencer par résoudre le système composé des 3 premières équations puis tester la compatibilité des solutions avec l'équation restante.

courbes de plus petites tailles. En particulier, une attaque (encore en cours) du challenge Certicom ECC2K-130 devrait pouvoir résoudre le DLP sur une courbe de Koblitz définie sur $\mathbb{F}_{2^{131}}$ en environ 2 ans avec 3000 CPU bi-cores à 3 Ghz [BBB+09]. Le groupe défini par cette courbe contenant un sous-groupe d'ordre premier de taille environ 2^{129}, contre 2^{151} pour Oakley, on peut estimer qu'une attaque similaire utilisant la même puissance de calcul prendrait environ $2 \times 2^{(151-129)/2} \simeq 4\,000$ ans sur la courbe d'Oakley.

7.3.5 Variante de l'approche de Nagao en genre $g > 1$

On donne une variante de l'approche de Nagao présentée en section 7.2.3, qui offre quelques similarités avec la méthode de crible utilisée dans les cribles de corps de nombres et de fonctions [LL93, Adl94]. L'objectif est d'obtenir de façon plus efficace des relations sur une courbe hyperelliptique admettant un modèle imaginaire $\mathcal{H} : \ y^2 + h_0(x)y = h_1(x)$ de genre g définie sur $\mathbb{F}_{q^n}$. On considère pour cela des relations faisant intervenir uniquement des diviseurs de la base de factorisation, dont on rappelle qu'elle est donnée par un ensemble de représentants de

$$\{D_Q \in \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{q^n}) \ : \ D_Q \sim (Q) - (\mathcal{O}_{\mathcal{H}}),\ Q \in \mathcal{H}(\mathbb{F}_{q^n}),\ x(Q) \in \mathbb{F}_q\}/\iota. \tag{7.12}$$

Autrement dit, au lieu de chercher à décomposer un diviseur R donné, on essaie de trouver des sommes nulles d'éléments de la base :

$$\sum_{i=1}^{m} \pm\left((Q_i) - (\mathcal{O}_{\mathcal{H}})\right) \sim 0. \tag{7.13}$$

Heuristiquement, le nombre attendu de relations de cette forme composées de m éléments de $\mathcal{F}$ est $q^{m-ng}/m!$. Comme il faut collecter au moins $q/2$ relations indépendantes avant de passer à la phase d'algèbre linéaire, on considère des sommes de $m = ng + 2$ éléments (si $q > (ng+2)!$, ce qui sera toujours le cas dans les applications considérées). La technique de décomposition d'un diviseur R reste cependant nécessaire pour la phase de descente.

Pour trouver ces relations de la forme (7.13), on commence comme dans l'approche originelle de Nagao en introduisant l'espace de Riemann-Roch $\mathcal{L}(m(\mathcal{O}_{\mathcal{H}}))$. C'est un $\mathbb{F}_{q^n}$-espace vectoriel de dimension $\ell + 1$ où $\ell = m - g = (n-1)g + 2$, dont une base est

$$\mathcal{B} = \{1, x, \ldots, x^{m_1}, y, xy, \ldots, x^{m_2}y\},$$

où $m_1 = \lfloor m/2 \rfloor$ et $m_2 = \lfloor (m-1)/2 \rfloor - g$. En particulier, une fonction $f \in \mathbb{F}_{q^n}(\mathcal{H})^*$ qui s'écrit comme combinaison linéaire des éléments de $\mathcal{B}$ et dont l'ordre du pôle en $\mathcal{O}_{\mathcal{H}}$ est bien m, va s'annuler en exactement m points de $\mathcal{H}$. Les abscisses de ces points sont également les racines du polynôme (paramétré par les coefficients λ_i de f dans la base $\mathcal{B}$)

$$F_{\lambda_1,\ldots,\lambda_\ell}(x) = x^{ng+2} + \sum_{i=0}^{ng+1} c_i(\lambda_1, \ldots, \lambda_\ell)x^i,$$

obtenu à partir du produit $f(x,y) \cdot f(x, -y - h_0(x))$ où l'on a remplacé $y(y + h_0(x))$ par $h_1(x)$, suivi d'une normalisation au point à l'infini (ce qui revient à fixer à 1 le coefficient de x^{m_1} lorsque m est pair ou celui de yx^{m_2} lorsque m est impair).

Pour obtenir des relations de la forme (7.13), il faut donc trouver des valeurs de $\lambda_1, \ldots, \lambda_\ell$ telles que F ait m racines dans $\mathbb{F}_q$. Une première condition est que F soit à coefficients dans $\mathbb{F}_q$, ce qui

signifie qu'après restriction de Weil, les $n(m-g)$ variables provenant des λ_i doivent satisfaire un système polynomial quadratique à $(n-1)m$ équations. Avec le choix de décomposition en $m = ng+2$ éléments, on obtient ainsi un système sous-déterminé à $n(n-1)g+2n-2$ équations et $n(n-1)g+2n$ variables.

Lorsque les paramètres n et g ne sont pas trop gros, on constate qu'il est possible de calculer une base de Gröbner pour l'ordre lexicographique de l'idéal associé à ce système, par exemple en utilisant l'algorithme de changement d'ordre Gröbner walk mentionné en section 2.5. En spécialisant deux variables supplémentaires, on obtient alors des systèmes faciles à résoudre (voir ci-dessous) ; il reste alors à vérifier pour chacune des solutions $\lambda_1, \ldots, \lambda_\ell$ obtenues si le polynôme F est bien scindé pour en déduire des relations entre les éléments de $\mathcal{F}$.

Remarque 7.3.7. *On note que ce type de précalcul n'est pas réalisable avec la version initiale de l'algorithme de Nagao : cela nécessiterait la résolution d'un système qui dépendrait des composantes de R dans la représentation de Mumford vues comme des paramètres formels. Le nombre de variables ajoutées est alors bien trop grand pour que le calcul reste raisonnable.*

Cas des courbes hyperelliptiques définies sur des extensions quadratiques

Un type de courbes pour lesquelles cette approche est très efficace est celui des courbes hyperelliptiques définies sur des extensions quadratiques (i.e. lorsque $n = 2$) ; on détaille la construction dans ce cas.

Soit $\mathcal{H}$ une courbe hyperelliptique de genre g définie sur $\mathbb{F}_{q^2} = \mathbb{F}_q(t)/(P(t))$ admettant un modèle imaginaire $y^2 = h(x)$ où $\deg h = 2g+1$. Les polynômes f et F définis dans l'approche Riemann-Roch présentée ci-dessus ont pour expression

$$f(x,y) = (x^{g+1} + \lambda_g x^g + \ldots + \lambda_0) + \mu y,$$

$$F_{\lambda_0,\ldots,\lambda_g,\mu}(x) = (x^{g+1} + \lambda_g x^g + \ldots + \lambda_0)^2 - \mu^2 h(x),$$

où $\lambda_0, \ldots, \lambda_g, \mu \in \mathbb{F}_{q^2}$. On remarque déjà à ce stade que lorsque la variable μ est nulle, le polynôme f ne dépend plus de la variable y ; ceci signifie que lorsqu'un point P intervient dans la relation de type (7.13) correspondante, son image $\iota(P)$ par l'involution ι intervient aussi dans celle-ci. Autrement dit, lorsque $\mu = 0$, on obtient des relations triviales de la forme

$$(P_1) + (\iota(P_1)) + \ldots + (P_{g+1}) + (\iota(P_{g+1})) - (2g+2)\mathcal{O}_{\mathcal{H}} \sim 0.$$

Pour les éviter, on doit donc chercher des solutions

$$(\lambda_{0,0}, \ldots, \lambda_{g,0}, \lambda_{0,1}, \ldots, \lambda_{g,1}, \mu_0, \mu_1) \in \mathbb{V}_{\mathbb{F}_q}(\mathrm{I}:(\mu_0, \mu_1)),$$

où I est l'idéal constitué des $2(g+1)$ polynômes quadratiques en $2(g+2)$ variables obtenus après restriction des scalaires sur le polynôme $F_{\lambda_0,\ldots,\lambda_g,\mu}(x) \in \mathbb{F}_{q^2}[x]$ en posant $\lambda_i = \lambda_{i,0} + t\lambda_{i,1}$ et $\mu^2 = \mu_0 + t\mu_1$.

Afin de simplifier considérablement les expressions des polynômes engendrant I (et donc les calculs de bases de Gröbner), il est pertinent de définir le corps $\mathbb{F}_{q^2}$ comme une extension de la forme $\mathbb{F}_q(t)/(t^2-\omega)$. En effet, la condition $F_{\lambda_0,\ldots,\lambda_g,\mu}(x) = (1\cdot x^{g+1} + \lambda_g x^g + \ldots + \lambda_0)^2 - \mu^2 h(x) \in \mathbb{F}_q[x]$ est alors équivalente à la condition

$$2(1 \cdot x^{g+1} + \lambda_{g,0} x^g + \cdots + \lambda_{0,0})(\lambda_{g,1} x^g + \cdots + \lambda_{0,1}) - \mu_0 h_1(x) - \mu_1 h_0(x) = 0 \in \mathbb{F}_q[x], \qquad (7.14)$$

où $h(x) = h_0(x) + th_1(x)$ et les $\lambda_{i,j}, \mu_{i,j}$ sont les variables apparaissant dans la restriction des scalaires. Avec cette expression simple, on constate que l'idéal $J \subset \mathbb{F}_q[\lambda_{0,0}, ..., \lambda_{0,g}, \lambda_{0,1}, ..., \lambda_{g,1}]$ obtenu après élimination des variables μ_0 et μ_1 est multi-homogène de degré $(1,1)$ en les variables $(1 : \lambda_{0,0} : \ldots : \lambda_{g,0}), (\lambda_{0,1} : \ldots : \lambda_{g,1})$. Cette propriété supplémentaire du système va permettre d'en accélérer grandement le calcul de la base de Gröbner pour l'ordre lexicographique ; par exemple, si l'on prend $g \log_2 q \simeq 70$, on obtient les temps suivants avec Magma

genre	2	3	4
nombre éq./var.	6/8	8/10	10/12
temps de calcul	<1 s	2 s	1 h

alors que le même calcul pour un système aléatoire composé de 6 équations quadratiques en 8 variables n'aboutit pas en plus de 6 h. De plus, si l'on note

$$\pi_1 : (\lambda_{0,0}, ..., \lambda_{g,0}, \lambda_{0,1}, ..., \lambda_{g,1}) \mapsto (\lambda_{0,0}, ..., \lambda_{g,0})$$

la projection sur le premier bloc de variables, alors $\pi_1(\mathbb{V}(J))$ est une variété de dimension 1, ce qui peut se voir en calculant sa classe dans l'anneau de Chow de $\mathbb{P}^{g+1}(\mathbb{F}_q)$. On rappelle que l'anneau de Chow d'une variété V est une généralisation du groupe de Picard, et est défini comme l'ensemble des sommes formelles de sous-variétés irréductibles de V modulo la relation d'équivalence rationnelle [Har77, Appendix A]. On a les isomorphismes $\mathrm{CH}(\mathbb{P}^{g+1} \times \mathbb{P}^g) \simeq \mathbb{Z}[h_1, h_2]/(h_1^{g+2}, h_2^{g+1})$ et $\mathrm{CH}(\mathbb{P}^{g+1}) \simeq \mathbb{Z}[h_1]/(h_1^{g+2})$, où le degré total d'un élément correspond à sa codimension, et le poussé en avant $\pi_{1*} : \mathrm{CH}(\mathbb{P}^{g+1} \times \mathbb{P}^g) \to \mathrm{CH}(\mathbb{P}^{g+1})$ est donné par $\pi_{1*}(h_1^{e_1} h_2^{e_2}) = h_1^{e_1}$ si $e_2 = g$ et 0 sinon. L'idéal J étant défini par $2g$ polynômes multi-homogènes de bidegré (1,1), la classe de $\mathbb{V}(J)$ est alors $[\mathbb{V}(J)] = (h_1 + h_2)^{2g} = \binom{2g}{g} h_1^g h_2^g + \binom{2g}{g+1} h_1^{g+1} h_2^{g-1}$, et $\pi_{1*}([\mathbb{V}(J)]) = \binom{2g}{g} h_1^g$. Par conséquent $\pi_1(\mathbb{V}(J))$ est de codimension g dans $\mathbb{P}^{g+1}$, donc de dimension 1. Cette observation est également valable si l'on projette sur le deuxième bloc.

On remarque alors que plutôt que de fixer la valeur d'une variable (par exemple $\lambda_{0,0}$), il est plus pertinent de choisir directement un point dans la projection $\pi_1(\mathbb{V}(J))$. D'après l'expression polynomiale donnée en (7.14), on voit qu'une fois ce point fixé, le système obtenu est linéaire en les variables restantes et la variété correspondante est en fait un espace vectoriel de dimension 1.

Si l'on exploite toutes ces propriétés du système, une méthode efficace de résolution consiste alors à :

1. choisir un point dans la variété $\pi_1(\mathbb{V}(J))$ de dimension 1, ce qui revient à déterminer des valeurs des variables $(\lambda_{0,0}, ..., \lambda_{g,0})$ pour une spécialisation de $\lambda_{0,0}$ donnée ;
2. spécialiser une seconde variable $\lambda_{0,1}$ puis déduire de façon immédiate les solutions exprimées linéairement en fonction de $\lambda_{0,1}$.

Pour les valeurs de λ et μ obtenues, on peut déduire une relation de la forme (7.13) en factorisant $F_{\lambda_0,...,\lambda_g,\mu}(x) \in \mathbb{F}_q[x]$ si ce polynôme est scindé. Avec cette première amélioration, on obtient déjà un gain non négligeable par rapport à l'approche initiale de Nagao (voir l'exemple du chapitre 8).

On peut cependant optimiser davantage l'algorithme de recherche de relations. En effet, il est possible de se passer du calcul de la factorisation du polynôme $F_{\lambda,\mu}$ correspondant à chacune des valeurs de λ et μ trouvées en modifiant le parcours d'évaluation des deux variables supplémentaires afin de mettre en place une technique de crible. Cette technique permet d'accélérer la recherche de relations en faisant un compromis temps/mémoire. L'idée consiste à remplacer la spécialisation de la seconde variable $\lambda_{0,1}$ par une évaluation en x ; dans ce cas, la méthode de résolution devient :

1. Spécialiser la variable $\lambda_{0,0}$, choisir un point correspondant dans $\pi_1(\mathbb{V}(J))$, et exprimer les variables restantes en fonction de $\lambda_{0,1}$; on a alors $F_{\lambda_0,\ldots,\lambda_g,\mu} \in \mathbb{F}_q[x, \lambda_{0,1}]$ de degré 2 en $\lambda_{0,1}$ (donc petit par rapport au degré en x qui vaut $2g+2$).
2. Spécialiser la variable $x \in \mathbb{F}_q$ (au lieu de la variable $\lambda_{0,1}$) et résoudre le trinôme du second degré correspondant en $\lambda_{0,1}$. On associe alors pour chacune des valeurs de $\lambda_{0,1}$ un compteur dont la valeur est comprise entre 0 et $2g+2$, correspondant au nombre de fois où une évaluation en x donne la valeur correspondante de $\lambda_{0,1}$.

Une fois les q valeurs de x parcourues, il ne reste plus qu'à sélectionner les valeurs de $\lambda_{0,1}$ pour lesquelles le compteur en x atteint le maximum, à savoir $2g+2$: on est alors assuré que pour chacune ces valeurs de $\lambda_{0,1}$, le polynôme $F_{\lambda_0,\ldots,\lambda_g,\mu}$ correspondant est scindé dans $\mathbb{F}_q[x]$.

La méthode est très efficace, puisque l'on remplace le coût de la factorisation d'un polynôme de degré $2g+2$ par celle d'un polynôme de degré 2 ; en contre-partie, on augmente le coût mémoire puisque l'on doit stocker un tableau de taille q contenant les valeurs des compteurs. On peut aller encore plus loin dans le compromis temps-mémoire en précalculant les racines carrées et les inverses des éléments de $\mathbb{F}_q$, ce qui accélère notablement la résolution des trinômes du second degré à l'étape 2. On renvoie à la section 8.3 pour un exemple concret du gain apporté par cette technique de crible dans le cas où $g = 3$. Si l'on n'est pas dans le cas d'une extension quadratique, on peut aussi tenter de faire fonctionner ce crible : la difficulté principale est d'être capable, après spécialisation de certaines variables, d'exprimer en fonction d'une seule variable toutes les variables restantes.

Du point de vue théorique, lorsque l'on fait l'analyse asymptotique à n, g fixés et $q \to \infty$, la complexité semble cependant moins bonne que celle de l'approche originelle de Nagao. En effet, puisque l'on fait des décompositions en $ng+2$ points (au lieu de ng points), une technique type variation "double large primes" donne une complexité asymptotique en $\tilde{O}(q^{2-2/(ng+2)})$, à comparer avec les $\tilde{O}(q^{2-2/ng})$ de l'approche originelle. Il est à noter toutefois que la technique de crible utilisée permet d'améliorer cette complexité asymptotique, dans l'esprit de la méthode de calcul d'indices présentée en section 5.5 : au lieu de parcourir les q valeurs possibles des abscisses des éléments de la base de factorisation définie en (7.12), on se contente de parcourir les q^α abscisses des éléments de la base des petits facteurs. La phase de recherche de relation est ainsi accélérée d'un facteur $q^{1-\alpha}$, pour un total en $\tilde{O}(q^\alpha q^{(1-\alpha)(ng)})$ (cf. section 5.4.2). Pour équilibrer avec l'algèbre linéaire en $\tilde{O}(q^{2\alpha})$, la valeur asymptotiquement optimale de α est $1 - 1/(ng+1)$, pour un coût total en

$$O(q^{2-2/(ng+1)}).$$

Cette complexité est encore supérieure à celle de l'approche de Nagao, mais les constantes cachées dans les notations de Landau sont bien plus faibles pour la méthode présentée ici. En pratique, cette attaque avec technique de crible est largement plus efficace pour les tailles de corps pertinentes en cryptographie.

Chapitre 8

Attaque par recouvrement et décomposition

On propose dans ce chapitre d'attaquer le problème du logarithme discret sur courbes elliptiques en combinant les techniques de descente de Weil et d'attaques par décomposition présentées dans les deux précédents chapitres. Cette nouvelle attaque [JV11a] s'applique dès lors que la courbe E considérée est définie sur une extension composée d'un corps fini ; elle consiste à d'abord transférer le DLP sur la jacobienne d'une courbe $\mathcal{C}$ définie sur une extension intermédiaire, puis à appliquer une méthode de décomposition à la courbe ainsi obtenue, plutôt qu'un calcul d'indices classique.

Après avoir expliqué et analysé en détail cette nouvelle attaque, on montrera comment la mettre en application sur des courbes définies sur des extensions quartiques et sextiques. En particulier, on listera toutes les courbes potentiellement vulnérables à cette attaque, en donnant pour chacune d'elles une analyse de la complexité ainsi qu'une comparaison aux autres attaques possibles. Enfin, on donnera un exemple complet de l'attaque réalisée sur une courbe elliptique de 132 bits, résistante a priori à toute attaque connue auparavant.

8.1 Description et analyse

On s'intéresse particulièrement aux courbes elliptiques E définies sur $\mathbb{F}_{q^n}$ pour lesquelles les revêtements construits par les techniques GHS présentées en chapitre 6 sont de genre trop élevé pour avoir un intérêt vis-à-vis du DLP (ce qui est le cas pour la plupart des courbes), et telles que le degré de l'extension n soit trop grand (typiquement $n \geq 6$) pour que les attaques par décomposition présentées en chapitre 7 puissent être appliquées.

Dans la suite, on supposera notamment que n *est composé*. La méthode de recouvrement et décomposition consiste à d'abord appliquer GHS sur l'extension intermédiaire $\mathbb{F}_{q^n}/\mathbb{F}_{q^d}$ (où d suffisamment petit) pour transférer le DL sur la jacobienne d'une courbe $\mathcal{C}$ définie sur $\mathbb{F}_{q^d}$, puis à appliquer l'attaque par décomposition sur $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{q^d})$ avec comme corps de base $\mathbb{F}_q$ pour résoudre le DLP. Ainsi, l'algorithme débute par un précalcul sur $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{q^d})$ durant lequel on collecte suffisamment de relations entre les éléments de la base de factorisation de façon à obtenir les logarithmes discrets de chacun de ces éléments. Une fois le précalcul fait, on utilise l'application de recouvrement entre $\mathcal{C}$ et E pour transférer le DLP de $E(\mathbb{F}_{q^n})$ sur $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{q^d})$; une décomposition de type Nagao permet alors d'obtenir une représentation de l'élément relevé comme somme d'éléments de

la base de factorisation, et donc de déduire le logarithme discret.

Le fait que n soit composé étend les possibilités du choix des extensions sur lesquelles appliquer GHS ou les décompositions. Cependant, on verra sur les exemples qu'en pratique la combinaison des deux attaques offre le meilleur compromis entre la taille de la base de factorisation et la complexité des tests de décomposition. D'un point de vue théorique, on peut utiliser les estimations de complexités données dans les chapitres précédents afin de comparer les méthodes :

Variété abélienne	$\#\mathcal{F}$	Complexité 2LP pour $q \to \infty$		Coût approximatif d'une décomposition	
$\mathrm{Jac}_{\mathbb{F}_q}(\mathcal{C}_1)$	q	$\tilde{O}(q^{2-2/g_1})$	$\tilde{O}(q^{2-2/(d_1-2)})$	$g_1!$	$(d_1-2)!$
$\mathrm{Jac}_{\mathbb{F}_{q^d}}(\mathcal{C}_2)$	q^d	$\tilde{O}(q^{d(2-2/g_2)})$	$\tilde{O}(q^{d(2-2/(d_2-2))})$	$g_2!$	$(d_2-2)!$
$W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$	q	$\tilde{O}(q^{2-2/n})$		$n!\,2^{3n^2}$	
$W_{\mathbb{F}_{q^n}/\mathbb{F}_{q^d}}(E)$	q^d	$\tilde{O}(q^{d(2-2/k)})$		$k!\,2^{3k^2}$	
$W_{\mathbb{F}_{q^d}/\mathbb{F}_q}(\mathrm{Jac}_{\mathbb{F}_{q^d}}(\mathcal{C}_2))$	q	$\tilde{O}(q^{2-2/(dg_2+1)})$		$(dg_2+2)!$	

TABLE 8.1 – Comparaison des différents calculs d'indices sur $E(\mathbb{F}_{q^n})$ lorsque $n = dk$.

Dans ce tableau, $\mathcal{C}_1$, resp. $\mathcal{C}_2$, désigne un recouvrement de E défini sur $\mathbb{F}_q$, resp. $\mathbb{F}_{q^d}$, de genre g_1 et de degré d_1, resp. g_2 et d_2, typiquement obtenu par la méthode GHS. Les valeurs différentes dans les deux premières lignes correspondent à l'utilisation de calcul d'indices suivant le genre (présenté en section 5.4.2) ou suivant le degré (présenté en section 5.5). Le coût mentionné en dernière ligne ne tient pas en compte le précalcul, fait une seule fois, de la base de Gröbner pour l'ordre lexicographique, voir section 7.3.5.

À cause de la taille de la base de factorisation, les calculs d'indices sur $\mathrm{Jac}_{\mathbb{F}_{q^d}}(\mathcal{C}_2)$ et $W_{\mathbb{F}_{q^n}/\mathbb{F}_{q^d}}(E)$ sont peu pertinents, et le coût des décompositions sur $W_{\mathbb{F}_{q^n}/\mathbb{F}_q}(E)$ est prohibitif. La comparaison entre les deux attaques restantes dépend essentiellement des genres des courbes obtenues (les valeurs données en dernière ligne correspondent à l'utilisation de la variante de Nagao présentée en section 7.3.5 après transfert). Or dans la technique GHS, plus le degré de l'extension de corps est grand, moins il est probable que le genre de la courbe résultante soit proche de la valeur optimale. En particulier, il est raisonnable d'espérer que le genre de $\mathcal{C}_2$ soit relativement proche de $k = n/d$, quitte à faire une marche d'isogénies, alors que le genre de $\mathcal{C}_1$ a de grandes chances d'être plutôt de l'ordre de 2^n. Dans ces conditions, l'attaque combinée par recouvrement puis décomposition est la plus performante.

8.2 Applications et comparaisons

Pour améliorer les implantations matérielles de l'arithmétique sur courbes elliptiques, des corps finis particuliers, appelés *optimal extension fields* (OEF), ont été proposés dans [BP01] : ce sont les corps de la forme $\mathbb{F}_{p^d}$ où p est un nombre premier type pseudo-Mersenne, i.e. de la forme $p = 2^n + c$ où $|c| \leq 2^{\lfloor n/2 \rfloor}$, et où d est tel qu'il existe un polynôme irréductible de la forme $X^d - \omega \in \mathbb{F}_p[X]$. L'utilisation de ces corps en cryptographie basée sur courbes elliptiques a été plusieurs fois proposée, comme dans le standard Oakley de l'IPSEC déjà mentionné [IET98] ou dans le standard coréen EC-KDSA [LL99]. Dans la plupart des exemples, le degré d'extension d est assez petit ; les valeurs $d = 6$ ou $d = 4$ font partie des choix recommandés. La cryptanalyse de courbes définies sur ce type

d'extension est donc légitime et non pas purement académique.

8.2.1 Extensions sextiques

Une courbe elliptique définie sur $\mathbb{F}_{q^6}$ est potentiellement sujette à d'autres attaques que les attaques génériques présentées en section 4.2, dont la complexité est $\tilde{O}(q^3)$. Avant de donner les résultats de la technique de recouvrement et décomposition sur ce type d'extension, on récapitule les autres possibilités d'attaques et leur efficacité respective, ainsi que le nombre de courbes auxquelles elles s'appliquent.

Si l'on considère l'extension $\mathbb{F}_{q^6}/\mathbb{F}_q$, les attaques présentées en chapitre 7 ne sont pas praticables. En effet, aucune des attaques par décomposition (que ce soit celle de Gaudry et Diem ou la variante $n-1$) ne permet d'obtenir ne serait-ce qu'une relation. Quant à l'attaque GHS, il est rare qu'elle donne des recouvrements de petit genre. Une analyse détaillée, similaire à celle menée en section 6.3.1 montre qu'on ne peut obtenir que des recouvrements de genre supérieur à 9, le genre 9 étant atteint dans des cas très rares.

1. En caractéristique impaire, les seules courbes elliptiques $E_{|\mathbb{F}_{q^6}}$ pour lesquelles la méthode GHS donne un recouvrement de genre 9 ont une équation de la forme

$$E_\alpha : \ y^2 = (x-\alpha)(x-\sigma(\alpha))(x-\sigma^2(\alpha))(x-\sigma^3(\alpha)), \tag{8.1}$$

où $\alpha \in \mathbb{F}_{q^6} \setminus (\mathbb{F}_{q^2} \cup \mathbb{F}_{q^3})$, ou de la forme

$$E_{\alpha,\beta} : \ y^2 = g(x)(x-\alpha)(x-\sigma(\alpha))(x-\beta)), \tag{8.2}$$

où $\alpha \in \mathbb{F}_{q^3} \setminus \mathbb{F}_q$, $\beta \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$, et $g \in \mathbb{F}_q[x]$, $\deg g \leq 1$. Dans les deux cas, on a $r = 6$ et $\overline{m} = m = 4$ avec $M_h = X^4 + X^2 + 1$ pour le premier type et $X^4 + X^3 + X + 1$ pour le deuxième. Pour toutes les autres courbes n'ayant pas une équation définie sur un sous-corps strict, on a soit $r = 6$ et $m = 5$, soit $r \geq 7$ et $m \geq 4$, pour un genre du recouvrement au moins égal à 13.

2. En caractéristique deux, le seul type d'équations de courbes elliptiques $E_{|\mathbb{F}_{q^6}}$ admettant un recouvrement de genre 9 avec la méthode GHS est de la forme

$$E_{\alpha,\beta,\gamma} : \ y^2 + y = \beta x + \alpha + \gamma/x, \tag{8.3}$$

où $\beta \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$, $\gamma \in \mathbb{F}_{q^3}^*$, $\mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(\gamma) = 0$ et $\alpha \in \mathbb{F}_{q^6}$. On a alors $M_\beta = X^2+1$, $M_\gamma = X^2+X+1$, et $M_h = X^4+X^3+X+1$. Le second plus petit genre possible est ensuite 11, pour $M_\beta = X^2+1$ et $M_\gamma = X^3 + 1$.

On peut donner une borne supérieure sur le nombre de classes d'isomorphismes de telles courbes. Parmi les isomorphismes de courbes elliptiques, on considère les changements de variables dans $\mathrm{PGL}_2(\mathbb{F}_q)$, de la forme $(x', y') = \big((ax+b)/(cx+d), y/(cx+d)^2\big)$. On vérifie alors aisément que deux courbes E_α et $E_{\alpha'}$ sont isomorphes (ou tordues l'une de l'autre) si α et α' appartiennent à la même $\mathrm{PGL}_2(\mathbb{F}_q)$-orbite. Comme le nombre de ces orbites dans $\mathbb{F}_{q^6} \setminus (\mathbb{F}_{q^2} \cup \mathbb{F}_{q^3})$ est $q^3 + q - 1$, il y a au plus $O(q^3)$ courbes de type (8.1). Pour les courbes du deuxième type (8.2), on peut utiliser la transitivité de l'action de $\mathrm{PGL}_2(\mathbb{F}_q)$ sur $\mathbb{F}_{q^3} \setminus \mathbb{F}_q$ pour fixer la valeur de α ; le nombre de ces courbes est alors au plus $O(q^2)$. Ceci montre que la proportion de courbes définies sur $\mathbb{F}_{q^6}$ pour lesquelles la méthode GHS donne un recouvrement de genre 9 défini sur $\mathbb{F}_q$ est au plus égal à $1/q^3$. Enfin, pour les courbes en caractéristique 2, le changement de variables $(x', y') = (\beta x, \beta(yx+\gamma))$ ramène

l'équation (8.3) à l'équation $y^2 + xy = x^3 + \alpha x^2 + (\beta\gamma)^2$, impliquant en particulier $j(E) = (\beta\gamma)^{-2}$. Deux courbes $E_{\alpha,\beta,\gamma}$ et $E_{\alpha',\beta',\gamma'}$ sont donc isomorphes ou tordues si et seulement s'il existe $c \in \mathbb{F}_q$ tel que $\beta' = c\beta$ et $\gamma' = \gamma/c$. On compte donc au total $(q^2 - q)(q^2 - 1)/(q - 1) = q^3 - q$ classes d'isomorphismes, soit au plus $O(q^3)$ courbes de ce type.

En revanche, si l'on considère l'extension $\mathbb{F}_{q^6}/\mathbb{F}_{q^2}$, on a vu en section 6.3 qu'il est possible d'obtenir des recouvrements réalisant la valeur optimale $g = 3$ pour le genre. En utilisant les méthodes de calcul d'indices du chapitre 5 sur le recouvrement, on obtient des attaques de complexité asymptotique en $\tilde{O}(q^{8/3})$ dans le cas hyperelliptique, donc légèrement meilleures que les attaques génériques, et en $\tilde{O}(q^2)$ dans le cas non-hyperelliptique. Dans les deux cas, la complexité en mémoire est en $\tilde{O}(q^2)$, ce qui correspond au stockage des relations "double large primes".

Le dénombrement des courbes de type (6.7), (6.8) et (6.9) a été réalisé par Momose et Chao dans [MC05]. Ils montrent que le nombre de classes d'isomorphismes de courbes elliptiques admettant un recouvrement non-hyperelliptique de genre 3 (type (6.8) ou (6.9)) est en $\Theta(q^6)$ – la moitié de celles ayant toute leur 2-torsion définie sur $\mathbb{F}_{q^6}$ – ce qui correspond à une proportion non négligeable de courbes. Les courbes admettant un recouvrement hyperelliptique de genre 3 (type (6.7)) sont par contre plus rares, le cardinal de l'ensemble de leurs classes d'isomorphimes étant en $\Theta(q^4)$. Néanmoins, en utilisant une marche d'isogénies comme décrit en section 6.2.4, on peut augmenter le nombre de courbes admettant un tel recouvrement. Comme il y a de l'ordre de q^3 courbes par classe d'isogénies [1], on s'attend à ce que chacune contienne de l'ordre de q courbes vulnérables. Ceci n'est pas tout à fait exact, dans la mesure où les courbes de la forme (6.7) ont toutes leur cardinalité divisible par 4. On peut cependant conjecturer que toute courbe de cardinalité divisible par 4 est isogène, après une marche de longueur en moyenne q^2, à une courbe admettant un revêtement hyperelliptique de genre 3 par GHS. En caractéristique 2, on peut dénombrer de la même façon les courbes de type (6.11) : $y^2 + y = x + \alpha + \gamma/x$, $\mathrm{Tr}_{\mathbb{F}_{q^6}/\mathbb{F}_{q^2}}(\gamma) = 0$. Comme $j(E) = \gamma^{-2}$, une courbe elliptique ordinaire admet une équation de ce type si et seulement si $j(E)^{-1}$ est de trace nulle sur $\mathbb{F}_{q^2}$, ce qui correspond donc à $\Theta(q^4)$ classes d'isomorphismes. Ici encore, on peut conjecturer que toutes les courbes ordinaires sont isogènes à une courbe de ce type, après une marche d'isogénies de longueur en moyenne q^2.

Enfin, considérer l'extension $\mathbb{F}_{q^6}/\mathbb{F}_{q^3}$ pour faire du calcul d'indices n'apporte pas d'amélioration par rapport aux attaques génériques, à cause de la taille des bases de factorisation dans ce contexte.

Le cas le plus favorable pour l'attaque par recouvrement et décomposition est lorsque la courbe $E_{|\mathbb{F}_{q^6}}$ admet un recouvrement hyperelliptique de genre 3 défini sur $\mathbb{F}_{q^2}$. Avec des décompositions à la Nagao, la complexité asymptotique est en $\tilde{O}(q^{5/3})$. En pratique, il vaut mieux appliquer la méthode donnée en section 7.3.5 avec la technique de crible, dont la complexité asymptotique est en $\tilde{O}(q^{12/7})$, mais avec une constante cachée bien meilleure. De plus la complexité en mémoire est réduite à $\tilde{O}(q)$. L'attaque ainsi obtenue est donc la plus performante de toutes celles présentées dans cette section. Dans le cas où le revêtement n'est pas hyperelliptique, la complexité asymptotique théorique reste la même, mais la constante correspondant au coût d'une décomposition devient nettement plus importante.

Finalement, il est aussi possible de travailler avec l'extension $\mathbb{F}_{q^6}/\mathbb{F}_{q^3}$. On peut montrer que la plupart des courbes admettent alors un recouvrement par une courbe de genre 2 ; une description simple de ce revêtement est donnée dans [Sch03]. Une courbe elliptique E est sous *forme de Scholten*

1. Cette valeur provient de la borne de Hasse et du fait que deux courbes sont isogènes si et seulement si elles ont la même cardinalité.

si son équation est de la forme :

$$y^2 = \alpha x^3 + \beta x^2 + \sigma^3(\beta)x + \sigma^3(\alpha), \tag{8.4}$$

où $\alpha, \beta \in \mathbb{F}_{q^6}$. Pour qu'une courbe possède une équation de ce type, il suffit que sa cardinalité soit divisible par 4 (quitte à passer par une 2-isogénie, voir [Sch03]) ou impaire si $j(E) \notin \mathbb{F}_{q^3}$ [AMNS06]. En remplaçant x par x^2 dans l'équation (8.4), on trouve immédiatement un recouvrement de E par une courbe hyperelliptique $\mathcal{H}$ de genre 2 d'équation

$$y^2 = \alpha x^6 + \beta x^4 + \sigma^3(\beta)x^2 + \sigma^3(\alpha),$$

a priori définie sur $\mathbb{F}_{q^6}$. En faisant le changement de variables $(x, y) = \left(\frac{X-c}{X-\sigma^3(c)}, \frac{Y}{(X-\sigma^3(c))^3}\right)$, on voit que la courbe $\mathcal{H}$ est en fait définie sur $\mathbb{F}_{q^3}$ par l'équation

$$Y^2 = \alpha(X-c)^6 + \beta(X-c)^4(X-\sigma^3(c))^2 + \sigma^3(\beta)(X-c)^2(X-\sigma^3(c))^4 + \sigma^3(\alpha)(X-\sigma^3(c))^6.$$

La complexité d'une attaque par décomposition sur la jacobienne de $\mathcal{H}$ est alors asymptotiquement en $\tilde{O}(q^{5/3})$. Cependant, les décompositions sont plus difficiles à calculer que dans le cas d'une courbe hyperelliptique de genre 3 définie sur $\mathbb{F}_{q^2}$: avec la technique de Nagao, il faut résoudre un système polynomial quadratique de 12 équations et 12 variables à la place d'un système de seulement 6 équations en 6 variables. De plus, la technique de crible ne fonctionne pas dans ce cadre, dans la mesure où le calcul initial de base de Gröbner pour l'ordre lexicographique n'aboutit pas sur un ordinateur personnel.

On résume en table 8.2 les différences entre toutes les méthodes exposées ci-dessus. Pour ne pas se limiter aux complexités asymptotiques, on donne aussi des estimations du temps de calcul pour la résolution du DLP sur une courbe elliptique E définie sur un corps $\mathbb{F}_{p^6}$ de type OEF, où p est un nombre premier de 27 bits. La cardinalité de E est supposée divisible par un grand nombre premier r de 160 bits. On insiste sur le fait que ces estimations, qui reposent sur des extrapolations et des hypothèses simplificatrices, sont uniquement indicatives et ne servent que comme base de comparaison. Toutes les expériences de recherches de relations (ou d'itérations dans le cas de Pollard-rho) ont été effectuées avec le logiciel Magma V2-17-5 sur un Intel Core 2 Duo à 2.6 GHz ; aucune garantie n'est donnée quant à l'optimalité des implantations réalisées.

La première attaque considérée est Pollard-rho. En utilisant la méthode de détection des cycles de Floyd présentée en section 4.2.3, l'espérance du nombre d'itérations nécessaire est $0.94\sqrt{r} \approx 1.14 \times 10^{24}$ (voir [CFA+06, §19.5.1]). On note qu'il existe d'autres méthodes de détection de cycles nécessitant moins d'évaluations de la fonction F, et qu'on peut encore gagner un facteur $\sqrt{2}$ en utilisant le fait que le calcul de l'inverse d'un point est très rapide, mais cela ne change pas l'ordre de grandeur du total. Le calcul de 10 000 itérations se fait en 13.91 s, ce qui correspond à 5×10^{13} années de calcul pour la résolution du DLP.

Pour les méthodes de calcul d'indices, la difficulté est d'estimer le coût de l'algèbre linéaire, qui sert aussi pour trouver le rééquilibrage optimal des variations "large primes". Comme point de départ des extrapolations, on utilise l'expérience réalisée sur une courbe de 132 bits (voir section suivante) où la résolution d'un système matriciel creux de taille environ 666 000 a nécessité de l'ordre de 3 500 h·CPU. On fera donc l'hypothèse que le coût de la phase d'algèbre linéaire pour une base de factorisation de taille n est de $(n/666\,000)^2 \cdot 3\,500 \cdot 160/130$ h, soit $n^2 \cdot 3.5 \times 10^{-5}$ s.

Les premières méthodes à base de calcul d'indices que l'on considère sont celles qui utilisent une base de factorisation $\mathcal{F}$ (ou de grands facteurs pour les variantes "large primes") dont la taille est en $q^2/2$. Il faut remarquer que la complexité spatiale associée à cette taille est extrêmement

Méthode	Complexité asymptotique	Complexité spatiale	Estimée temps de calcul (ans)
Pollard-rho sur $E(\mathbb{F}_{p^6})$	$\tilde{O}(p^3)$	$\tilde{O}(1)$	5.0×10^{13}
Calcul d'indices sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{p^2})$, $g = 3$ (*)	$\tilde{O}(p^{8/3})$	$\tilde{O}(p^2)$	7.2×10^{10}
Calcul d'indices sur $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_{p^2})$, $d = 4$	$\tilde{O}(p^2)$	$\tilde{O}(p^2)$	670 000
Décompositions sur $E((\mathbb{F}_{p^2})^3)$	$\tilde{O}(p^{8/3})$	$\tilde{O}(p^2)$	1.3×10^{12}
Décompositions sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{p^3})$, $g = 2$	$\tilde{O}(p^{5/3})$	$\tilde{O}(p)$	4.5×10^6
Calcul d'indices sur $\mathrm{Jac}_{\mathcal{C}}(\mathbb{F}_p)$, $d = 10$ (**)	$\tilde{O}(p^{7/4})$	$\tilde{O}(p)$	1 370
Décompositions sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{p^2})$, $g = 3$ (*)	$\tilde{O}(p^{5/3})$	$\tilde{O}(p)$	730
Crible sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{p^2})$, $g = 3$ (*)	$\tilde{O}(p^{12/7})$	$\tilde{O}(p)$	430

(*) : seulement pour $\Theta(p^4)$ courbes. (**) : seulement pour $\Theta(p^3)$ courbes.

TABLE 8.2 – Comparaison entre toutes les attaques du DLP sur $E(\mathbb{F}_{p^6})$ où $|p|_2 = 27$ avec Magma V2-17-5 sur un Intel Core 2 Duo.

problématique pour toute implantation concrète : rien que la liste des points de $\mathcal{F}$ nécessite au moins $54 \times 2^{54}/2 \approx 2^{60}$ bits, soit de l'ordre de l'exa-octet, ce qui est des dizaines de fois supérieur aux tailles des plus grosses bases de données actuelles. Cette observation reste valable pour les méthodes "one large prime" ou "double large primes" ; seule une restriction de base simple à la Harley ne serait pas concernée par cette difficulté, au prix d'une complexité en temps supérieure.

Dans le cas où E admet un recouvrement hyperelliptique $\mathcal{H}_{|\mathbb{F}_{p^2}}$ de genre 3, on peut utiliser le calcul d'indices sur sa jacobienne pour résoudre le DLP après transfert. L'expérience montre que 10 000 tests, correspondant à 1 689 relations, nécessitent 13.27 s, soit un temps de calcul d'environ 2.2×10^6 années pour trouver $p^2/2$ relations. Avec l'hypothèse que l'on a faite, la phase d'algèbre linéaire (nonobstant les problèmes d'accès mémoire) prend un temps bien supérieur, de 9×10^{19} années. Pour rééquilibrer les deux phases avec la méthode "double large primes", il faut réduire la taille de la base de factorisation d'un facteur environ 50 000. Le temps total de calcul devient alors de 7.2×10^{10} années.

Si E admet un recouvrement non-hyperelliptique $\mathcal{C}_{|\mathbb{F}_{p^2}}$ de genre 3, on peut trouver un modèle plan de degré 4 de $\mathcal{C}$ pour appliquer la méthode de calcul d'indices de Diem ; l'expérience montre alors que 10 000 tests, correspondant à 4 972 relations, nécessitent 11.74 s ; cela amène à 670 000 années le temps nécessaire à la collecte de $p^2/2$ relations. Pour cette méthode, on a vu que la taille optimale de la base de factorisation était la plus petite permettant d'engendrer de l'ordre de $p^2/2$ relations, ce qui correspond à considérer de l'ordre de $p \approx 2^{27}$ "petits facteurs", et le coût de la phase d'algèbre linéaire devient alors négligeable par rapport à la recherche de relations.

Enfin, on peut appliquer directement à E la technique de décomposition de Gaudry et Diem en utilisant $\mathbb{F}_{p^2}$ comme corps de base, de sorte que l'on cherche des décompositions en somme de trois points. Il faut alors 22.35 s pour effectuer 100 tests, produisant 36 relations. Autrement dit, trouver une relation est 80 fois plus lent qu'en passant par le revêtement hyperelliptique de genre 3, et le rééquilibrage entre les deux phases n'est pas le même. Il faut dans ce cas réduire la base de factorisation d'un facteur 12 000, pour un temps total de calcul de 1.3×10^{12} années ; le gain par rapport aux attaques génériques est alors faible.

On considère maintenant les méthodes de calcul d'indices pour lesquelles la taille de la base de factorisation $\mathcal{F}$ est en $p/2$. Comme expliqué précédemment, le calcul d'une seule décomposition pour la méthode de Gaudry et Diem appliquée à $E(\mathbb{F}_{p^6})$ avec $\mathbb{F}_p$ comme corps de base dépasse déjà les capacités d'un ordinateur personnel, et cette méthode n'apparaît donc pas en table 8.2. Dans le cas très rare où E admet une équation de la forme (8.1), et donc un revêtement non-hyperelliptique de genre 9, la méthode de Diem utilisée sur un modèle plan de degré 10 permet d'obtenir, après 200 000 tests de décomposition, 6 relations en 123 s. Par extrapolation, on trouve un temps de calcul de 43.5 années pour trouver de l'ordre de $p/2$ relations. Avec l'hypothèse que l'on a faite sur le coût de la phase d'algèbre linéaire, le calcul de cette phase sans rebalancement prend environ 5 000 ans. En utilisant la variante "double large primes" adaptée, on trouve que le rééquilibrage optimal consiste à réduire la base de factorisation d'un facteur 2.7, pour un total de 1 370 années de calcul.

Si E admet un revêtement hyperelliptique $\mathcal{H}_{|\mathbb{F}_{p^2}}$ de genre 3, on peut appliquer l'attaque par recouvrement et décompositions qui est l'objet de ce chapitre. On cherche alors des décompositions sur $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{p^2})$, soit avec la méthode de Nagao, soit avec la variante présentée en section 7.3.5. Dans le premier cas, avec des décompositions en somme de 6 points cela prend 126 s d'effectuer 5 000 tests, pour un total de 9 relations. Par extrapolation, on trouve un temps de calcul de 29.8 années pour la collecte de $p/2$ relations, l'algèbre linéaire demandant toujours environ 5 000 ans. Le rééquilibrage optimal consiste alors à réduire la base d'un facteur 3.7, pour un temps total de 730 années de calcul. Dans le deuxième cas, en utilisant la technique de crible on obtient 3 300 relations en 1 800 s, ce qui est 25 fois plus rapide qu'avec la méthode de Nagao [2]. Avec la variante "double large primes" adaptée au crible, on trouve que le rebalancement optimal correspond à une réduction de la base de factorisation d'un facteur 4.8, pour un temps total de 430 années de calcul.

Finalement, on peut appliquer l'attaque par recouvrement et décompositions sur un revêtement hyperelliptique de genre 2 défini sur $\mathbb{F}_{p^3}$. On a vu alors qu'on ne pouvait pas appliquer la technique de crible, faute de pouvoir réaliser le précalcul. Pour cette attaque combinée avec des décompositions en somme de 6 points, un unique test prend 3 780 s, ce qui est 150 000 fois plus lent que la même méthode appliquée à un revêtement de genre 3 défini sur $\mathbb{F}_{p^2}$. Cela signifie qu'un rebalancement n'est pas nécessaire et que la phase de recherche de relations, qui prend 4.5×10^6 années, domine le temps de calcul.

8.2.2 Extensions quartiques

On peut faire la même analyse que ci-dessus pour les courbes elliptiques définies sur $\mathbb{F}_{q^4}$, et appliquer l'attaque par recouvrement et décomposition avec $\mathbb{F}_{q^2}$ comme extension intermédiaire. Comme précédemment, une courbe elliptique $E_{|\mathbb{F}_{q^4}}$ admet un revêtement de genre 2 défini sur $\mathbb{F}_{q^2}$ si elle possède une équation sous forme de Scholten, ce qui arrive dès que sa cardinalité est impaire (et $j(E) \notin \mathbb{F}_q$) ou divisible par 4. Ainsi, environ trois quarts des courbes elliptiques définies sur $\mathbb{F}_{q^4}$ sont directement vulnérables. On peut alors appliquer la technique de crible et résoudre le DLP avec une complexité en $\tilde{O}(q^{8/5})$, et un faible coût de décomposition. Si l'on utilise des décompositions à la Nagao, la complexité asymptotique devient seulement $\tilde{O}(q^{3/2})$, mais avec une constante plus importante, correspondant au coût de résolution d'un système polynomial quadratique de 4 équations en 4 variables. Si l'on applique directement l'attaque par décomposition de Gaudry et Diem sur $E(\mathbb{F}_{p^4})$, la complexité asymptotique reste en $\tilde{O}(q^{3/2})$, mais avec une constante encore plus grande : les systèmes à résoudre à chaque test de décomposition sont encore composés de 4 équations en 4 inconnues, mais le degré total des polynômes est cette fois 8 au lieu de 2. Enfin, il est aussi

2. En pratique, lors d'expériences réalisées avec des implatations optimisées en langage C, le rapport de temps entre les deux méthodes est plutôt de l'ordre de 750, voir section suivante.

possible de faire du calcul d'indices avec un recouvrement défini sur $\mathbb{F}_q$. Le cas des extensions quartiques a été étudié dans [AMNS06], où il est montré que la plupart des courbes elliptiques $E_{|\mathbb{F}_{p^4}}$ admettent un revêtement par une courbe non-hyperelliptique de genre 9. L'utilisation de ce revêtement donne une complexité asymptotique en $\tilde{O}(q^{16/9})$, ou potentiellement $\tilde{O}(q^{7/4})$ avec un modèle plan de degré 10.

Toutes ces attaques sont asymptotiquement plus performantes que les méthodes génériques, mais la différence est moins significative que dans le cas des extensions sextiques. Néanmoins, une proportion plus importante de courbes est directement vulnérable à l'attaque par recouvrement et décomposition, et ne nécessite pas de passer par une marche d'isogénies.

8.3 Un exemple de calcul

On donne un exemple pratique d'attaque par recouvrement et décomposition appliquée à une courbe elliptique définie sur $\mathbb{F}_{p^6}$ avec $p = 4\,194\,319 = 2^{22} + 15$ un nombre premier de 23 bits. On définit $\mathbb{F}_{p^2}$ comme $\mathbb{F}_p[i]$ où $i^2 = -1$ et $\mathbb{F}_{p^6}$ comme $\mathbb{F}_{p^2}[\theta]$ où $\theta^3 = 2$.

La courbe elliptique E considérée est donnée par l'équation de Weierstrass suivante :

$$y^2 = (x - c)(x - \alpha)(x - \sigma^2(\alpha))$$

où $\sigma : x \mapsto x^p$, $c = 1\,048\,587$ et $\alpha = 3\,812\,894\,\theta^2 + 3\,527\,164\,\theta + 1\,048\,580\,i$.

Cette courbe est intéressante à étudier, puisque si l'on applique la technique GHS pour l'extension $\mathbb{F}_{p^6}/\mathbb{F}_p$, on trouve que le recouvrement défini sur $\mathbb{F}_p$ vérifie $m = 5$ ($M_h = X^5+X^4+X^3+X^2+X+1$) et $r = 8$ ($R = \{\sigma^i(\alpha); 0 \leq i < 6\} \cup \{c, \infty\}$), ce qui correspond à un cas plutôt favorable. Cependant le genre, égal à 33, est bien trop grand pour que les méthodes de calcul d'indices soient efficaces.

En revanche, comme déjà vu en section 6.3.1, une courbe elliptique admettant une telle équation admet un recouvrement hyperelliptique $\mathcal{H}$ de genre 3 défini sur $\mathbb{F}_{p^2}$, dont l'équation est donnée par $y^2 = \left(x + \phi(x) + \phi^{\sigma^2}(x) + \phi^{\sigma^4}(x) - 4c\right) N(x)^2$, avec $N(x)$ le polynôme minimal de α sur $\mathbb{F}_{p^2}$ et $\phi : x \mapsto \frac{(\alpha - \sigma^4(\alpha))(\sigma^2(\alpha) - \sigma^4(\alpha))}{x - \sigma^4(\alpha)} + \sigma^4(\alpha)$. Pour les valeurs des paramètres choisis, l'équation de $\mathcal{H}$ est donc donnée par

$$\begin{aligned} y^2 \;=\; & x^7 + (1\,048\,579\,i + 4\,194\,290)x^6 + (2\,097\,203\,i + 2\,359\,305)x^5 + (2\,686\,984\,i + 393\,267)x^4 + \\ & (3\,538\,925\,i + 1\,359\,881)x^3 + (126\,973\,i + 2\,424\,826)x^2 + (589\,830\,i + 3\,083\,272)x + \\ & 4\,021\,007\,i + 1\,363\,461. \end{aligned}$$

L'application de recouvrement $\pi : \mathcal{H} \to E$ est alors

$$\pi(x, y) = \left(\frac{x + \phi(x) + \phi^{\sigma^2}(x) + \phi^{\sigma^4}(x)}{4}, \frac{y(x - \phi^{\sigma^2}(x))(x - \phi^{\sigma^4}(x))}{2(x - \sigma^4(\alpha))} \right).$$

La cardinalité commune de l'ensemble des points rationnels de $E(\mathbb{F}_{p^6})$ et des éléments de la jacobienne $\mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{p^2})$ est égale à

$$N = 4\ell = 4 \cdot 136115867461471233446652598568206220 1601,$$

où ℓ est un nombre premier de 131-bits, et la base de factorisation $\mathcal{F} \subset \mathrm{Jac}_{\mathcal{H}}(\mathbb{F}_{p^2})$ contient de l'ordre de 2.1 millions d'éléments.

Afin d'obtenir de meilleurs performances, on utilise le crible présenté en section 7.3.5, ainsi qu'une élimination gaussienne structurée (plutôt que la variation "double large prime"). Le calcul de la base de Gröbner pour l'ordre lexicographique du système quadratique composé de 10 équations en 8 variables prend environ 2 s avec Magma V2.16-12 sur un processeur Intel Core 2 Duo à 2.6 GHz. La collecte de 51 883 659 relations, soit à peu près 25 fois le nombre de relations nécessaires, a été obtenue en 3 751 s avec 200 cœurs répartis sur des processeurs Intel Xeon 5570 quadri-cœurs à 2.93 GHz [3]. À titre de comparaison, sur un seul cœur du même type seulement 344 relations sont obtenues avec la méthode de Nagao durant le même temps de calcul. Ceci montre que la variante de Nagao proposée en section 7.3.5 est environ 750 fois plus rapide pour la recherche de relations que l'approche originelle, et au moins 100 fois plus rapide si l'on prend en compte le coût de l'algèbre linéaire. L'élimination gaussienne structurée, ainsi que les relations supplémentaires obtenues durant la phase de crible, permettent de réduire par un facteur 3 la taille des matrices intervenant dans la phase d'algèbre linéaire ; ainsi, après un calcul de 1 357 s sur un unique cœur, l'élimination gaussienne produit un système composé de 666 062 équations en 665 061 variables, où chaque équation fait intervenir entre 8 et 62 éléments de la base de factorisation. Le nombre total de coefficients non nuls de la matrice est 33 761 662 et tous ces coefficients sont à valeurs dans $\{-1; 1\}$.

La phase la plus coûteuse en temps de calcul est clairement l'algèbre linéaire. En utilisant une implantation MPI de l'algorithme de Lanczos, celle-ci prend environ 27 h 16 mn sur 128 cœurs des mêmes processeurs Intel. Une grande proportion de ce temps est utilisée par les communications MPI, puisqu'à chaque étape de l'algorithme, 42.5 MB de données sont échangées entre les 16 machines bi-processeurs. On obtient ainsi les logarithmes de tous les éléments de la base qui n'ont pas été supprimés durant l'élimination gaussienne structurée, et en substituant ces valeurs dans le système linéaire initial, on obtient en moins de 10 mn sur un unique cœur tous les logarithmes discrets modulo ℓ des éléments de la base de factorisation :

$$\begin{aligned}
\log(1, 1\,778\,117 + 4\,043\,006\,i) &= 47810632712543597011455056269164844169 1\\
\log(2, 2\,470\,708 + 2\,816\,377\,i) &= 60274613536196417229379928410886682674 6\\
\log(3, 2\,962\,826 + 1\,627\,410\,i) &= 70530820889464725509484952408111454024 6\\
&\vdots\\
\log(4\,194\,313, 3\,987\,487 + 990\,581\,i) &= 77168988270700157762936609474336346218 7\\
\log(4\,194\,316, 2\,427\,954 + 2\,537\,863\,i) &= 13535723186646887259684604165458160945 64\\
\log(4\,194\,317, 1\,149\,909 + 103\,530\,i) &= 29756031028093138340311206649817815592 8
\end{aligned}$$

(les éléments de $\mathcal{F}$ sont donnés par les coordonnées des points correspondants de $\mathcal{H}$).

Il devient alors facile de calculer des logarithmes discrets de points quelconques de la courbe elliptique E. Pour illustrer ce fait, on génère un point aléatoire de E d'abscisse

$$\begin{aligned}
X_0 &= \sum_{j=0}^{5} \left(\left\lfloor \pi \cdot p^{j+1} \right\rceil \bmod p\right) i^{j \bmod 2} \theta^{j \bmod 3}\\
&= (593\,885 + 3\,175\,989\,i) + (199\,943 + 841\,508\,i)\theta + (411\,724 + 2\,224\,599\,i)\theta^2,
\end{aligned}$$

et on considère alors les points P_1, P_2, P_3, P_4, P_5, P_6 et P_7 dont les abscisses sont $X_0 + \delta$ pour les valeurs respectives de δ égales à 0, 1, 2, 3, 5, 11 et 12. Pour calculer les logarithmes discrets de

3. Ce calcul a pu être réalisé grâce aux ressources HPC du Centre de Calcul pour la Recherche et la Technologie du CEA, via l'allocation 2010-t201006445 du GENCI (Grand Équipement National de Calcul Intensif).

Points P_i	Mult. Conorm-Norm	Mult. Nagao	Points dans la décomposition					
$(X_0)^-$	2	341	370864^-	2471314^+	2517710^-	3195688^-	3512289^-	3700196^-
$(X_0+1)^-$	2	1664	1030818^+	2692469^+	2731382^-	3612676^+	3920772^-	4172888^+
$(X_0+2)^-$	4	85	399440^-	705045^-	901013^-	1366937^+	2079739^+	3419126^+
$(X_0+3)^+$	1	655	37064^+	2305706^+	2573803^+	2665635^-	3263560^-	4118343^-
$(X_0+5)^-$	2	72	311191^-	1011994^+	2166025^-	2649962^-	2777633^-	2900897^+
$(X_0+11)^-$	4	140	291295^+	518109^-	863097^-	1733917^+	3082470^-	3588239^+
$(X_0+12)^+$	3	1139	230555^-	385454^+	790502^-	985560^+	1466691^-	4062680^+

TABLE 8.3 – Détails des calculs de logarithmes discrets individuels.

ces points, on commence par les relever sur la jacobienne de $\mathcal{H}$ via l'application conorme-norme, ce qui prend un temps négligeable en Magma. Il est à noter à ce stade que si un point donné ne peut pas être relevé, il suffit de considérer à la place un petit multiple de ce point. On procède alors à la phase de descente (cf section 4.3.1) pour trouver une décomposition d'un multiple de chacun des relevés en somme d'éléments de la base de factorisation, et en déduire le logarithme discret de chacun des points P_i, $1 \leq i \leq 7$. En moyenne, on s'attend à tester de l'ordre de $6! = 720$ multiples pour chacun des points avant d'obtenir une décomposition ; en pratique, un maximum de 2 000 multiples par point a suffi pour trouver les 7 décompositions. Cette phase de descente prend environ 10 s par point sur un processeur Intel Core Duo à 2.6 GHz.

On donne les détails des calculs réalisés dans la table 8.3 : les points apparaissant dans les décompositions (ainsi que les points P_i) sont décrits par leurs abscisses ainsi qu'un signe $\pm$ indiquant si la partie "réelle" de leur ordonnée est donnée par un représentant positif ou négatif dans $[-q/2; q/2]$.

La structure du groupe des points $\mathbb{F}_{q^6}$-rationnels de la courbe E étant $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/(2\ell)\mathbb{Z}$, on multiplie tous les points P_i par un facteur 2 afin d'obtenir les logarithmes discrets modulo ℓ à constante près. Pour obtenir les logarithmes en base P_1, il suffit alors de diviser par le logarithme de P_1 :

$$\begin{aligned}
2 \cdot P_2 &= 771503218032571282830154288894596893 83 \cdot 2 \cdot P_1 \\
2 \cdot P_3 &= 277607596028887848748187645469867507392 \cdot 2 \cdot P_1 \\
2 \cdot P_4 &= 950556100385309676489669420946201334105 \cdot 2 \cdot P_1 \\
2 \cdot P_5 &= 317720686887855216292082605854593050146 \cdot 2 \cdot P_1 \\
2 \cdot P_6 &= 131228309389018991767706064340727221 4266 \cdot 2 \cdot P_1 \\
2 \cdot P_7 &= 119035784514809263757574253742461295 5882 \cdot 2 \cdot P_1.
\end{aligned}$$

En résumé, on est donc capable de résoudre avec l'attaque par recouvrement et décomposition, le problème du logarithme discret sur une courbe elliptique de 132 bits en à peine 30 h de temps de calcul réel sur moins de 200 cœurs (ce qui correspond à environ 3 700 h·CPU). Ces temps sont clairement plus rapides par plusieurs ordres de grandeur que ceux qu'on pourrait obtenir avec des algorithmes génériques. À titre d'exemple, on peut en effet comparer ce résultat à l'attaque (toujours en cours) du challenge Certicom ECC2K-130 utilisant l'implantation la plus efficace connue de Pollard-rho sur environ 3 000 processeurs à 3 GHz et qui est lancée depuis presque deux ans [BBB+09].

Annexe A

Classification des courbes hyperelliptiques de genre 3 bi-elliptiques

Soit $\mathbb{F}_{q^3}$ un corps fini de caractéristique impaire, et $\mathcal{H}_{|\mathbb{F}_q}$ une courbe hyperelliptique de genre 3, d'équation $y^2 = h$ où $h \in \mathbb{F}_q[x]$ est un polynôme de degré 7 ou 8. On note R l'ensemble des points de $\mathbb{P}^1$ au-dessus desquels la projection naturelle $\mathcal{H} \to \mathbb{P}^1$ est ramifiée, si bien que R s'identifie avec les abscisses des racines de h dans $\overline{\mathbb{F}_q}$, union $\{\infty\}$ si h est de degré 7. L'objectif est d'énumérer toutes les involutions $\phi \in \mathrm{Aut}_{\mathbb{F}_{q^3}}(\mathcal{H})$ telles que $\phi^\sigma \neq \phi$, dans le but d'obtenir des recouvrements $\pi : \mathcal{H}(\mathbb{F}_{q^3}) \to E(\mathbb{F}_{q^3})$, où la courbe E définie sur $\mathbb{F}_{q^3}$ est égale au quotient $\mathcal{H}/\phi$.

Soit ϕ un automorphisme de $\mathcal{H}$; son tiré en arrière ϕ^* est un automorphisme du corps de fonction $\overline{\mathbb{F}_q}(\mathcal{H})/\overline{\mathbb{F}_q}$. Comme $\overline{\mathbb{F}_q}(x)$ est l'unique sous-corps rationnel d'indice 2 de $\overline{\mathbb{F}_q}(\mathcal{H}) = \overline{\mathbb{F}_q}(x, y)$, il est nécessairement laissé globalement invariant par ϕ^*, qui induit donc un automorphisme de $\overline{\mathbb{F}_q}(x)/\overline{\mathbb{F}_q}$, correspondant à un automorphisme de $\mathbb{P}^1(\overline{\mathbb{F}_q})$ que l'on note encore ϕ. Si $\phi^*(x) = x$, i.e. ϕ est l'identité sur $\mathbb{P}^1$, il est facile de voir que ϕ est soit l'identité, soit l'involution hyperelliptique ι, qui sont définies sur $\mathbb{F}_q$; on ne s'intéressera pas à ces deux cas. Si ϕ est une involution de $\mathcal{H}$ différente de ι, elle induit donc une involution de $\mathbb{P}^1$; les seuls automorphismes d'un corps rationnel étant les homographies, on a alors

$$\phi^*(x) = \begin{cases} \frac{b}{x-a} + a, & a, b \in \overline{\mathbb{F}_q}, \\ \quad \text{ou} & \\ a - x, & a \in \overline{\mathbb{F}_q}, \end{cases}$$

cette forme étant complètement déterminée dès que l'on connaît l'image de trois points.

Par ailleurs une involution ϕ induit (comme tout automorphisme) une permutation de R, et par la remarque qui précède cette permutation détermine $\phi^*(x)$. On peut montrer que cette action sur R est sans point fixe. En effet, dans le cas contraire ϕ fixe au moins deux points de R, et quitte à faire un changement de variables (défini sur $\overline{\mathbb{F}_q}$) on peut supposer que ces deux points sont 0 et ∞, et que par conséquent $\phi^*(x) = -x$. Alors le polynôme h est de la forme $h(x) = xg(x^2)$ et l'élément $\phi^*(y) \in \overline{\mathbb{F}_q}(\mathcal{H})$ vérifie $\phi^*(y)^2 = \phi^*(xg(x^2)) = -xg(x^2) = -y^2$. Donc $\phi^*(y) = \sqrt{-1}\, y$, et $\phi^*(\phi^*(y)) = -y$, ce qui contredit le fait que ϕ est une involution. Réciproquement, si ϕ est une involution de $\mathbb{P}^1$ agissant sur R sans point fixe, elle se relève en une involution de $\mathcal{H}$. En effet, quitte à faire un changement de variables défini sur $\overline{\mathbb{F}_q}$ on peut encore supposer que $\phi^*(x) = -x$.

Comme 0 n'est pas dans R, le polynôme h est alors de la forme $h(x) = g(x^2)$ et on peut choisir $\phi^*(y) = y$, ce qui définit bien une involution. Le quotient de $\mathcal{H}$ par ϕ est dans ce cas la courbe elliptique d'équation $y^2 = g(x)$, l'application quotient étant donné par $\pi(x, y) = (x^2, y)$.

Dans le cas plus général où $\phi^*(x)$ est de la forme $\frac{b}{x-a} + a$, on a alors $\phi^*(y) = y\,\frac{b^2}{(x-a)^4}$, et l'application quotient $\pi : \mathcal{H} \to E = \mathcal{H}/\phi$ est donnée par $\pi(x, y) = (x + \phi^*(x), y/(x-a)^2)$. L'équation du quotient E est $y^2 = LC(h)\prod_{i=1}^{3}(x - (x_i + \phi(x_i)))$ si $R = \{x_1, x_2, x_3, \phi(x_1), \phi(x_2), \phi(x_3), \infty, \phi(\infty)\}$, et $y^2 = LC(h)\prod_{i=1}^{4}(x - (x_i + \phi(x_i)))$ si $R = \{x_1, x_2, x_3, x_4, \phi(x_1), \phi(x_2), \phi(x_3), \phi(x_4)\}$, $\infty \notin R$.

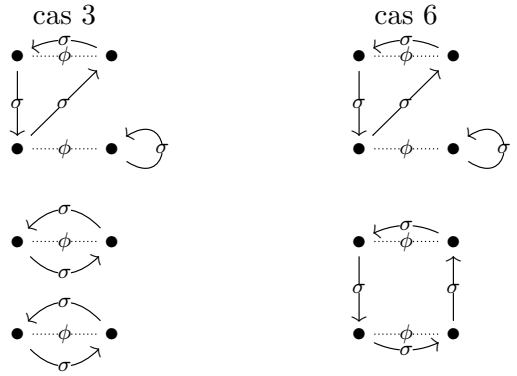
On se concentre maintenant sur le cas où ϕ est définie sur $\mathbb{F}_{q^3}$, de telle sorte que $\phi^{\sigma^3} = \sigma^3 \circ \phi \circ \sigma^{-3} = \phi$. Le polynôme h étant à coefficients dans $\mathbb{F}_q$, l'automorphisme de Frobenius $\sigma \in \mathrm{Gal}(\overline{\mathbb{F}_q}/\mathbb{F}_q)$ induit aussi une permutation de R. Soit n l'ordre de σ restreint à R. La restriction à R de l'involution ϕ, qui est définie sur $\mathbb{F}_{q^3}$, commute avec σ^3 et avec σ^n ; si n n'est pas un multiple de 3 alors ϕ commute avec σ sur R, donc sur tout $\mathbb{P}^1$, ce qui est exclu.

On peut alors distinguer 9 cas différents, selon la classe de conjugaison de $\sigma \in \mathfrak{S}(R)$. Par ailleurs, il est toujours possible sans perdre de généralités de conjuguer ϕ par n'importe quelle homographie à coefficients dans $\mathbb{F}_q$, ce qui revient à remplacer R par son image par cette homographie.

cas	nombre de cycles disjoints	tailles respectives des cycles
1	6	$3-1-\cdots-1$
2	5	$3-2-1-1-1$
3	4	$3-2-2-1$
4	4	$3-3-1-1$
5	3	$3-3-2$
6	3	$4-3-1$
7	2	$5-3$
8	3	$6-1-1$
9	2	$6-2$

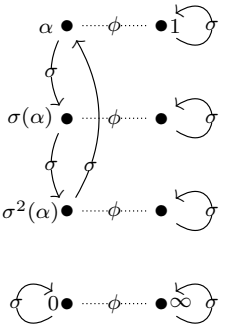
On commence par les cas faciles à éliminer (cas 3,6,7). On traite les cas restants dans l'ordre, les seuls cas délicats étant les 4 et 9.

- **Cas 7 :** ce cas peut être facilement éliminé. En effet comme ϕ commute avec σ^3, elle induit une permutation des 3 points fixes de σ^3 ; or ceci n'est pas possible puisque l'involution ϕ n'a pas de point fixe dans R.
- **Cas 3 et 6 :** un argument similaire permet d'éliminer ces deux cas. En effet ϕ doit permuter les 4 points fixes par σ^3, et donc se restreint à une permutation des 4 points restants.



L'involution ϕ permute dans le premier cas 4 points invariants par σ^2 (définis sur $\mathbb{F}_{q^2}$), ou dans le deuxième cas 4 points invariants par σ^4 (définis sur $\mathbb{F}_{q^4}$), et ϕ est alors définie à la fois sur $\mathbb{F}_{q^3}$, et sur $\mathbb{F}_{q^2}$ ou $\mathbb{F}_{q^4}$, donc sur $\mathbb{F}_q$, ce qui est exclu.

– **Cas 1 :** comme ϕ ne peut pas échanger plus de deux points de $\mathbb{F}_q$ (sinon elle serait définie sur $\mathbb{F}_q$), le graphe de l'action de ϕ et σ sur R a la forme suivante.

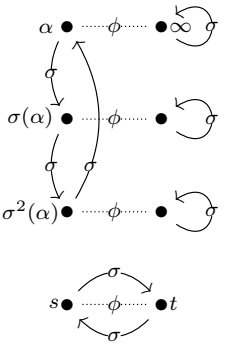


Quitte à faire un changement de variables, on peut supposer que ϕ échange α avec 1 et 0 avec ∞, ce qui signifie que ϕ est de la forme $\phi(x) = \frac{\alpha}{x}$. Comme $\phi(\sigma(\alpha))$ et $\phi(\sigma^2(\alpha))$ sont invariants par σ, on a nécessairement $\sigma(\alpha) = j\alpha$ où $j^2 + j + 1 = 0$, et $\alpha^3 = N_{K/k}(\alpha)$. Le polynôme h est alors de la forme $h(x) = x(x^3 - 1)(x^3 - \alpha^3)$. Du quotient $(w, z) = (x + \alpha/x, x^{-2}y)$, on déduit que la courbe E a pour équation

$$E : \ z^2 = w^3 - 3\alpha w - \alpha^3$$

dont le j-invariant est dans $\mathbb{F}_q$.

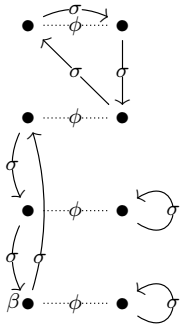
– **Cas 2 :** avec les résultats obtenus au cas 1, il devient facile d'éliminer ce cas.



Si l'on considère l'action de σ^2 au lieu de celle de σ (i.e. on considère $\mathcal{H}$ comme étant défini sur $\mathbb{F}_{q^2}$ au lieu de $\mathbb{F}_q$) et que l'on compose par l'homographie $\psi(x) = \frac{x-s}{x-t}$ (de telle sorte que $\phi' = \psi \circ \phi \circ \psi^{-1}$ est maintenant définie sur $\mathbb{F}_{(q^2)^3}$), on retombe sur le cas 1. Après quelques calculs, on trouve alors que $\alpha' = \psi(\alpha)$ vérifie à la fois $\alpha'\sigma(\alpha') = 1$ et $j^2\alpha'\sigma(\alpha') = 1$, ce qui est contradictoire. Ce cas n'est donc pas possible.

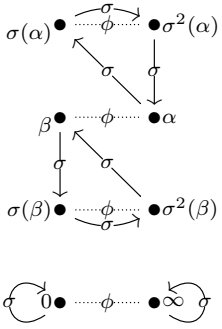
– **Cas 4 :** on distingue 6 sous-cas.

4.a.



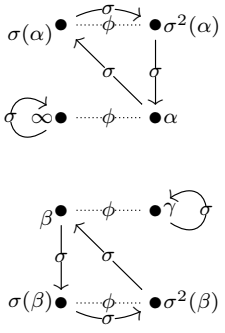
Ce cas n'est en fait pas possible : le groupe engendré par $\{\phi, \phi^\sigma, \phi^{\sigma^2}\}$ est de cardinalité 168, ce qui ne peut pas être un sous-groupe de $\mathrm{Aut}(\mathbb{F}_{q^3}(x, y)/\mathbb{F}_{q^3})/\langle\iota\rangle$ (voir [GSS05, Table 1]).

4.b.



L'involution ϕ est de la forme $x \mapsto \frac{D}{x}$ où $D = \alpha\beta = \sigma(\alpha)\sigma^2(\alpha) = \sigma(\beta)\sigma^2(\beta)$. Avec un calcul simple, on trouve alors que $\alpha = \beta$, ce qui n'est pas possible.

4.c.



Les relations sur α impliquent $\phi(x) = \frac{(\sigma^2(\alpha)-\sigma(\alpha)(\sigma(\alpha)-\alpha)}{x-\alpha} + \alpha$. Les relations sur β et γ sont alors automatiquement vérifiées. On retrouve ainsi l'involution bi-elliptique utilisée par Momose et Chao [MC05]. Le polynôme h est alors de la forme

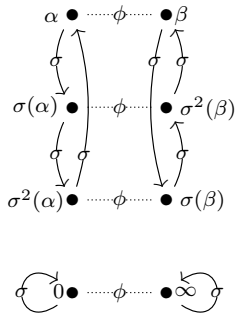
$$h(x) = N_\alpha(x)(x-\gamma)N_\beta(x) = N_\alpha(x)(F(x) - cN_\alpha(x)),$$

où N_α et N_β sont les polynômes minimaux respectifs de α et β sur $\mathbb{F}_q$, le polynôme F vaut $N_\alpha(x)(x+\phi(x)+\phi^\sigma(x)+\phi^{\sigma^2}(x))$, et γ et β sont deux racines de $F(x) - cN_\alpha(x)$, $c \in \mathbb{F}_q$.

L'équation du quotient est alors

$$\begin{aligned} E:\ z^2 &= (w-(\gamma+\beta))\left(w-(\sigma(\alpha)+\sigma^2(\alpha))\right)\left(w-(\sigma(\beta)+\sigma^2(\beta))\right) \\ &= (w^2 - 4\sigma(\alpha)\sigma^2(\alpha))(w-(\sigma(\alpha)+\sigma^2(\alpha))) + c(w-(\sigma(\alpha)+\sigma^2(\alpha)))^2. \end{aligned}$$

4.d.

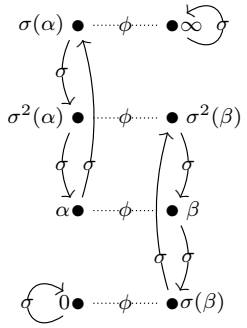


Comme dans le cas 4.b, ϕ est de la forme $x \mapsto \frac{D}{x}$ où $D = \alpha\beta = \sigma(\alpha)\sigma^2(\beta) = \sigma^2(\alpha)\sigma(\beta)$. Un calcul simple donne alors $\sigma(\alpha) = j\alpha$ et $\sigma(\beta) = j\beta$, où $j^3 = 1$. Le polynôme h est donc de la forme $h(x) = x(x^3-\alpha^3)(x^3-\beta^3)$. Du quotient $(w,z) = (x + \alpha\beta/x, x^{-2}y)$, on déduit que la courbe E a pour équation

$$E:\ z^2 = w^3 - 3\alpha\beta w - \alpha^3 - \beta^3,$$

qui a un j-invariant dans $\mathbb{F}_q$.

4.e.



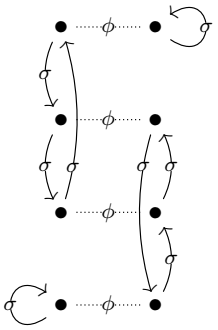
Après quelques calculs, on trouve que $\sigma(\alpha) = j\alpha, \beta = -\alpha/2$ et $\phi(x) = \frac{3}{2}\frac{(j\alpha)^2}{x-j\alpha} + j\alpha$. Le polynôme h est donc de la forme $h(x) = (x^3-\alpha^3)(x^3+\alpha^3/8)$. Du quotient $(w,z) = (x + \phi(x), (x-j)^{-2}y)$, on déduit que

$$E:\ z^2 = w^3 + j\alpha w^2 + \frac{j^2\alpha^2}{2}w + \frac{\alpha^3}{8}.$$

En posant $w' = w - j\alpha/3$, on obtient alors

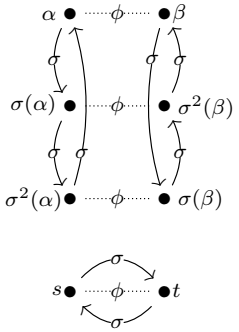
$$E:\ z^2 = w'^3 + \frac{\alpha^2 j^2}{6}w' + \frac{7\alpha^3}{216}$$

qui a pour j-invariant $2^{11}3^{-1} \in \mathbb{F}_q$.

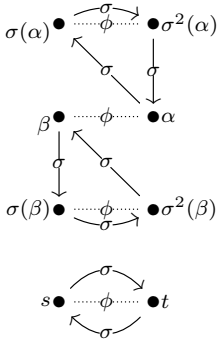
4.f. 

Ce cas n'est en fait pas possible : le groupe généré par $\{\phi, \phi^\sigma, \phi^{\sigma^2}\}$ est isomorphe à $(\mathbb{Z}/2\mathbb{Z})^3$, qui ne peut pas être un sous-groupe de $\mathrm{Aut}(K(x,y)/K)/\langle\imath\rangle$ selon [GSS05, Table1].

– **Cas 5 :** Comme σ^3 et ϕ commutent, ϕ induit une permutation des six points fixés par σ^3 ; en particuliers, ϕ échange les 2 points du 2-cycle. On distingue alors deux sous-cas :

5.a. 

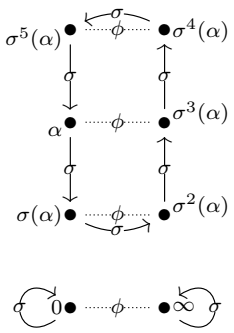
Ce cas n'est en fait pas possible. En effet, si l'on considère l'action de σ^2 au lieu de celle de σ et que l'on compose par l'homographie $\psi(x) = \frac{x-s}{x-t}$, on retombe sur le cas 4.d. Après quelques calculs, on trouve que $\alpha' = \psi(\alpha)$ doit vérifier à la fois $\alpha'\sigma(\alpha') = 1$ et $j\alpha'\sigma(\alpha') = 1$, ce qui est contradictoire.

5.b. 

Ce cas non plus n'est pas possible : en considérant à nouveau l'action de σ^2 au lieu de celle de σ, on retombe sur le cas 4.b qui n'était déjà pas possible.

– **Cas 8 :** ϕ échangeant les points fixe de σ, on distingue seulement deux sous-cas possibles.

8.a.



L'involution ϕ est de la forme $x \mapsto \frac{D}{x}$ où $D = \alpha\sigma^3(\alpha) = \sigma(\alpha)\sigma^2(\alpha) = \sigma^4(\alpha)\sigma^5(\alpha)$. En particulier $\alpha^{q^3-q^2-q+1} = 1$ et $\alpha^{q^6-1} = 1$, dont on déduit que $\alpha^{3(q^2-1)} = 1$. Comme $\sigma^2(\alpha) \neq \alpha$, ceci signifie que $\sigma^2(\alpha) = j\alpha$ où $j^2+j+1 = 1$. Le polynôme minimal de α sur $\mathbb{F}_q$ est donc de la forme $X^6 + aX^3 + b$ et h est tel que

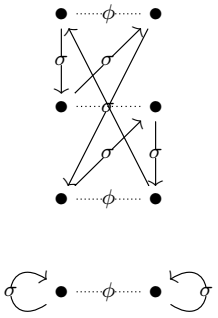
$$h(x) = x(x^6 + ax^3 + b).$$

Du quotient $(w, z) = (x + \phi(x), x^{-2}y)$, on déduit que le quotient bi-elliptique est

$$E : \ z^2 = w^3 - 3\alpha\sigma^3(\alpha)w - \alpha^3 - \sigma^3(\alpha)^3,$$

qui a un j-invariant dans $\mathbb{F}_q$.

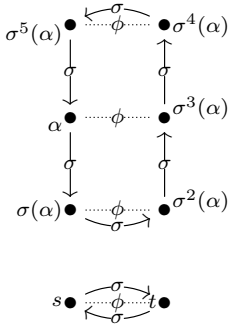
8.b.



Ce cas n'est en fait pas possible : si l'on considère l'action de σ^2 au lieu de σ, on retombe sur le cas 4.b qui était déjà impossible.

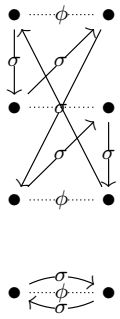
– **Cas 9 :** on distingue cinq sous-cas

9.a.

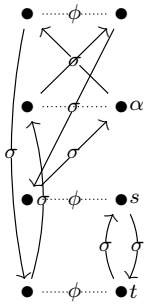


Ce cas n'est en fait pas possible. Si l'on considère l'action de σ^2 au lieu de σ et que l'on compose par l'homographie $\psi(x) = \frac{x-s}{x-t}$, on retombe sur le cas 4.d. Après quelques calculs, on peut montrer que $\alpha' = \psi(\alpha)$ et $\beta' = \psi(\sigma^3(\alpha))$ doivent vérifier à la fois $\beta'\sigma(\alpha') = 1$ et $\beta'\sigma(\alpha') = j$, ce qui est contradictoire.

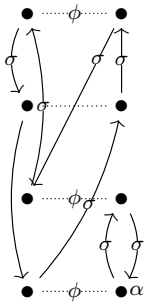
9.b.



Ce cas n'est pas possible : en regardant l'action de σ^2 au lieu de σ, on retombe sur le cas 4.b qui n'était déjà pas possible.

9.c. 

Ce cas n'est pas possible. En regardant l'action de σ^2 au lieu σ et en composant par l'homographie $\psi(x) = \frac{x-s}{x-t}$, on retombe sur le cas 4.e. Après quelques calculs, on trouve que $\alpha' = \psi(\alpha)$ doit vérifier à la fois $\alpha'\sigma(\alpha') = -2$ et $\alpha'\sigma(\alpha') = -2j$, ce qui est contradictoire.

9.d. 

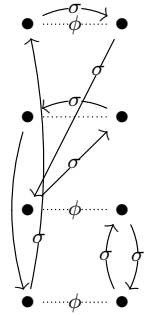
Le groupe engendré par $\{\phi, \phi^\sigma, \phi^{\sigma^2}\}$ est isomorphe à $(\mathbb{Z}/2\mathbb{Z})^2$. On a donc nécessairement $\phi \circ \phi^\sigma = \phi^\sigma \circ \phi = \phi^{\sigma^2}$ et $\phi(x) = \frac{(\sigma^2(u)-\sigma(u)(\sigma(u)-u)}{x-u} + u$, où $u \in \mathbb{F}_{q^3}$. On retrouve l'involution bi-elliptique utilisée par Momose et Chao [MC05]. Le polynôme h est de la forme

$$h(x) = N_\alpha(x)N_{\phi(\alpha)}(x) = F(x)^2 + bF(x)N_\alpha(x) + cN_\alpha(x)^2,$$

où $\alpha \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$, $b, c \in \mathbb{F}_q$, et N_α et $N_{\phi(\alpha)}$ sont les polynômes minimaux respectifs de α et $\phi(\alpha)$ sur $\mathbb{F}_q$.

L'équation du quotient est alors

$$E: \; z^2 = (w^2 - 4\sigma(\alpha)\sigma^2(\alpha))^2 + b(w^2 - 4\sigma(\alpha)\sigma^2(\alpha))(w - (\sigma(\alpha) + \sigma^2(\alpha))) + c(w - (\sigma(\alpha) + \sigma^2(\alpha)))^2.$$

9.e. 

Ce dernier cas n'est pas possible : le groupe engendré par $\{\phi, \phi^\sigma, \phi^{\sigma^2}\}$ est isomorphe à $(\mathbb{Z}/2\mathbb{Z})^3$, qui ne peut pas être un sous-groupe de $\mathrm{Aut}(\mathbb{F}_{q^3}(x,y)/\mathbb{F}_{q^3})/\langle\imath\rangle$ (cf. [GSS05, Table 1].

Notations utilisées

$[\![a;b]\!]$	intervalle $[a;b] \cap \mathbb{Z}$ des entiers compris entre a et b
$f \wedge g$	pgcd des polynômes f et g
$f \vee g$	ppcm des polynômes f et g
$K[\underline{X}]$	ensemble des polynômes multivariés en les variables $X_1, \ldots, X_n$
$\langle f_1, \ldots, f_r \rangle$	idéal engendré par les polynômes $f_1, \ldots, f_r$
$\mathcal{T}$	ensemble des monômes de $K[\underline{X}]$
$LM(f)$	monôme de tête du polynôme f
$LT(f)$	terme de tête de f
$LC(f)$	coefficient dominant de f
$\overline{f}^G$	forme normale de f modulo G
f^h	homogénéisé du polynôme f
g^*	déshomogénéisé du polynôme g
HF_I	fonction de Hilbert de l'idéal I
HP_I	polynôme de Hilbert de l'idéal I
$d_{reg}(I)$	degré de régularité de l'idéal I
$S(f,g)$	S-polynôme de f et g
$p = o_G(m)$	voir définition 2.1.4
$\mathrm{Syl}(f,g)$	matrice de Sylvester des polynômes f et g
$\mathrm{Res}_X(f,g)$	résultant des polynômes f et g par rapport à la variable X
$Sgn(r)$	signature du polynôme étiqueté r
$Poly(r)$	polynôme correspondant au polynôme étiqueté r
$\binom{n}{k}$	coefficient binomial de n et k
$\propto$	$P_1 \propto P_2$ si la résolution du problème P_2 permet celle du problème P_1
$\mathbb{P}^n$	espace projectif de dimension n
$\mathbb{V}(I)$	variété associée à l'idéal I
$I(V)$	idéal radical correspondant à la variété V
$V(k)$	ensemble des points k-rationnels de la variété V
$\mathrm{Gal}(\mathbb{K}/k)$	groupe de Galois absolu de k
$k[V]$	anneau de coordonnées affines
$k[V]_P$	anneau local de V en P
m_P	idéal maximal de $k[V]_P$
$k(V)$	corps de fonctions de la variété irréductible V

ϕ^*	tiré en arrière de la fonction ϕ
ϕ_*	poussé en avant de la fonction ϕ
$\mathrm{ord}_P(f)$	ordre d'annulation de la fonction f en P
$e_\phi(P)$	indice de ramification de ϕ en P
$\mathrm{Div}(\mathcal{C})$	groupe des diviseurs de $\mathcal{C}$
$\mathrm{Div}_k^0(\mathcal{C})$	groupe des diviseurs de degré 0 de $\mathcal{C}$ définis sur k
$\mathrm{div}(f)$	diviseur associé à une fonction f non nulle de $k(\mathcal{C})$
$\mathcal{L}(D)$	espace de Riemann-Roch associé au diviseur D
$\ell(D)$	dimension de l'espace de Riemann-Roch associé au diviseur D
$\mathrm{Pic}(\mathcal{C})$	groupe de Picard de $\mathcal{C}$
$\mathrm{Jac}_\mathcal{C}$	variété jacobienne de $\mathcal{C}$
f^σ	polynôme obtenu à partir de f en appliquant σ à tous ses coefficients
V^σ	variété image de V par σ
ι	l'involution hyperelliptique
$\log_2()$	logarithme en base 2
$\tilde{O}$	$f(n) = \tilde{O}(g(n))$ si $f(n) = O(g(n))$ à un facteur logarithmique de n près
Θ	$f(n) = \Theta(g(n))$ si $\exists c_1, c_2 > 0,\ c_1\lvert g(n)\rvert \leq \lvert f(n)\rvert \leq c_1\lvert g(n)\rvert$ pour n assez grand
Ω	$f(n) = \Omega(g(n))$ si $\exists c > 0,\ c\lvert g(n)\rvert \leq \lvert f(n)\rvert$ pour n assez grand
$L_N(\alpha, c)$	fonction d'estimation de complexités sous-exponentielles définie en (4.1)
$W_{K/k}(V)$	restriction de Weil de V relativement à l'extension $\mathbb{K}/k$
$\lfloor x \rceil$	entier le plus proche de x
$\mathrm{Aut}(K)$	groupe des automorphismes du corps K

Index

Bibliographie

[ABF09] D. Augot, M. Bardet, and J.-C. Faugère. On the decoding of binary cyclic codes with the Newton identities. *J. Symbolic Comput.*, 44(12) :1608–1625, 2009.

[ADH94] L. M. Adleman, J. DeMarrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 28–40. Springer, Berlin, 1994.

[Adl94] L. M. Adleman. The function field sieve. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 108–121. Springer, Berlin, 1994.

[AM93] A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61(203) :29–68, 1993.

[AMNS06] S. Arita, K. Matsuo, K.-I. Nagao, and M. Shimura. A Weil descent attack against elliptic curve cryptosystems over quartic extension fields. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E89-A :1246–1254, May 2006.

[AP10] M. Albrecht and J. Perry. F4/5. Preprint, available on arXiv :1006.4933, 2010.

[Arn03] E. A. Arnold. Modular algorithms for computing Gröbner bases. *J. Symbolic Comput.*, 35(4) :403–419, 2003.

[Ars05] G. Ars. *Applications des bases de Gröbner à la cryptographie.* PhD thesis, Université de Rennes I, 2005.

[Bar04] M. Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université Pierre et Marie Curie, Paris VI, 2004.

[BBB+09] D. V. Bailey, L. Batina, D. J. Bernstein, P. Birkner, J. W. Bos, H.-C. Chen, C.-M. Cheng, G. van Damme, G. de Meulenaer, L. J. D. Perez, J. Fan, T. Güneysu, F. Gurkaynak, T. Kleinjung, T. Lange, N. Mentens, R. Niederhagen, C. Paar, F. Regazzoni, P. Schwabe, L. Uhsadel, A. V. Herrewege, and B.-Y. Yang. Breaking ECC2K-130. Cryptology ePrint Archive, Report 2009/541, 2009.

[BCP97] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4) :235–265, 1997. Computational algebra and number theory (London, 1993).

[BF91] G. Björck and R. Fröberg. A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots. *J. Symbolic Comput.*, 12(3) :329–336, 1991.

[BFP10] L. Bettale, J.-C. Faugère, and L. Perret. Hybrid approach for solving multivariate systems over finite fields. *J. Math. Cryptology*, 3(3) :177–197, 2010.

[BK98] R. Balasubramanian and N. Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *J. Cryptology*, 11(2) :141–145, 1998.

[BLS01] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Advances in cryptology—ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 514–532. Springer, Berlin, 2001.

[BMMT94] E. Becker, M. G. Marinari, T. Mora, and C. Traverso. The shape of the shape lemma. In *Proceedings of ISSAC'94*, pages 129–133, Oxford, 1994. ACM.

[Bol01] B. Bollobás. *Random graphs*, volume 73 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2001.

[BP01] D. V. Bailey and C. Paar. Efficient arithmetic in finite field extensions with application in elliptic curve cryptography. *J. Cryptology*, 14(3) :153–176, 2001.

[BSS05] I. F. Blake, G. Seroussi, and N. P. Smart, editors. *Advances in elliptic curve cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2005.

[Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, Austria, 1965.

[Buc79] B. Buchberger. A criterion for detecting unnecessary reductions in the construction of Gröbner bases. In E. W. Ng, editor, *Proc. of the EUROSAM '79*, volume 72 of *Lecture Notes in Computer Science*, pages 3–21. Copyright : Springer, Berlin - Heidelberg - New York, 1979.

[Buc85] B. Buchberger. Gröbner bases : An algorithmic method in polynomial ideal theory. In N. Bose, editor, *Multidimensional systems theory, Progress, directions and open problems, Math. Appl. 16*, pages 184–232. D. Reidel Publ. Co., 1985.

[Can87] D. G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177) :95–101, 1987.

[CFA+06] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, editors. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications. Chapman & Hall/CRC, Boca Raton, FL, 2006.

[CKM97] S. Collart, M. Kalkbrener, and D. Mall. Converting bases with the Gröbner walk. *J. Symbolic Comput.*, 24(3-4) :465–469, 1997. Computational algebra and number theory (London, 1993).

[CKPS00] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology – EUROCRYPT 2000*, Lecture Notes in Comput. Sci., pages 392–407. Springer, 2000.

[CLO07] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer, New York, third edition, 2007.

[Cou01] N. Courtois. Efficient zero-knowledge authentication based on a linear algebra problem MinRank. In *Advances in cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 402–421, Berlin, 2001. Springer.

[CvA90] D. Chaum and H. van Antwerpen. Undeniable signatures. In *Advances in cryptology—CRYPTO 1989*, volume 435 of *Lecture Notes in Comput. Sci.*, pages 212–216, Berlin, 1990. Springer.

[CW90] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9(3) :251–280, 1990.

[Cza91] S. Czapor. A heuristic selection strategy for lexicographic Gröbner bases? In *Proceedings of ISSAC'91*, pages 39–48, New York, NY, USA, 1991. ACM.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, IT-22(6) :644–654, 1976.

[Die03] C. Diem. The GHS attack in odd characteristic. *J. Ramanujan Math. Soc.*, 18(1) :1–32, 2003.

[Die06] C. Diem. An index calculus algorithm for plane curves of small degree. In *Algorithmic number theory*, volume 4076 of *Lecture Notes in Comput. Sci.*, pages 543–557. Springer, Berlin, 2006.

[Die11] C. Diem. On the discrete logarithm problem in elliptic curves. *Compos. Math.*, 147(1) :75–104, 2011.

[DS03] C. Diem and J. Scholten. Cover attacks. Technical report, AREHCC project, 2003.

[Ebe83] G. L. Ebert. Some comments on the modular approach to Gröbner-bases. *SIGSAM Bull.*, 17(2) :28–32, 1983.

[EG02] A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arith.*, 102(1) :83–103, 2002.

[EGP11] C. Eder, J. Gash, and J. Perry. Modifying Faugère's F5 algorithm to ensure termination. To appear in ACM Commun. Comput. Algebra, 2011.

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in cryptology - CRYPTO 1984*, volume 196 of *Lecture Notes in Comput. Sci.*, pages 10–18. Springer, Berlin, 1985.

[EP10] C. Eder and J. Perry. F5C : a variant of Faugère's F5 algorithm with reduced Gröbner bases. *J. Symbolic Comput.*, 45(12) :1442–1458, 2010.

[ES02] A. Enge and A. Stein. Smooth ideals in hyperelliptic function fields. *Math. Comp.*, 71(239) :1219–1230 (electronic), 2002.

[Fau99] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *J. Pure Appl. Algebra*, 139(1-3) :61–88, June 1999.

[Fau02] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of ISSAC'02*, pages 75–83, New York, NY, USA, 2002. ACM.

[FGLM93] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symbolic Comput.*, 16(4) :329–344, 1993.

[FJ03] J.-C. Faugère and A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In B. Dan, editor, *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *Lecture Notes in Comput. Sci.*, pages 44–60. Springer Berlin / Heidelberg, 2003.

[FLDVP08] J.-C. Faugère, F. Levy-Dit-Vehel, and L. Perret. Cryptanalysis of MinRank. In *Advances in Cryptology—CRYPTO 2008*, volume 5157 of *Lecture Notes in Comput. Sci.*, pages 280–296, Berlin, Heidelberg, 2008. Springer-Verlag.

[Flo67] R. Floyd. Nondeterministic algorithms. *J. Assoc. Comput. Mach.*, 14 :636–644, 1967.

[FR94] G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206) :865–874, 1994.

[Fre98] G. Frey. How to disguise an elliptic curve (Weil descent). Talk at the 2nd Elliptic Curve Cryptography Workshop (ECC), 1998. Preprint, available at http://www.cacr.math.uwaterloo.ca/conferences/1998/ecc98/frey.ps.

[Fre05] D. Freeman. Pairing-based identification schemes. Technical report, Hewlett-Packard Laboratories HPL-2005-154, 2005.

[G03] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, second edition, 2003.

[Gau00] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In *Advances in cryptology—EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Comput. Sci.*, pages 19–34. Springer, Berlin, 2000.

[Gau08] P. Gaudry. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. *J. Symbolic Comput.*, 44(12) :1690–1702, 2008.

[GGV10] S. Gao, Y. Guan, and F. Volny IV. A new incremental algorithm for computing Gröbner bases. In *Proceedings of ISSAC'10*, pages 13–19, New York, NY, USA, 2010. ACM.

[GHS02a] S. D. Galbraith, F. Hess, and N. P. Smart. Extending the GHS Weil descent attack. In *Advances in cryptology—EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Comput. Sci.*, pages 29–44. Springer, Berlin, 2002.

[GHS02b] P. Gaudry, F. Hess, and N. P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, 15(1) :19–46, 2002.

[Giu84] M. Giusti. Some effectivity problems in polynomial ideal theory. In *EUROSAM 84 (Cambridge, 1984)*, volume 174 of *Lecture Notes in Comput. Sci.*, pages 159–171. Springer, Berlin, 1984.

[GJV10] R. Granger, A. Joux, and V. Vitse. New timings for oracle-assisted SDHP on the IPSEC Oakley 'Well Known Group' 3 curve. Announcement on the NBRTHRY mailing list, July 2010. http://listserv.nodak.edu/archives/nmbrthry.html.

[GM88] R. Gebauer and H. M. Möller. On an installation of Buchberger's algorithm. *J. Symbolic Comput.*, 6(2-3) :275–286, 1988.

[GMN+91] A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso. One sugar cube, please. In *Proceedings of ISSAC'91*, pages 49–54, New York, NY, USA, 1991. ACM.

[Gra10] R. Granger. On the Static Diffie-Hellman Problem on elliptic curves over extension fields. In *Advances in cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Comput. Sci.*, pages 283–302, 2010.

[GS91] A. García and H. Stichtenoth. Elementary abelian p-extensions of algebraic function fields. *Manuscripta Math.*, 72(1) :67–79, 1991.

[GS99] S. D. Galbraith and N. P. Smart. A cryptographic application of Weil descent. In *Cryptography and coding (Cirencester, 1999)*, volume 1746 of *Lecture Notes in Comput. Sci.*, pages 191–200. Springer, Berlin, 1999.

[GSS05] J. Gutierrez, D. Sevilla, and T. Shaska. Hyperelliptic curves of genus 3 with prescribed automorphism group. In *Computational aspects of algebraic curves*, volume 13 of *Lecture Notes Ser. Comput.*, pages 109–123. World Sci. Publ., Hackensack, NJ, 2005.

[GTTD07] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. A double large prime variation for small genus hyperelliptic index calculus. *Math. Comp.*, 76 :475–492, 2007.

[Har77] R. Hartshorne. *Algebraic geometry*. Springer-Verlag, New York, 1977. Graduate Texts in Mathematics, No. 52.

[Hes02] F. Hess. Computing Riemann-Roch spaces in algebraic function fields and related topics. *J. Symbolic Comput.*, 33(4) :425–445, 2002.

[Hes03] F. Hess. The GHS attack revisited. In *Advances in cryptology—EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Comput. Sci.*, pages 374–387. Springer, Berlin, 2003.

[Hes04] F. Hess. Generalising the GHS attack on the elliptic curve discrete logarithm problem. *LMS J. Comput. Math.*, 7 :167–192 (electronic), 2004.

[Hes05] F. Hess. Weil descent attacks. In *Advances in elliptic curve cryptography*, volume 317 of *London Math. Soc. Lecture Note Ser.*, pages 151–180. Cambridge Univ. Press, Cambridge, 2005.

[IET98] IETF. The Oakley key determination protocol, IETF RFC 2412, 1998.

[JL07] A. Joux and R. Lercier. Algorithmes pour résoudre le problème du logarithme discret dans les corps finis. In *Nouvelles Méthodes Mathématiques en Cryptographie*, Fascicules Journées Annuelles, pages 23–53. Société Mathématique de France, June 2007.

[JMS01] M. Jacobson, A. Menezes, and A. Stein. Solving elliptic curve discrete logarithm problems using Weil descent. *J. Ramanujan Math. Soc.*, 16(3) :231–260, 2001.

[JN03] A. Joux and K. Nguyen. Separating decision Diffie-Hellmann from computational Diffie-Hellman in cryptographic groups. *J. Cryptology*, 16(4) :239–247, 2003.

[JV10] A. Joux and V. Vitse. Elliptic curve discrete logarithm problem over small degree extension fields. Application to the static Diffie–Hellman problem on $E(\mathbb{F}_{q^5})$. Cryptology ePrint Archive, Report 2010/157, 2010.

[JV11a] A. Joux and V. Vitse. Cover and Decomposition Index Calculus on Elliptic Curves made practical. Application to a seemingly secure curve over $\mathbb{F}_{p^6}$. Cryptology ePrint Archive, Report 2011/020, 2011.

[JV11b] A. Joux and V. Vitse. A variant of the F4 algorithm. In A. Kiayias, editor, *Topics in cryptology—CT-RSA 2011*, volume 6558 of *Lecture Notes in Comput. Sci.*, pages 356–375, Berlin, 2011. Springer.

[KAF+10] T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, K., E. Thomé, J. Bos, W., P. Gaudry, A. Kruppa, P. Montgomery, L., D. A. Osvik, H. Te Riele, A. Timofeev, and P. Zimmermann. Factorization of a 768-bit RSA modulus. In T. Rabin, editor, *Advances in Cryptology—CRYPTO 2010*, volume 6223 of *Lecture Notes in Comput. Sci.*, pages 333–350. Springer Verlag, 2010.

[KFI+87] S. Katsura, W. Fukuda, S. Inawashiro, N. M. Fujiki, and R. Gebauer. Distribution of effective field in the Ising spin glass of the $\pm J$ model at $T = 0$. *Cell Biochem. Biophys.*, 11(1) :309–319, 1987.

[KM08] N. Koblitz and A. Menezes. Another look at non-standard discrete log and Diffie-Hellman problems. *J. Math. Cryptol.*, 2(4) :311–326, 2008.

[Knu69] D. E. Knuth. *The art of computer programming. Vol. 2 : Seminumerical algorithms.* Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont, 1969.

[Kob87] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48(177) :203–209, 1987.

[KPG99] A. Kipnis, J. Patarin, and L. Goubin. Unbalanced oil and vinegar signature schemes. In *Advances in cryptology—EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Comput. Sci.*, pages 206–222, Berlin, 1999. Springer.

[Kra26] M. Kraitchik. *Théorie des nombres. Tome II. Analyse indéterminée du second degré et factorisation.* IV + 252 p. Paris, Gauthier-Villars , 1926.

[KS99] A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Advances in Cryptology – CRYPTO 1999*, volume 1666 of *Lecture Notes in Comput. Sci.*, pages 19–30. Springer Berlin, Heidelberg, 1999.

[Laz83] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *Computer algebra (London, 1983)*, volume 162 of *Lecture Notes in Comput. Sci.*, pages 146–156. Springer, Berlin, 1983.

[LL93] A. K. Lenstra and H. W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1993.

[LL99] C. H. Lim and P. J. Lee. The Korean certificate-based digital signature algorithm. *Computers & Electrical Engineering*, 25(4) :249–265, 1999.

[LM94] A. K. Lenstra and M. S. Manasse. Factoring with two large primes. *Math. Comp.*, 63(208) :785–798, 1994.

[LO91] B. A. LaMacchia and A. M. Odlyzko. Computation of discrete logarithms in prime fields. *Des. Codes Cryptogr.*, 1(1) :47–62, 1991.

[Mac02] F. Macaulay. Some formulae in elimination. *Proceedings of London Mathematical Society*, pages 3–38, 1902.

[MC05] F. Momose and J. Chao. Scholten forms and elliptic/hyperelliptic curves with weak Weil restrictions. Cryptology ePrint Archive, Report 2005/277, 2005.

[MI88] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Advances in cryptology—EUROCRYPT 1988*, volume 330 of *Lecture Notes in Comput. Sci.*, pages 419–453. Springer, Berlin, 1988.

[Mil86a] V. S. Miller. Use of elliptic curves in cryptography. In *Advances in cryptology – CRYPTO 1985*, volume 218 of *Lecture Notes in Comput. Sci.*, pages 417–426. Springer, Berlin, 1986.

[Mil86b] J. S. Milne. Abelian varieties. In G. Cornell and J. H. Silverman, editors, *Arithmetic geometry (Storrs, Conn., 1984)*, pages 103–150. Springer-Verlag, New York, 1986.

[Mil86c] J. S. Milne. Jacobian varieties. In G. Cornell and J. H. Silverman, editors, *Arithmetic geometry (Storrs, Conn., 1984)*, pages 167–212. Springer-Verlag, New York, 1986.

[Mil04] V. S. Miller. The Weil pairing, and its efficient calculation. *J. Cryptology*, 17(4) :235–261, 2004. Version publiée d'un manuscript de 1986.

[MM84] H. M. Möller and F. Mora. Upper and lower bounds for the degree of Groebner bases. In *EUROSAM 84 (Cambridge, 1984)*, volume 174 of *Lecture Notes in Comput. Sci.*, pages 172–183. Springer, Berlin, 1984.

[MMDB08] M. S. E. Mohamed, W. S. A. E. Mohamed, J. Ding, and J. Buchmann. MXL2 : Solving polynomial equations over $GF(2)$ using an improved mutant strategy. In *PQCrypto 2008*, volume 5299, pages 203–215. Springer, 2008.

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

[MTW04] A. Menezes, E. Teske, and A. Weng. Weak fields for ECC. In T. Okamoto, editor, *Topics in cryptology—CT-RSA 2004*, volume 2964 of *Lecture Notes in Comput. Sci.*, pages 366–386. Springer, Berlin, 2004.

[MW00] U. M. Maurer and S. Wolf. The Diffie-Hellman protocol. *Des. Codes Cryptogr.*, 19(2-3) :147–171, 2000. Towards a quarter-century of public key cryptography.

[Nag10] K. Nagao. Decomposed attack for the Jacobian of a hyperelliptic curve over an extension field. In *Algorithmic Number Theory – ANTS-IX*, volume 6197 of *Lecture Notes in Comput. Sci.*, pages 285–300. Springer, Berlin, 2010.

[OW99] P. C. van Oorschot and M. J. Wiener. Parallel collision search with cryptanalytic applications. *J. Cryptology*, 12(1) :1–28, 1999.

[Pat96] J. Patarin. Asymmetric cryptography with a hidden monomial and a candidate algorithm for $\simeq 64$ bits asymmetric signatures. In *Advances in cryptology—CRYPTO 1996*, volume 1109 of *Lecture Notes in Comput. Sci.*, pages 45–60. Springer, Berlin, 1996.

[PH78] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Transactions on Information Theory*, IT-24 :106–110, 1978.

[Pol78] J. M. Pollard. Monte Carlo methods for index computation (mod p). *Math. Comp.*, 32(143) :918–924, 1978.

[Pom82] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In *Computational methods in number theory, Part I*, volume 154 of *Math. Centre Tracts*, pages 89–139. Math. Centrum, Amsterdam, 1982.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM*, 21(2) :120–126, 1978.

[SA98] T. Satoh and K. Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Comment. Math. Univ. St. Paul.*, 47(1) :81–92, 1998.

[Sat00] T. Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.*, 15(4) :247–270, 2000.

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

[Sch91] C. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3) :161–174, 1991.

[Sch95] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1) :219–254, 1995. Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993).

[Sch03] J. Scholten. Weil restriction of an elliptic curve over a quadratic extension. Preprint, available at http://homes.esat.kuleuven.be/~jscholte/weilres.pdf, 2003.

[Sem98] I. A. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Math. Comp.*, 67(221) :353–356, 1998.

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

[SG00] E. Schulte-Geers. Collision search in a random mapping : some asymptotic results. Talk at the Workshop on Elliptic Curve Cryptography (ECC), 2000. http://www.cacr.math.uwaterloo.ca/conferences/2000/ecc2000.

[Sha71] D. Shanks. Class number, a theory of factorization, and genera. In *1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, N.Y., 1969)*, pages 415–440. Amer. Math. Soc., Providence, R.I., 1971.

[Sho97] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in cryptology—EUROCRYPT 1997*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 256–266. Springer, Berlin, 1997.

[Sil86] J. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986.

[Sma99] N. P. Smart. The discrete logarithm problem on elliptic curves of trace one. *J. Cryptology*, 12(3) :193–196, 1999.

[Sma01] N. P. Smart. How secure are elliptic curves over composite extension fields? In *Advances in cryptology—EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Comput. Sci.*, pages 30–39. Springer, Berlin, 2001.

[Sma10] N. Smart (ed.). Yearly report on algorithms and keysizes. Technical report, European Network of Excellence in Cryptology II, 2010. www.ecrypt.eu.org/documents/D.SPA.13.pdf.

[ST89] T. Sasaki and T. Takeshima. A modular method for Gröbner-basis construction over $\mathbb{Q}$ and solving system of algebraic equations. *J. Inf. Process.*, 12(4) :371–379, 1989.

[Ste08] W. Stein. *Sage : Open Source Mathematical Software (Version 2.10.2)*. The Sage Group, 2008. http://www.sagemath.org.

[Sti93] H. Stichtenoth. *Algebraic function fields and codes*. Universitext. Springer-Verlag, Berlin, 1993.

[Str69] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13 :354–356, 1969.

[Tes01] E. Teske. On random walks for Pollard's rho method. *Math. Comp.*, 70(234) :809–825, 2001.

[Thé03] N. Thériault. Weil descent attack for Kummer extensions. *J. Ramanujan Math. Soc.*, 18(3) :281–312, 2003.

[Tra89] C. Traverso. Gröbner trace algorithms. In *Symbolic and algebraic computation (Rome, 1988)*, volume 358 of *Lecture Notes in Comput. Sci.*, pages 125–138. Springer, Berlin, 1989.

[Vit10] V. Vitse. F4 traces and index calculus on elliptic curves over extension fields. Talk at the 25th Elliptic Curve Cryptography Workshop (ECC), 2010. Preprint, available at http://2010.eccworkshop.org/Vitse.pdf.

[Vit11] V. Vitse. Cover and decomposition attacks. Talk at the Elliptic Curve Discrete Logarithm Workshop, 2011. École Polytechnique Fédérale de Lausanne, LACAL.

[Wei92] V. Weispfenning. Comprehensive Gröbner bases. *J. Symbolic Comput.*, 14(1) :1–29, 1992.

Résumé

Le problème du logarithme discret sur courbes elliptiques est à la base de nombreux protocoles cryptographiques, dans la mesure où on ne connaît jusqu'à présent aucun algorithme permettant de l'attaquer efficacement. Du point de vue de la cryptanalyse, certaines approches basées sur des méthodes de calcul d'indices, et s'appuyant sur la résolution de systèmes pour la recherche de relations, sont toutefois prometteuses.

La première partie de cette thèse est consacrée aux techniques de calcul de bases de Gröbner appliquées à la résolution de systèmes polynomiaux. Après une description détaillée des algorithmes F4 et F5 de Faugère considérés comme les plus performants actuellement, on présente et analyse une variante de l'algorithme F4, particulièrement utile pour la résolution de nombreux systèmes "similaires". Plusieurs exemples d'applications de ce nouvel algorithme sont donnés à la fois au domaine du calcul formel et de la cryptographie, montrant que pour certaines attaques algébriques, cette variante est plus efficace que F4 et F5.

Étant munis de ces nouveaux outils, on étudie dans la seconde partie le problème du logarithme discret sur courbes algébriques. Après une présentation rapide des attaques existantes sur ce type de courbes dans un contexte général, on s'intéresse plus particulièrement aux courbes elliptiques définies sur des extensions de corps finis. On donne ainsi une description complète des techniques GHS, puis des méthodes d'attaques par décomposition introduites par Gaudry et Diem. On présente notamment des variantes de ces méthodes de décompositions permettant, grâce aux outils introduits en première partie de cette thèse, de fragiliser le DLP (et des problèmes reliés) sur courbes elliptiques sur une gamme plus large d'extensions de corps finis. Enfin, une nouvelle approche combinant les attaques par recouvrement ainsi que les méthodes de décompositions est proposée : cette attaque permet entre autres de calculer complètement le logarithme discret sur des courbes elliptiques définies sur des extensions sextiques de taille jamais atteinte auparavant.

Mots-clefs : problème du logarithme discret, courbes elliptiques, courbes hyperelliptiques, calcul d'indices, bases de Gröbner.

Abstract

Up to now, very few algorithms exist that solve the discrete logarithm problem in the group of points of an elliptic curve defined over finite field. This problem is thus the keystone of many cryptographic protocols. However, some approaches based on index calculus methods and using polynomial system solving for relation search are promising for cryptanalysis, and we propose to study them in this Ph.D. thesis.

The first part of this work focuses on Gröbner basis computation for polynomial system solving. We detail Faugère's F4 and F5 algorithms, which are considered as references in this field, and then propose and analyze a variant of F4 devised for the resolution of many systems having the same shape. We show on several examples that in this context, this new algorithm outperforms both F4 and F5, and thus should be added to the cryptanalytic toolbox.

The second part is devoted to the study of the discrete logarithm problem on algebraic curves. We begin with a short review of existing attacks in a general context, then focus on elliptic curves defined over extensions of finite fields. A complete description of the GHS transfer techniques and of the decomposition attacks introduced by Gaudry and Diem is given. Notably, we present variants of these decomposition methods, enlarging the range of extension fields over which the elliptic curve DLP (and related problems) is weak. Finally, a new approach based on a combination of cover and decomposition attacks is proposed. In particular, this attack allows to compute discrete logarithms on elliptic curves defined over sextic extensions whose sizes had never been reached before.

Keywords : discrete logarithm problem, elliptic curves, hyperelliptic curves, index calculus, Gröbner bases.