Counting points on hyperelliptic curves in large characteristic : algorithms and complexity

Simon Abelard

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Comptage de points de courbes hyperelliptiques en grande caractéristique : algorithmes et complexité

THÈSE

présentée et soutenue publiquement le 7 septembre 2018

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Simon Abelard

Composition du jury

Président :	Guillaume Hanrot	Professeur, ÉNS Lyon
Rapporteurs :	Christophe Ritzenthaler	Professeur, Université Rennes 1
	Fréderik Vercauteren	Associate Professor, KU Leuven
Examinateurs :	Magali Bardet	Maîtresse de Conférences, Université de Rouen
	Elisa Gorla	Professeure, Université de Neuchatel
	Guillaume Hanrot	Professeur, ÉNS Lyon
Directeurs de thèse :	Pierrick Gaudry	Directeur de Recherche CNRS, Nancy
	Pierre-Jean Spaenlehauer	Chargé de Recherche Inria, Nancy

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

À mes grands-parents.

Remerciements

Théorème. *Pour toute thèse en position générique, les remerciements sont la partie la plus lue et la plus délicate à rédiger.*

Preuve. C'est de notoriété publique, d'ailleurs les thèses contenant cet énoncé forment un sous-ensemble dense de l'ensemble des thèses muni de la topologie de Zariski. □

Corollaire. *Malgré mes efforts, cette partie contient son lot de formulations approximatives et d'oublis qui sont de surcroît plus faciles à remarquer que les éventuelles erreurs tapies dans le chapitre 4 ou dans la section 5.4.*

Ces fondements théoriques solides étant posés, je sollicite l'indulgence de celles et ceux qui me liront concernant les erreurs qui se trouvent dans mes remerciements (ou ailleurs) ainsi que les répétitions peu élégantes du verbe remercier dans les quelques paragraphes qui suivent. J'en profite également pour préciser que l'ordre de mes remerciements est globalement peu significatif, même si j'ai tenté autant que possible de séparer ce qui relève du scientifique de ce qui relève du personnel.

Mes premiers remerciements sont pour Pierrick et Pierre-Jean qui m'ont proposé un sujet de recherche passionnant et dans lequel j'ai pu m'épanouir, notamment grâce à leur encadrement remarquable. Merci infiniment d'avoir guidé mes premiers pas dans le monde de la recherche par vos conseils avisés et par l'attention que vous avez portée à la préparation de mes divers exposés ainsi qu'à la rédaction de ce manuscrit. Merci pour tout ce que vous m'avez transmis, pour votre bienveillance et pour le plaisir que j'ai eu à travailler avec vous. Merci à Pierre-Jean pour son enthousiasme communicatif, ses bonnes questions («affine ou homogène ?») et pour m'avoir fait découvrir que le système polynomial c'est trop génial. Merci Pierrick d'avoir partagé ta culture et ton expertise dans le domaine des courbes, et de m'avoir fait découvrir [28] dans lequel nous nous sommes (re)plongés à deux reprises, avec un plaisir toujours renouvelé.

Je remercie Christophe Ritzenthaler pour avoir accepté d'être rapporteur de ma thèse, pour l'attention qu'il y a portée ainsi que pour les remarques et les discussions enrichissantes qui en ont découlé. I wish to thank Fréderik Vercauteren for accepting the task of reviewing my thesis, and for his thorough reading. Many thanks also to Elisa Gorla for being a member of my committee and attending my defense from overseas. Je remercie Magali Bardet dont la thèse m'a beaucoup aidé à enrichir, à clarifier et à réorganiser mes connaissances en matière de bases de Gröbner et qui a accepté de faire partie de mon jury. Merci également à Guillaume Hanrot d'être toujours fidèle au poste dans le jury d'un énième doctorant CARAMEL / CARAMBA. Merci enfin à Monique Teillaud d'avoir été ma référente de thèse au cours de ces trois ans.

Je remercie toute l'équipe CARAMBA pour cette ambiance agréable et stimulante : j'y ai rencontré des gens remarquables autant par leurs compétences scientifiques et techniques que par leurs qualités humaines. Même si j'ai dû m'habituer au troll alors que c'était tout sauf ma spécialité, je garderai un excellent souvenir de ces années passées avec vous. Merci d'avoir fait honneur à l'ADN Inria en encourageant ma fibre entrepreneuriale disruptive et en supportant mes nombreux pitchs (et je ne parle pas de la brioche fourrée) et autres idées de jeunes pousses dans des domaines allant des objets connectés pour chevaux à la blockchain du froid. Merci également d'avoir tous contribué à ma culture scientifique mais aussi culinaire, agricole, musicale, cinématographique et hippique. Merci en particulier à Cécile dont le chat m'a bien aidé à rédiger l'introduction, j'espère que tu y reconnaîtras son style littéraire et que ton cheval trouvera ça beau (vous l'avez ?).

En parlant d'ambiance, je remercie tous ceux qui ont fait vivre l'esprit du bureau A215 : Svyat, Élise, Shashank, Sandra et nos illustres prédécesseurs et notamment Hamza que je n'ai jamais rencontré mais dont le poster et l'héritage spirituel m'ont grandement influencés. Parmi les piliers de ce bureau, Laurent mérite des remerciements tous particuliers pour ses nombreux conseils, ses scripts et sa grande culture. Merci également à Paul et Simon avec qui j'ai partagé le bureau B225 pendant ma rédaction, ainsi qu'à Ludovic, Joseph, Itsaka, Ivan, Alicia et toute-s les Camarades du pique-nique des doctorant-e-s pour les nombreuses conversations aussi philosophiques qu'animées. Merci encore aux stagiaires que j'ai croisé-e-s, notamment Aude qui commence sa thèse tandis que je termine la mienne.

Merci également à tous les collègues avec qui j'ai eu d'agréables discussions, qu'elles soient mathématiques ou non. Il me serait difficile de tous vous citer, mais je tiens à remercier Maike Massierer, Jan Tuitman, Cyril Hugounenq, Alexandre Gélin, David Kohel, Ben Smith, Benjamin Wesolowski, Marius Vuille, Chloe Martindale, Daniel Lazard, Grégoire Lecerf, Reynald Lercier, Enea Milio, mon "grand-père" Mohab et bien d'autres. Un grand merci à Éric Schost pour son invitation à Waterloo en avril 2017 et ses efforts pour financer mon postdoc à venir. J'ai beaucoup appris lors de ma première visite et je suis impatient de retourner à Waterloo.

Au cours de ces trois ans, j'ai aussi eu l'opportunité d'enseigner aux Mines de Nancy, ce qui fut une expérience agréable et très enrichissante. Je remercie Antoine Henrot, pour m'avoir fait confiance, Bernardetta Addis grâce à qui j'ai découvert et apprécié la recherche opérationnelle, Yannick Toussaint avec qui j'ai été très content de partager bon nombre d'heures de TD, Frédéric Sur aussi bien pour son aide précieuse concernant les subtilités administratives que pour son expérience pédagogique ainsi que Guillaume Bonfante, Pierre-Etienne Moreau et Cédric Zanni pour les TP de python et leur organisation bien rodée. Merci enfin à tous mes élèves pour avoir servi de cobayes à mes expériences pédagogiques, j'espère qu'elles vous ont été profitables autant qu'à moi et que vous en garderez un bon souvenir.

Mais l'enseignement et la recherche ne sont pas uniquement le fait des enseignant-chercheurs eux-mêmes, aussi je voudrais remercier toutes les personnes qui m'ont grandement aidé en gérant les aspects pratiques et administratifs associés à ma recherche. À ce titre, je remercie Sophie, Emmanuelle, Christine, Laurence, Virginie et Françoise ainsi que tous les services de l'école doctorale, de l'Université de Lorraine, des Mines de Nancy et du LORIA qui m'ont aidé dans mes démarches ou qui ont contribué, parfois sans que j'en aie conscience, aux excellentes conditions de travail dont j'ai bénéficié pendant ma thèse. Un grand merci en particulier aux équipes du restaurant du centre (que tout le monde nous envie) et notamment à Isabelle, Tarek et Caroline pour leur énergie et leur gentillesse.

Bien avant ma thèse, j'ai croisé la route de personnes qui m'ont encouragé à poursuivre dans les sciences mathématiques, ou qui m'ont permis de clarifier mes projets d'études ou de recherches. Parmi eux, je remercie Olivier Leguay et Daniel Souquet grâce à qui j'ai réalisé à quel point les Mathématiques étaient un domaine vivant et rempli de liberté et d'opportunités. Je remercie mes enseignants de licence et de master pour leurs conseils et plus particulièrement Jean-Michel Morel, Bernard Landreau et Michael Harris avec qui j'ai effectué des stages qui ont été très formateurs et qui m'ont aidé dans mon orientation. Un grand merci enfin à Célestin Rakotoniaina qui a été mon professeur à une période charnière de mes études et qui est devenu un ami.

Les mots me manquent pour exprimer des remerciements à la hauteur de la gratitude que j'ai pour ma famille. Je pense notamment à mes parents qui m'ont fait grandir par leur dévouement, leur affection inconditionnelle et l'éducation qu'ils m'ont donnée. Votre implication dans mon parcours scolaire puis académique, votre soutien logistique à toute épreuve et la liberté que vous m'avez laissée dans mes choix personnels ont été déterminants au point que cette thèse est aussi

la vôtre.

Je dois aussi énormément à mes frères qui ont grandement contribué à l'environnement stimulant dans lequel j'ai grandi. Merci Guillaume d'avoir été mon premier prof de maths, pour les heures passées en jeux de stratégie qui m'ont appris la persévérance, et pour tout ce que tu m'as transmis du haut de tes dix ans de plus. Merci Sylvain pour nos nombreux échanges numériques *de qualitay* qui ont égayé mes études supérieures, pour ton hospitalité lors de mes passages en région parisienne et pour ta relecture attentive de ces remerciements et de mon résumé en français (on a frôlé la catastrophe !).

Je n'oublie pas non plus mes grands-parents pour tout ce qu'ils m'ont apporté. Aucun d'eux n'aura pu voir l'aboutissement de ce travail, mais leur souvenir ne m'a jamais quitté. Cette thèse leur est dédiée.

I am deeply grateful to my American family for the time we had the opportunity to spend together in Atlanta and Milwaukee during my thesis. Being with you gave me the peace of mind and the energy I needed to keep moving.

Z celého srdce děkuji paní doktorce Carole Wastiaux, která sledovala s porozuměním a osobním pochopením mou práci v posledních pěti letech. Tato dizertační práce a její autor jí vděčí za mnohé (merci à Lenka Froulíková pour la traduction).

Enfin, comme on dit dans My Little Pony, les amis c'est magique et c'est donc tout naturellement que je souhaite remercier les miens. Je pense en particulier à Ève pour notre trafic d'animaux mignons en tous genres ; aux camarades cachanais-e-s et notamment Lilian, Pierre, Alexandre et Édouard ; à tous mes rowers et pourvoyeurs de bons mots, d'informations insolites, décalées et trollesques parmi lesquels se distinguent Rémi et mon directeur de la publication Henri Vullierme. Merci à tous mes amis de prépa et d'avant avec qui j'ai gardé contact de façon plus ou moins dématérialisée, notamment Déborah, Kévin et Laurène, Matthias, Thomas et bien d'autres qui, je l'espère, ne m'en voudront pas de ne pas être nommés ici.

Contents

Introduction

Curves over finite fields and their applications

Algebraic curves have been part of the mathematical landscape for over 2000 years, from the foundations of geometry in the Antiquity to the proof of Fermat's last theorem in the late 1990's. Such curves are often described as the solution set of a polynomial system and can model various situations, hence their wide range of applications even outside mathematics. In this thesis, we focus on algebraic plane curves, i.e. curves given by an equation of the form $f(x, y) = 0$ with f a bivariate polynomial. A point of the curve corresponds to a solution of an associated equation, but we must be clear about what we call a solution: much to their dismay, the mathematicians of ancient Greece were confronted with the fact that even when considering equations with coefficients in $\mathbb{Z}$, the associated points may live outside $\mathbb{Q}$. We must therefore specify the field in which the coefficients of f live, which we call the base field of the curve, and consider points in the algebraic closure of this field. Some of the points on the curve may still belong to the base field, and they are called rational when it is the case. While the real field $\mathbb{R}$ seems quite a natural field to study plane curves and in particular to plot them, curves defined over finite fields also have many interesting applications and properties. In this thesis, we consider almost exclusively curves defined over a finite field of odd characteristic although we sometimes take advantage of special properties of reductions modulo primes of curves defined over $\mathbb{Q}$.

Algebraic curves over finite fields can lead to efficient algorithms used in practice for factoring integers and primality testing. Indeed, the elliptic curve method (ECM) of [93] is still competitive compared to algorithms based on the number field sieve for finding factors of size less than 64 bits. The elliptic curve primality proving (ECPP) introduced by Goldwasser and Kilian and improved by Atkin and Morain [65, 109] is still among the fastest algorithms to generate primality certificates and it was used recently to prove the primality of a 34987-bit integer [76]. Although the ECCP-based algorithms are efficient in practice, the complexity of ECPP is not proven. Using genus-2 curves, Adleman and Huang [4] designed a polynomial-time Las Vegas algorithm for primality proving. More general curves have also been investigated to achieve a deterministic polynomial-time algorithm for factoring polynomials over finite fields. One can also mention the use of interpolation on algebraic curves by Chudnovsky and Chudnovsky [31] in the late 1980's to study the complexity of multiplying polynomials over finite fields. This is still ongoing research and there is an important literature [118, 9] on how to improve this method, for instance by a careful choice of the interpolating curves. To this end, the curves are chosen to have many rational points, and [63] provides an open database of such curves over some finite fields.

The same goes for ECM, as families of curves are chosen to increase the efficiency of the algorithm, either because they are more likely to have a smooth cardinality [6, 10], or because they allow for faster arithmetic [108, 16]. Elliptic curves defined over finite fields have also been of interest to cryptographers as their rational points form a group in which computing discrete

logarithms is hard in general. They now represent a widespread standard which benefits from much smaller keysizes compared to RSA. The reason is that contrary to factoring integers or computing discrete logarithms in the multiplicative group of a finite field, there is still no subexponential algorithm for computing discrete logarithms on an elliptic curve. Yet, a result by Pohlig and Hellman shows in [116] that even an exponential algorithm turns out to be efficient if the elliptic curve has a smooth number of rational points. Therefore, elliptic curves must be carefully chosen for cryptographic applications, and in particular the number of their rational points has to be known.

As the theory around curves developed, other objects were designed or related to curves. Examples are the various zeta and L functions associated to curves, which are now central tools in modern number theory. Indeed, there are various examples of number-theoretical results that were achieved by proving analytical results for these complex functions, such as the Sato-Tate conjecture. This conjecture gives a result on the behavior of the statistical distribution of the number of rational points of the reduction modulo p of an elliptic curve over $\mathbb{Q}$ when p varies, and was proven circa 2005 [67, 32, 139]. For more general curves, work is in progress to formulate generalizations of the Sato-Tate conjecture such as [49]. To this end, heavy experiments are made and point-counting represents a major part of the computations [71].

All these applications entail different contexts, from the nature of the curves involved to their fields of definition. In this thesis, we focus on hyperelliptic curves given by an odd-degree model $y^2 = f(x)$, with f a monic squarefree polynomial of odd degree. The degree $\deg f = 2g + 1$ determines the genus g of the associated hyperelliptic curve which will be an important parameter throughout the whole manuscript. The two additional parameters p and n determine the base field $\mathbb{F}_{p^n}$ of the curve, and we set $q = p^n$ when only the size of the field matters. In the whole manuscript, we use the usual O notation, the $\widetilde{O}$ notation for the O notation in which we omit (poly)logarithmic terms, and O_g when we further omit all the terms depending only on g. Using fast arithmetic (see for instance [24]), we assume that field operations in $\mathbb{F}_q$ have a cost in $\widetilde{O}(\log q)$.

Schoof's algorithm

We have seen several reasons why knowing the number of rational points on an elliptic curve can be crucial. One approach is to find methods to build curves with a prescribed number of points, such as the CM-method of [6] used for cryptographic applications in [86]. Another way is to consider "random" curves and count their points until we are satisfied with the outcome. While there are low-brow methods for so doing, such as testing all the pairs $(x, y) \in \mathbb{F}_q$ and check if they satisfy the curve's equation, their complexities considerably limit their use. A groundbreaking progress was made by Schoof in 1985, who proposed in [127] an algorithm for counting points on elliptic curves in time polynomial in $\log q$. Although at that time his algorithm was not considered efficient enough for practical use, he set the path for numerous improvements and extensions that are now known as ℓ-adic algorithms. A few years later, Elkies and Atkin designed improvements [128] to Schoof's algorithm that contributed to its practicality and remarkable efficiency. Under the name SEA (Schoof-Elkies-Atkin), the variant of Schoof's algorithm is still used for generating cryptographic curves and successfully addresses the problem of counting points on elliptic curves.

The idea of Schoof's algorithm is to compute the number of rational points modulo prime numbers ℓ until the actual value can be recovered by the Chinese remainder theorem (CRT). Indeed, the Weil bounds imply that it lies in an interval of size $\lceil 4\sqrt{q} \rceil$ so that the number and maximal size of primes ℓ to consider is in $O(\log q)$. To obtain the modular information, Schoof

considers the action of the Frobenius endomorphism $\pi : (x, y) \mapsto (x^q, y^q)$ on the ℓ-torsion, i.e. the sets of points P such that ℓP is the point at infinity, which is the zero element for the addition on the curve. For $\ell \neq p$ a prime number, the ℓ-torsion is actually a vector space isomorphic to $(\mathbb{Z}/\ell\mathbb{Z})^2$. The action of the Frobenius endomorphism can therefore be represented by a 2×2 matrix, and its trace determines the number of rational points modulo ℓ. The bottleneck of this algorithm is the computation of π in the ℓ-torsion, which costs $\widetilde{O}(\ell^2 \log q)$ field operations. Taking into account the cost of such operations, the size of the largest ℓ and the number of ℓ, the overall complexity of Schoof's algorithm is in $\widetilde{O}(\log^5 q)$. The SEA improvement consists of replacing the ℓ-torsion by a subgroup isomorphic to $\mathbb{Z}/\ell\mathbb{Z}$ in which each operation costs $\widetilde{O}(\ell \log q)$ field operations, so that the SEA algorithm runs in time $\widetilde{O}(\log^4 q)$.

Jacobians of curves

For some applications such as cryptography, the natural extension of elliptic curves are not curves of larger genera because their rational points no longer form a group. A more suitable tool for this purpose is to consider the Jacobian of the curve, which is a group —actually an Abelian variety— built from formal sums of points on the curve. The same goes for the ℓ-torsion of an elliptic curve which has to be replaced by that of the Jacobian of the curve. In fact, we will see that determining the ℓ-torsion is a prominent step in order to extend Schoof's algorithm, and this relies extensively on arithmetic in Jacobians.

Although algorithms for group operations in non-hyperelliptic Jacobians have been designed in [73, 83], this thesis focuses on the hyperelliptic case because it greatly simplifies the arithmetic of the associated Jacobians, and in particular the description of the ℓ-torsion. Elements of genus-g hyperelliptic Jacobians can be represented by their Mumford form, which is a pair of polynomials of respective degrees at most g and $g-1$. Arithmetic on elements given in Mumford form is performed using Cantor's algorithm [27], for a space and time complexity quasi-linear in $g \log q$. Through binary exponentiation, Cantor's addition algorithm provides an efficient way to perform scalar multiplications in the Jacobian.

Counting points on curves

In the early 1990's, Pila [114] noticed that the theoretical machinery behind Schoof's algorithm still held in a much more general context. He therefore extended Schoof's algorithm into an algorithm for counting points on Abelian varieties and in particular on (Jacobians of) algebraic curves. The output of Pila's algorithm is not only the number of rational points, but the full characteristic polynomial of the Frobenius endomorphism, or equivalently the local zeta function of the curve. The complexity of Pila's algorithm is still polynomial in $\log q$ but depends on additional parameters of the input such as its genus / dimension in a much more critical way. This algorithm was not intended to be practical but considering the particular case of genus-2 hyperelliptic curves and using suitable tools from computer algebra to describe the torsion subgroups and the associated Frobenius action, a practical analogue of Schoof's algorithm was designed by Gaudry-Harley [57]. It was further improved by Gaudry and Schost to the point of using it to generate a cryptographic genus-2 curve [60, 62]. As in genus 1, it is also possible to build curves with a prescribed number of points, for example with the CM-method [146, 135, 44].

In the early 2000's, other methods also based on computing the action of (p-adic approximations of) lifts of the Frobenius endomorphism were developped, first by Satoh [125] for elliptic curves. This was later extended in a much broader context and many algorithms regrouped under the denomination of p-adic methods were designed, considering other lifts or their actions

on different spaces. Among the vast literature on the subject, one can point another p-adic approach for hyperelliptic curves based on Monsky-Washnitzer cohomology by Kedlaya [80] and its counterpart in characteristic 2 by Denef and Vercauteren [41], and further extensions to more and more general curves [30, 29, 142]. In characteristic 2, a variant of Satoh's algorithm was independently designed by Mestre [102], who proposed an expression of the Frobenius in terms of an arithmetic-geometric sequence which is still the fastest option for counting points on elliptic curves over $\mathbb{F}_{2^n}$. Also in [102], Mestre suggested an extension of his method to genus 2. This was further extending in two directions: either over field of (small) odd characteristic [94] or for curves of larger genus [121, 95].

An interesting fact is that these methods yield practical algorithms and that their complexity is polynomial in g and n but exponential in $\log p$, so that both the p-adic and ℓ-adic provide complementary approaches when either one of p or g is small. There is still no classical point-counting algorithm that runs in time polynomial both in g and $n \log p$[1], but Harvey designed in [70] an algorithm that, given a curve over $\mathbb{Q}$ as input, computes the zeta functions of its reduction modulo p for all primes p of good reduction lower than a bound N. This algorithm runs in time quasi-linear in N, meaning that the average time spent counting points on each reduction modulo p is polynomial in $\log p$ for each p. This is particularly relevant when running experiments for analogues of the Sato-Tate conjecture.

In this thesis we focus on the following problem, which we sometimes also call counting points although we retrieve more information than the number of rational points on the curve (or its Jacobian).

Computing local zeta functions of hyperelliptic curves. Given an odd prime power q, a positive integer g and a squarefree univariate polynomial $f \in \mathbb{F}_q[X]$ of degree $2\,g+1$, let $\mathcal{C}$ be the hyperelliptic curve with Weierstrass form $Y^2 = f(X)$. Compute the numerator $P_{\mathcal{C}} \in \mathbb{Z}[T]$ of the local zeta function of $\mathcal{C}$:

$$Z(\mathcal{C}/\mathbb{F}_q, T) = \exp\left(\sum_{i=1}^{\infty} \#\mathcal{C}(\mathbb{F}_{q^i}) \cdot \frac{T^i}{i}\right) = \frac{P_{\mathcal{C}}(T)}{(1-T)(1-qT)}.$$

Where $\mathcal{C}(\mathbb{F}_{q^i})$ is the set of points of $\mathcal{C}$ whose coordinates live in $\mathbb{F}_{q^i}$.

Torsion subgroups

A key ingredient to the ℓ-adic methods is the determination of the action of the Frobenius on the ℓ-torsion subgroups. In Schoof's algorithm, the ℓ-torsion of the input elliptic curve is the set of points whose abscissae are the roots of the so-called ℓ-division polynomial ψ_ℓ of degree $(\ell^2-1)/2$. Therefore, the action of the Frobenius endomorphism $\pi : (x, y) \mapsto (x^q, y^q)$ on the torsion can be computed by repeatedly squaring and reducing by the equations defining the ℓ-torsion: $y^2 = f(x)$ and $\psi_\ell(x) = 0$. In a more general context, Pila calls this step computing a low-degree representation of the Frobenius.

For elliptic curves, the division polynomials give a straightforward representation of the ℓ-torsion. For curves of larger genera, *a priori*, we do not have access to a representation that would allow us to compute a low-degree representation of the Frobenius by performing binary exponentiation in a quotient ring. This entails an additional step in which we compute a

[1] Allowing quantum primitives, such an algorithm was designed by Kedlaya in [81].

"nice representation" (e.g. a Gröbner basis) for the torsion ideal before using it to reduce the Frobenius.

In this thesis, we follow the approach of Gaudry-Harley-Schost [57, 60, 62] and first start by writing the equation $\ell D = 0$. To this end, we need a description of the multiplication by ℓ as a rational map. For P a point of an hyperelliptic curve, there are $2g + 2$ polynomials describing the Mumford form of the divisor $\ell(P - P_\infty)$ in the Jacobian. These polynomials introduced in [28] are called Cantor's ℓ-division polynomials and they extend the ℓ-division polynomial. Writing D, an element of the Jacobian, as a sum of points, we deduce a first way of describing the ℓ-torsion as the solution set of the system $\ell D = 0$.

Once this first system is computed, we solve it in order to have a representation of the ℓ-torsion in which we can compute the action of the Frobenius endomorphism. This accounts for most of the cost of our algorithms, both in theory and practice. We thus take particular care of the way we model the ℓ-torsion by polynomial systems and the techniques we use to solve them, as they have a significant impact on the final complexities and running time of our point-counting algorithms.

Solving polynomial systems

Given multivariate polynomials $f_1, \ldots, f_m$ in $K[x_1, \ldots, x_n]$, we want to find equations defining the ideal $I = \langle f_1, \ldots, f_m \rangle$ such that it becomes possible to perform arithmetic operations in $K[x_1, \ldots, x_n]/I$. In this thesis, the task is achieved for $I = I_\ell$, the ℓ-torsion ideal of a Jacobian by either computing a triangular form of the polynomial system $f_1 = 0, \ldots, f_m = 0$, or a geometric resolution, i.e. a parametrization of the coordinates of the solutions by the roots of a univariate polynomial. In both cases, we refer to this as solving the input polynomial system. The literature provides numerous ways of doing so, and we will use three of them depending on the complexity or performance they offer. This depends on many parameters such as the number of variables and the degrees of the polynomials f_i, but also on less conspicuous properties of the system such as its dimension, its degree (i.e. its number of solutions in $\bar{K}$ if it is finite) and some potential structural particularities. In this thesis, we solve systems that model subsets of ℓ-torsion subgroups, so we already know that they are zero-dimensional and we can bound their degrees by ℓ^{2g}.

For instance, the torsion of genus-2 curves involve bivariate polynomials for which a triangular form can be computed using bivariate resultants, which is currently the best option both in terms of asymptotic complexity and practical efficiency. In Chapter 6, we model the ℓ-torsion of a genus-3 hyperelliptic Jacobian by a trivariate polynomial system, which is put in triangular form by computing resultants. Although this gives satisfactory asymptotic complexity bounds, computing a Gröbner basis with the F4 algorithm [45] is much more efficient in practice so we used it instead of the resultants for practical experiments. However, although the complexity of the F4 and F5 algorithms [45, 46] are subject of thorough investigations [11, 12], none of the existing complexity bounds were sharp enough for us to use them both in theory and practice.

In Chapter 5, we model the ℓ-torsion of hyperelliptic curves of arbitrary genus in a different way, involving $O(g^2)$ variables instead of g. However, there are only g variables whose degree depends on ℓ, while the others have degrees in $O_g(1)$. In this case, the choice of the geometric resolution algorithm of [26, 64] was dictated by the necessity of invoking complexity results that take advantage of this particular multihomogeneous structure.

Contributions

This thesis focuses on ℓ-adic methods derived from Schoof-Pila's algorithm. A central question of the whole manuscript is the complexity of such methods and in particular the dependency on g of the exponent of $\log q$. The first contribution of this manuscript, to appear as [1], is a point-counting algorithm for hyperelliptic curves, whose complexity is such that this exponent asymptotically grows linearly in g when the characteristic p is large enough. This improves on previous results by Adleman and Huang [3] who proved that this exponent was in general polynomial in g and even quadratic in the case of hyperelliptic curves. The state of the art concerning this exponent is detailed in Table 1. To achieve this complexity result, our algorithm itself is no different from that of Pila but our complexity analysis benefits from a novel modelling of the ℓ-torsion by a structured polynomial system, as explained above. This structure is the key of the improvement, and performing our analysis without exploiting it yields a result similar to that of Adleman and Huang in $O\left((\log q)^{O(g^2 \log g)}\right)$. This involves some technicalities, however, as we must first ensure that our system satisfies some genericity hypothesis to invoke complexity bounds for the computation of a geometric resolution; also, our modelling involves in fact many polynomial systems to handle "special" torsion elements.

Table 1: Asymptotic complexity bounds for computing the local zeta function of a g-dimensional Abelian variety defined over $\mathbb{F}_q$

Authors (year)	Complexity	Context
Pila [114] (1990)	$O\left((\log q)^{g^{O(g)}}\right)$	Abelian varieties
Huang-Ierardi [75] (1998)	$O\left((\log q)^{g^{O(1)}}\right)$	Plane curves
Adleman-Huang [3] (2001)	$O\left((\log q)^{g^{O(1)}}\right)$	Abelian varieties
Adleman-Huang [3] (2001)	$O\left((\log q)^{O(g^2 \log g)}\right)$	Hyperelliptic curves
Chapter 5 (2017)	$\boldsymbol{O_g\left((\log q)^{O(g)}\right)}$	**Hyperelliptic curves**
Chapter 7 (2018)	$\boldsymbol{\widetilde{O}_\eta\left((\log q)^8\right)}$	**Hyp. curves with explicit RM**

Another aspect we study is the practicality of Schoof-Pila's algorithm in small genus, which goes along with the value of the exponent of $\log q$ for a fixed genus. Although Pila's algorithm seems unfit for straightforward implementation, what he calls a small representation of the Frobenius, i.e. the Frobenius modulo the ℓ-torsion ideal can be computed in practice using standard tools from computer algebra. This was studied and implemented in genus 2 by Gaudry, Harley and Schost in [57, 60, 62]. Due to the size of the objects to manipulate, the complexity is much larger than in genus 1 but the algorithm is practical enough so as to provide a cryptographic curve defined over a 128-bit prime field. In this thesis, we informally analyze the feasibility of designing such a secure genus-2 curve over a field of 192-bit characteristic, which seems quite unlikely at the moment. Curves equipped with an explicit and efficient real multiplication (RM) benefit from additional structure that is used in [59] to decrease the exponent of $\log q$ from 8 to 5, reaching a complexity similar to that of Schoof's algorithm.

One step further, the other main contribution within this manuscript deals with hyperelliptic curves of genus 3 [2]. Practical experiments in that case seem almost hopeless for primes $\ell > 3$.

However, for genus-3 hyperelliptic curves with explicit RM, the work of [59] extends modulo several additional subtleties with a complexity in $\widetilde{O}(\log^6 q)$, even lower than that of the general genus-2 case. As expected from such a result, the algorithm is quite practical, although efficiency requires some modifications compared to the version used to establish the complexity bound. In particular, we count points on a genus-3 hyperelliptic curve with RM defined over the prime field $\mathbb{F}_{2^{64}-59}$, which has a 192-bit Jacobian. Our algorithm can readily be turned into a point-counting algorithm for general genus-3 hyperelliptic curves (i.e. without explicit RM) with a much larger complexity in $\widetilde{O}(\log^{14} q)$, thus giving a partial answer for the complexity of the Schoof-Pila algorithm in genus 3. As in the genus-2 case, the bottleneck of our algorithm is the resolution of the polynomial system describing the ℓ-torsion. This system is trivariate but successive elimination using resultants is still sufficient to achieve our reference complexity which is the square of the degree of the ideal. In practice however, we computed a Gröbner basis using the F4 [45] and FGLM [47] algorithms because they were far more efficient, although their theoretical complexity is much harder to control in our case.

Since the literature presents numerous examples of RM-curves of any genus [87, 23, 43, 101, 138], it is quite natural to wonder what changes this additional structure brings to the asymptotic complexity when g is no longer fixed to 2 or 3. We therefore extended some results and methods of the genus-3 case to design a point-counting algorithm for hyperelliptic curves with explicit RM of arbitrary high genus. The main primitive we use is the computation of a geometric resolution for the kernel of an endomorphism of degree ℓ^2. This is done by adapting the machinery of Chapter 5 which was applied to the kernel of the multiplication by ℓ, itself being an endomorphism of degree ℓ^{2g}. The difference of degrees impacts our modelling by reducing the degrees of the equations from $O_g(\ell^3)$ to $O_g(\ell^{3/g})$. Therefore, after checking that the hypotheses still hold and applying the geometric resolution algorithm, we achieve a complexity in $O_g\left((\log q)^c\right)$, with c an absolute constant and O_g hiding a term that depends both on g and the ring by which the curve has RM. However, we emphasize that our algorithm is not polynomial both in g and $\log q$ because the factor hidden by the O_g-notation remains exponential in g. We nonetheless analyze the cause of that exponential dependency in the hope that further results might provide tighter complexity estimates for the exponential steps, or find a way to replace or remove them.

Table 2: Asymptotic complexities for computing the local zeta function of hyperelliptic curves of genus ≤ 3

Genus	Complexity	Authors (year)
$g = 1$	$\widetilde{O}(\log^5 q)$	Schoof [127] (1985)
$g = 1$	$\widetilde{O}(\log^4 q)$	Schoof-Elkies-Atkin [128] ($\sim$ 1990)
$g = 2$	$\widetilde{O}(\log^8 q)$	Gaudry-Harley-Schost [62] ($\sim$ 2000)
$\boldsymbol{g = 3}$	$\boldsymbol{\widetilde{O}(\log^{14} q)}$	**Chapter 6 (2018)**
$g = 2$ with RM	$\widetilde{O}(\log^5 q)$	Gaudry-Kohel-Smith [59] (2011)
$\boldsymbol{g = 3}$ **with RM**	$\boldsymbol{\widetilde{O}(\log^6 q)}$	**Chapter 6 (2018)**

Organization of the thesis

Chapter 1 gives definitions and an overview on (hyperelliptic) curves, arithmetic in their Jacobians and point-counting. We survey in deeper details some applications of point-counting and recall fundamental results that lie at the heart of the Schoof-Pila algorithms. Since modelling the ℓ-torsion by polynomial systems and controlling their degrees and structures are cornerstones of our contributions, Chapter 2 presents three techniques for solving polynomial systems, along with their complexities, that will be used in all the following chapters apart from Chapter 4. Chapter 3 reviews previous work on point-counting over genus-2 curves, and finishes with an updated analysis on prospective and change that occurred in the last years. Although most of its content was produced before this thesis, we emphasize on the parts that are later reused or adapted.

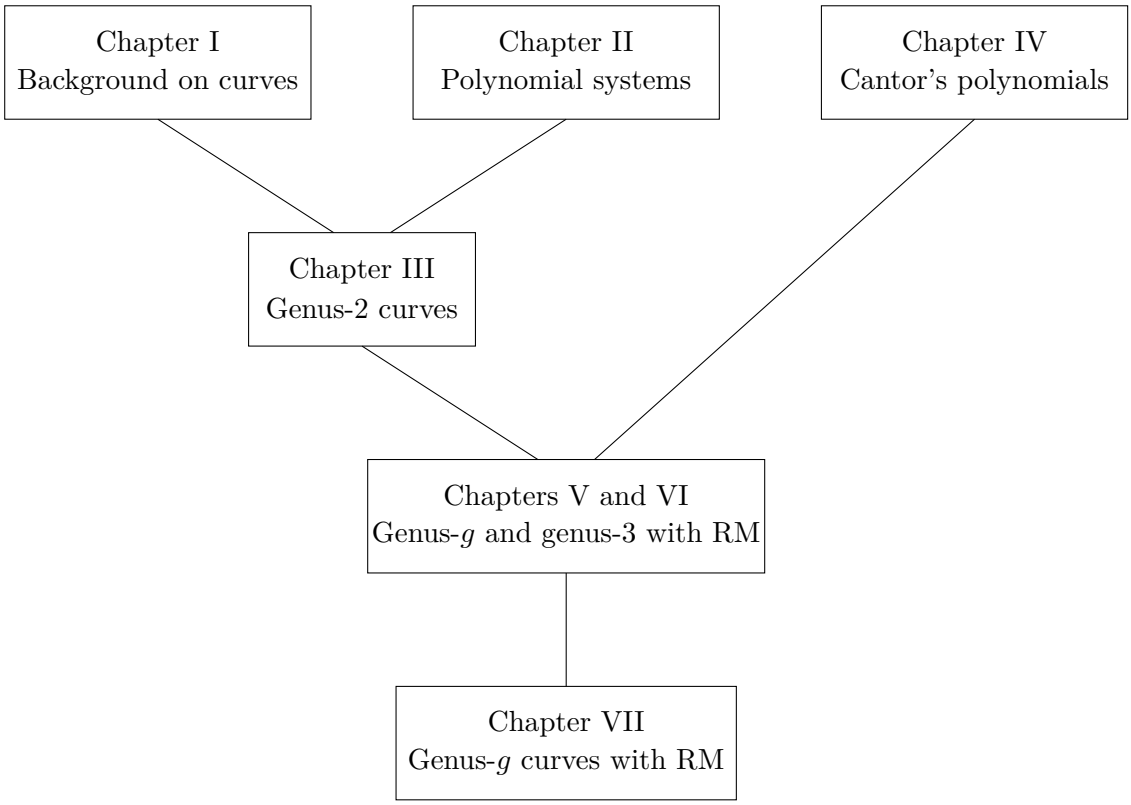

Figure 1: Chapters' dependencies

Chapter 4 provides bounds on Cantor's analogue to ℓ-division polynomials, which we use to bound the degrees of the systems modelling the ℓ-torsion. These bounds were originally proven in [1, Sec. 6] and [2, Sec. 6] but we regrouped them to form a chapter that does not rely on any other, and that can be skipped on first reading. Indeed, the results are restated when needed so that a reader willing to skip the proofs can avoid to read this technical chapter. Chapter 5 is based on [1] and presents a probabilistic algorithm for counting points on hyperelliptic curves over fields of sufficiently large characteristic with time and space complexity in $O_g\left((\log q)^{O(g)}\right)$. Chapter 6 follows [2] and deals with point-counting on genus-3 hyperelliptic curves, mostly for

curves with an explicit RM. Lastly, Chapter 7 combines the approaches of both Chapters 6 and 5 to improve the bounds of Chapter 5 in the case of hyperelliptic curves with explicit real multiplication (RM). To this end, we extend the algorithm and results of Chapter 6 in any genus and then prove that the systems involved in the extended algorithm satisfy genericity hypotheses similar to those of 5, so that the complexity bounds for computing a geometric resolution of these systems still apply. The complexity gain over the general case is then a pure consequence of the smaller degrees of the systems involved. Figure 1 sums up the dependencies between all the chapters.

Part I

Background and preliminaries

Chapter 1

Point-counting and applications

In this chapter, we introduce objects and concepts of algebraic geometry that are ubiquitous in this thesis such as curves, Jacobians and point-counting. We also recall fundamental results used by point-counting algorithms such as the Weil conjectures. Section 1.2 reviews the main families of point-counting algorithms and their principles, and Section 1.3 presents applications of point-counting.

In the whole manuscript, p stands for a prime number and $q = p^n$ is a power of that prime. We denote by $\mathbb{F}_p$ the finite field of cardinality p and by $\mathbb{F}_q$ its extension of degree n, up to isomorphism. In this first chapter, we consider objects (curves and varieties) defined over a perfect field K which will often, but not always, be a finite field in the other chapters. We denote by $\bar{K}$ the algebraic closure of K.

1.1 Background and definitions

1.1.1 Abelian varieties

Definition 1.1. *We denote by* $\mathbb{P}^n(\bar{K})$ *the quotient set* $\{(X_0 : X_1 : \cdots : X_n) \mid X_i \in \bar{K}, \exists\, j,\ X_j \neq 0\}/\sim$*, for the equivalence relation*

$$(X_0 : X_1 : \cdots : X_n) \sim (Y_0 : Y_1 : \cdots : Y_n) \Leftrightarrow \exists \lambda \in \bar{K},\ \forall i\ X_i = \lambda Y_i.$$

Definition 1.2. *We define* $\mathbb{A}^n(\bar{K}) = \{(x_1, \ldots, x_n) \mid x_i \in \bar{K}\}$ *the set of affine points.*

For an extension $L \subset \bar{K}$ of K, its absolute Galois group $\mathrm{Gal}(\bar{K}/L)$ acts coordinate-wise on $\mathbb{P}^n(\bar{K})$ and we define the set of L-rational points $\mathbb{P}^n(L)$ as the subset of $\mathbb{P}^n(\bar{K})$ fixed by this action. The same can be done to define $\mathbb{A}^n(L)$ the set of L-rational points of $\mathbb{A}^n(\bar{K})$. In other words, we have

$$\mathbb{P}^n(L) = \{(X_0 : X_1 : \cdots : X_n) \in \mathbb{P}^n(\bar{K}) \mid \exists \lambda \in \bar{K}\ \forall i,\ \lambda X_i \in L\},$$

and

$$\mathbb{A}^n(L) = \{(x_1, \ldots, x_n) \mid x_i \in L\}.$$

Both the affine and projective spaces can be endowed with the Zariski topology, for which we refer to [68, Chap. 1, Sec. 1 and 2]. A subset of $\mathbb{P}^n(\bar{K})$ (resp. $\mathbb{A}^n(\bar{K})$) is closed for the Zariski topology if and only if it is the set of simultaneous zeroes of homogeneous polynomials in $\bar{K}[X_0, \ldots, X_n]$ (resp. of polynomials in $\bar{K}[x_1, \ldots, x_n]$).

For S a set of polynomials in $\bar{K}[X_0, \ldots, X_n]$ (resp. $\bar{K}[x_1, \ldots, x_n]$), we denote $Z(S)$ the associated closed set in $\mathbb{P}^n(\bar{K})$ (resp. $\mathbb{A}^n(\bar{K})$).

Let V be a Zariski closed subset of either $\mathbb{P}^n(\bar{K})$ or $\mathbb{A}^n(\bar{K})$, and I_K be the associated ideal of (homogeneous) polynomials of either $K[X_0, \ldots, X_n]$ or $K[x_1, \ldots, x_n]$ vanishing on V. We say that V is defined over K if and only if $Z(I_K) = V$.

If I_K is a prime ideal, we say that V is irreducible over K. Note that irreducibility depends on the field K as, for instance, the ideal $I = \langle x_1^2 - 2x_2^2 \rangle$ is a prime ideal in $\mathbb{Q}[x_1, x_2]$ but it splits in $\mathbb{Q}(\sqrt{2})[x_1, x_2]$. When $I_{\bar{K}}$ is a prime ideal, we say that V is absolutely irreducible.

Definition 1.3. *A projective (resp. affine) variety over K is an irreducible projective closed set over K.*

Definition 1.4. *The dimension* $\dim(V)$ *of a variety V is the largest integer k such that there exist a chain $S_0 \supsetneq S_1 \supsetneq \cdots \supsetneq S_k$ of subsets of V that are closed and absolutely irreducible. A variety of dimension* 1 *is called a curve.*

Definition 1.5. *Let $V \subset \mathbb{A}^n(\bar{K})$ be an affine variety over K. It corresponds to a prime ideal $I(V) = \{f \in K[x_1, \ldots, x_n] \mid \forall P \in V, f(P) = 0\}$. Denote $K[V] = K[x_1, \ldots x_n]/I$, since it is an integral domain we can define its quotient field $K(V)$. The ring $K[V]$ and the field $K(V)$ are respectively called the coordinate ring and function field of V.*

For V a projective variety, defining $I(V)$ as the set of homogeneous polynomials vanishing on V, we similarly define the notion of coordinate ring $K[V]$ and we define $K(V)$ as the set of quotients of homogeneous polynomials of identical degrees.

Definition 1.6. *[34, Def. 4.33 & 4.34] A morphism φ from $\mathbb{A}^n(\bar{K})$ to $\mathbb{A}^1(\bar{K})$ is given by a polynomial $f \in K[x_1, \cdots, x_n]$ and defined by $\varphi : P = (a_1, \ldots, a_n) \mapsto f(a_1, \ldots, a_n) = f(P)$.*

Likewise, a morphism between $\mathbb{A}^n(\bar{K})$ and $\mathbb{A}^m(\bar{K})$ is given by a m-tuple of polynomials in $K[x_1, \cdots, x_n]$.

Definition 1.7. *[34, Def. 4.35] A K-rational morphism between two affine varieties $V \subset \mathbb{A}^n(\bar{K})$ and $W \subset \mathbb{A}^m(\bar{K})$ is defined as a morphism $\varphi : \mathbb{A}^n(\bar{K}) \to \mathbb{A}^m(\bar{K})$ between their associated affine spaces such that $\varphi(V) \subset W$.*

Definition 1.8 (Rational map)**.** *[34, Def. 4.40] Let U be a nonempty open set of an affine variety V, a rational map from V to $\mathbb{A}^1(\bar{K})$ with definition set U is a map $r_U : U \to \mathbb{A}^1(\bar{K})$ given by $r_U(P) = \psi(P)\varphi(P)^{-1}$ for some $\psi, \varphi \in \bar{K}[V]$ such that φ does not vanish on U.*

We say that two rational maps are equivalent if they coincide on the intersection of their respective definition sets. This defines an equivalence relation whose classes are called rational functions.

Proposition 1.9. *[34, Prop. 4.42] Let V be an affine variety, the set of rational functions on V is a field which is isomorphic to its function field $K(V)$.*

Definition 1.10 (Regularity at a point)**.** *[34, Def. 4.48] A rational function $f \in K(V)$ is regular at a point $P \in V$ if it has a rational map with set of definition containing P as a representative.*

As in Definition 1.6, considering tuples of rational maps and functions, these notions extend to rational maps and functions between varieties.

Replacing polynomials by homogeneous polynomials, and affine spaces by projective spaces, rational maps, rational functions and regularity are similarly defined for projective varieties.

Definition 1.11. *[34, Def. 4.53] An algebraic group $\mathcal{G}$ over K is an absolutely irreducible variety defined over K, along with*

- *a K-rational morphism $\oplus : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ for the group law,*
- *a K-rational morphism $\iota : \mathcal{G} \to \mathcal{G}$ for the inverse,*
- *a K-rational point $0 \in \mathcal{G}(K)$ for the neutral,*

such that $\oplus$ is associative, 0 is the neutral element for $\oplus$ and for any $e \in \mathcal{G}$, $\oplus(e, \iota(e)) = 0$.

For L an extension of K, denote $\mathcal{G}(L)$ the set of L-rational points, it is a group in which the group law is computed by evaluating the previous morphisms that are defined on K and do not depend on L.

Surprisingly, when $\mathcal{G}$ is a projective variety one can prove that the group law induced by $\oplus$ has to be commutative, leading to the following definition:

Definition 1.12. *An Abelian variety over a field K is a projective algebraic group over K.*

1.1.2 Curves and their Jacobians

In general, Abelian varieties are not easy objects to manipulate, as representing their elements may require a number of coordinates that is exponential in the dimension. See for instance [96] for group laws in Abelian varieties using theta functions. However, many examples of Abelian varieties come from simpler cases, for which this difficulty can be avoided. Let us now focus on a particular class of Abelian varieties: Jacobians of curves.

Definition 1.13. *Let P be a point on a curve $\mathcal{C}$. The set of rational functions that are regular at P is a subring of $K(\mathcal{C})$ denoted $\mathcal{O}_P$.*

Definition 1.14. *A point P on a curve $\mathcal{C}$ is called nonsingular if $\mathcal{O}_P$ is integrally closed. We say that P is singular otherwise and that $\mathcal{C}$ is a nonsingular or smooth curve if every point of $\mathcal{C}(\bar{K})$ is nonsingular.*

From now on, the word curve will refer to a smooth projective curve unless mentioned otherwise.

Definition 1.15. *Let $\mathcal{C}$ be a smooth projective and absolutely irreducible curve over K. The free Abelian group with basis $\mathcal{C}(\bar{K})$ is called the divisor group of $\mathcal{C}$, written $\mathrm{Div}_{\mathcal{C}}$. An element D of $\mathrm{Div}_{\mathcal{C}}$ has the form*

$$D = \sum_{P \in \mathcal{C}(\bar{K})} n_P P,$$

where the n_P are integers such that only a finite number of them are non-zero. We define $\mathrm{Supp}(D)$ the support of D as the set of points P such that $n_P \neq 0$ and the degree of D as $\deg D = \sum_{P \in \mathcal{C}(\bar{K})} n_P$.

Definition 1.16. *The set of degree-zero divisors forms a subgroup of $\mathrm{Div}_{\mathcal{C}}$ that we denote $\mathrm{Div}^0_{\mathcal{C}}$.*

Definition 1.17. *A divisor $D = \sum_{P \in \mathcal{C}(\bar{K})} n_P P$ is said to be effective if for all P we have $n_P \geq 0$, and for D and E two divisors, we write $D \geq E$ if $D - E$ is effective.*

Definition 1.18. *Let L be an intermediate field between K and $\bar{K}$, the action of $\mathrm{Gal}(\bar{K}/L)$ on $\mathcal{C}(\bar{K})$ induces an action on $\mathrm{Div}_{\mathcal{C}}$ (resp. $\mathrm{Div}^0_{\mathcal{C}}$). We define $\mathrm{Div}_{\mathcal{C}}(L)$ (resp. $\mathrm{Div}^0_{\mathcal{C}}(L)$) the subgroup of L-rational divisors (resp. degree-zero divisors) as the subgroup of $\mathrm{Div}_{\mathcal{C}}$ (resp. $\mathrm{Div}^0_{\mathcal{C}}$) fixed under that action.*

Definition 1.19. *Let L be an intermediate field between K and $\bar{K}$, let φ be a non-zero rational function in $L(\mathcal{C})$ and set $v_P(\varphi)$ equal to either the multiplicity of P as a zero of φ, minus its multiplicity as a pole of φ or zero if P is neither a pole nor a zero of φ. We define the associated divisor as $(\varphi) = \sum_{P \in \mathcal{C}(\bar{K})} v_P(\varphi) P$. A divisor of this form is said to be principal, and we denote $\mathrm{Pr}_{\mathcal{C}}(L)$ the group of principal divisors.*

Remark that a principal divisor has to be in $\mathrm{Div}^0_{\mathcal{C}}$ ([34, Prop. 4.104]), which allows the following definition:

Definition 1.20. *Let L be an intermediate field between K and $\bar{K}$, we define the degree-zero Picard group of $\mathcal{C}$ as the quotient $\mathrm{Pic}^0_{\mathcal{C}}(L) = \mathrm{Div}^0_{\mathcal{C}}(L)/\mathrm{Pr}_{\mathcal{C}}(L)$.*

Definition 1.21. *Let D be a divisor. We define the associated Riemann-Roch space as*

$$L(D) = \{\varphi \in K(\mathcal{C}) \mid (\varphi) \geq -D\}.$$

This is a vector space over K whose dimension is denoted $\ell(D)$.

Theorem 1.22 (Riemann's inequality)**.** *Let $\mathcal{C}$ be as in Definition 1.15. Then there exists an integer $g \geq 0$ such that for any $D \in \mathrm{Div}_{\mathcal{C}}$,*

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

The smallest such g is called the genus of the curve $\mathcal{C}$.

Theorem 1.23. *[106, Th. 1.1 & Prop. 2.1] Let $\mathcal{C}$ be a smooth projective and absolutely irreducible curve of genus $g > 0$ over K and L/K an extension. Then, there exists an Abelian variety J of dimension g over K such that $J(K) = \mathrm{Pic}^0_{\mathcal{C}}(\bar{K})^{\mathrm{Gal}(\bar{K}/K)}$ and such that $J(L) = \mathrm{Pic}^0_{\mathcal{C}}(L)$ as soon as $\mathcal{C}(L) \neq \emptyset$. This Abelian variety J is called the Jacobian (variety) of the curve $\mathcal{C}$, and it is denoted either $\mathrm{Jac}\,\mathcal{C}$ or $J_{\mathcal{C}}$.*

1.1.3 Hyperelliptic curves and their Jacobians

Performing explicit group operations in Jacobians of curves has drawn a lot of attention and many algorithms were proposed to achieve this goal with a polynomial-time complexity in g, such as [83, 73]. However, since the contributions presented in this thesis only apply to hyperelliptic curves, we do not give further detail on those algorithms and restrict to one of the simplest example of Abelian varieties: hyperelliptic Jacobians. In particular, following [27], we present a way to store elements of such Jacobians using $O(g)$ elements in the base field, and an algorithm to add points in the Jacobian in time quasi-linear in g.

Definition 1.24. *An elliptic curve over K is a nonsingular absolutely irreducible projective curve of genus 1 over K with at least one K-rational point.*

Definition 1.25. *A nonsingular projective curve $\mathcal{C}$ of genus $g > 1$ over K is called a hyperelliptic curve if there exists a function $x \in \bar{K}(\mathcal{C})$ such that the function field $K(\mathcal{C})$ is a separable quadratic extension of the rational function field $K(x)$.*

By [34, Theorem 4.122], if we characterize hyperelliptic curves by their affine plane parts, we can rewrite the previous definition in a more concrete way:

Definition 1.26. *Let K be a field of characteristic $\neq 2$, any plane affine curve given by an equation of the form*

$$\mathcal{C} : y^2 = f(x),$$

with f in $K[x]$ such that f is monic of degree $2g+1$ and squarefree is birationally equivalent to a hyperelliptic curve of genus g over K. Such hyperelliptic curves are called imaginary hyperelliptic curves.

In the remainder of this thesis, we will sometimes refer to "the hyperelliptic curve $\mathcal{C}$ of equation $y^2 = f(x)$". To be accurate, this refers to the nonsingular projective curve birationally equivalent to $\mathcal{C}$ which is indeed a hyperelliptic curve in the sense of Definition 1.25.

Note that when setting $g = 1$ in the equations of imaginary hyperelliptic curves, we fall back to the case of elliptic curves, which are famous for their use as cryptographic groups (i.e. groups in which the discrete logarithm problem is hard). Curves of genus 2 are no longer groups but their Jacobians also offer good candidates for cryptosystems, in a sense that we detail later on. The first requirement for constructing a cryptographic group is to provide an efficient way to represent and manipulate its elements: this is achieved thanks to the Mumford form for divisors and Cantor's algorithm to add and reduce them. Before giving details on this, we first review the specificities of hyperelliptic Jacobians.

Like elliptic curves, imaginary hyperelliptic curves have a unique K-rational point P_∞ at infinity and an involution sending an affine point (x, y) to its opposite $(x, -y)$. In what follows, we see that these additional properties give a simpler description of divisors on $\mathcal{C}$.

Definition 1.27. *Let $\mathcal{C}$ be a hyperelliptic curve and $D = \sum_{P \in \mathcal{C}(\bar{K})} n_P P$ be a divisor in $\mathrm{Div}^0_{\mathcal{C}}$. We say that D is semi-reduced if for any $P \neq P_\infty$ we have $n_P \geq 0$ and $n_P n_{-P} = 0$. Furthermore, we say that a semi-reduced divisor D is reduced if $\sum_{P \neq P_\infty} n_P \leq g$.*

Theorem 1.28. *Any element of $\mathrm{Pic}^0_{\mathcal{C}}$ is uniquely represented by a reduced divisor.*

The following theorem gives an efficient way of manipulating elements of $J_{\mathcal{C}}$ which is used in computer algebra systems.

Theorem 1.29. *Let $\mathcal{C}$ be a hyperelliptic curve of genus g given by an equation of the form $y^2 = f(x)$ with f a monic squarefree polynomial of degree $2g + 1$. Each element of $\mathrm{Pic}^0_{\mathcal{C}}(K)$ can be represented by a unique pair of polynomials $u, v \in K[x]$ where u is monic, $\deg v < \deg u \leq g$ and $u | v^2 - f$. The pair $\langle u, v \rangle$ is called the Mumford form of the divisor class.*

The link between the previous two representations is the following. If an element of $\mathrm{Pic}^0_{\mathcal{C}}$ is represented by a reduced divisor $\sum_{i=1}^{r} (P_i - P_\infty)$ where each P_i has coordinates (x_i, y_i), then its Mumford is $\langle u, v \rangle$ with u of degree r whose roots are the x_i's counted with multiplicities and v satisfying $v(x_i) = y_i$. The integer $r \leq g$ is called the weight of the divisor.

From the group isomorphism of Theorem 1.23 between $\mathrm{Pic}^0_{\mathcal{C}}(L)$ and $J_{\mathcal{C}}(L)$ for any $K \subset L \subset \bar{K}$, the Mumford form also gives a way of representing the points of the Jacobian of $\mathcal{C}$.

1.1.4 Arithmetic in hyperelliptic Jacobians

Algorithm 1, originally described by Cantor [27] in odd characteristic and later extended by Koblitz [85] to arbitrary fields, performs additions of reduced divisors given in Mumford form.

input : Two reduced divisors $D_1 = \langle u_1, v_1 \rangle$ and $D_2 = \langle u_2, v_2 \rangle$ on the curve $\mathcal{C} : y^2 = f(x)$, given in Mumford form.
output: The unique reduced divisor $D = D_1 \oplus D_2$.
Composition step:
Compute $d_1 = \gcd(u_1, u_2)$ and e_1, e_2 such that $d_1 = e_1 u_1 + e_2 u_2$
Compute $d = \gcd(d_1, v_1 + v_2)$ and c_1, c_2 such that $d = c_1 d + c_2(v_1 + v_2)$
$s_1 \leftarrow c_1 e_1,\ s_2 \leftarrow c_1 e_2,\ s_3 \leftarrow c_2$
$u \leftarrow \frac{u_1 u_2}{d^2}$, $v \leftarrow \frac{s_1 u_1 v_2 + s_2 u_1 v_1 + s_3(v_1 v_2 + f)}{d} \bmod u$
Reduction step:
while $\deg u > g$ **do**
  $U \leftarrow \frac{f - v^2}{u}$, $V \leftarrow -v \bmod U$
  $u \leftarrow U$, $v \leftarrow V$
end
Make u monic
return u,v

Algorithm 1: Cantor's algorithm

The algorithm we describe has a complexity in $\widetilde{O}(g^2)$, but this complexity can be reduced to $\widetilde{O}(g)$ by replacing the reduction step by a more efficient one inspired by the fast gcd algorithm. Since performing group operations in $J_{\mathcal{C}}$ is essential, faster algorithms have been designed in [90, 88, 55] to reduce the number of field operations involved in less general frameworks. In what follows, we make use of some of them in genus 2 and 3.

Another fundamental operation is scalar multiplication of a divisor. Once the addition is known, this can be done by a double-and-add approach but we emphasize here on the form of the result rather than the method to achieve it. In the case of elliptic curves, given an affine point $P \in \mathcal{C}$ of coordinates (x, y) and an integer $\ell > 1$, we have

$$\ell P = \left(x - \frac{\psi_{\ell-1}\psi_{\ell+1}(x)}{\psi_\ell^2(x)}, \frac{\psi_{2\ell}(x, y)}{2\psi_\ell^4(x)} \right),$$

where the ψ_i's are called division polynomials and they are defined inductively by $\psi_0 = 0$, $\psi_1 = 1$ and

$$\psi_{2m+1} = \psi_{m+2}\psi_m^3 - \psi_{m-1} - \psi_{m-2}\psi_{m+1}^2 \text{ for } m \geq 2,$$
$$\psi_{2m} = \frac{\psi_m}{2y}\left(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2\right) \text{ for } m \geq 3.$$

These polynomials have been generalized in [28] as follows: given $\ell > g$ and the weight-one divisor $D = P - \infty$ with $P \in \mathcal{C}$ of coordinates (x, y) the generic point, there exist $2g + 2$ polynomials $(d_i)_{0 \leq i \leq g}$ and $(e_i)_{0 \leq i \leq g}$ such that the Mumford form of ℓD is

$$\left\langle X^g + \sum_{i=0}^{g-1} \frac{d_i(x)}{d_g(x)} X^i, y \sum_{i=0}^{g-1} \frac{e_i(x)}{e_g(x)} X^i \right\rangle.$$

As in the elliptic case, there exist recurrence formulas for those division polynomials, which we use later to bound their degrees. To compute them, however, it is much simpler to directly multiply the generic affine point (x, y) by ℓ in the function field of the curve. In Chapter 4, we present two bounds for the degrees of Cantor's division polynomials, one for hyperelliptic curves of arbitrary genera and another sharper bound specific to genus-3 hyperelliptic curves.

1.1.5 Endomorphisms, torsion and Tate modules

To be more precise about what is point-counting and the main algorithms to do so, we give further theoretical background. We switch back to a broader context as we later present algorithms capable of counting points on Abelian varieties.

Definition 1.30. *Let A and B be two Abelian varieties over K, and let $\varphi \in \mathrm{Hom}_K(A, B)$ be a morphism of Abelian varieties, i.e. a morphism of varieties that is also a group homomorphism. We say that φ is an isogeny if the induced morphism $A(\bar{K}) \to B(\bar{K})$ is surjective and has a finite kernel. If there exists such an isogeny, we say that A and B are isogenous.*

Definition 1.31. *The degree of such an isogeny is defined as its degree as a rational map.*

Definition 1.32. *Given an isogeny φ of degree n between A and B, there exists a unique isogeny $\varphi^\vee$ of degree n between B and A such that $\varphi\varphi^\vee = [n]$. We call it the contragredient isogeny of φ.*

Definition 1.33. *The set $\mathrm{Hom}_K(A, A)$ of endomorphisms of A, denoted $\mathrm{End}_K(A)$, is a ring with composition as a multiplicative structure, called the endomorphism ring of A.*

Example 1.34. *Let A be an Abelian variety over $\mathbb{F}_q$. Let π be the Frobenius map $x \mapsto x^q$ of $\overline{\mathbb{F}_q}$, it extends to a map of projective spaces which stabilizes A, since A is defined over $\mathbb{F}_q$. The group law and zero-element of A are also defined over $\mathbb{F}_q$ so π is also an endomorphism for the group structure of A. Thus, $\pi \in \mathrm{End}_{\mathbb{F}_q}(A)$ can be seen as an endomorphism called the Frobenius endomorphism.*

Another common endomorphism is the aforementioned scalar multiplication that we denote $[\ell]$. We say that an element of $\ker[\ell]$ is an ℓ-torsion point and denote $A[\ell]$ the ℓ-torsion, i.e. the elements of $A(\bar{K})$ that vanish after multiplication by ℓ. This set is at the heart of Schoof-like algorithms and so is the following statement about its structure.

Proposition 1.35. *[34, Th. 4.73] Let A be an Abelian variety of dimension g defined over K of positive characteristic, and let n be an integer coprime to the characteristic of K. Then $A[n]$ is a $\mathbb{Z}/n\mathbb{Z}$-module isomorphic to $(\mathbb{Z}/n\mathbb{Z})^{2g}$.*

Note that it is important to highlight the fact that we consider the torsion elements in the algebraic closure, for they have no reason to be rational, and in general they live in (large) extensions of the base field.

In what follows, let ℓ be a prime number different from the characteristic. For any positive k, $[\ell]A[\ell^{k+1}] = A[\ell^k]$. Thus, the groups $A[\ell^k]$ form a projective system, which brings us to the following definition.

Definition 1.36. *Let ℓ be a prime different from $\mathrm{char}(K)$, the ℓ-adic Tate module of A is defined as $T_\ell(A) = \varprojlim A[\ell^k]$.*

We have seen that for n coprime to $\mathrm{char}(K)$, $A[n]$ has a structure of free $\mathbb{Z}/n\mathbb{Z}$-module of dimension $2g$, from which we deduce that $T_\ell(A)$ is a free $\mathbb{Z}_\ell$-module, also of dimension $2g$. Thus, $\mathrm{Aut}(T_\ell(A))$ and $\mathrm{Aut}(A[n])$ can be respectively identified with $\mathrm{GL}_{2g}(\mathbb{Z}_\ell)$ and $\mathrm{GL}_{2g}(\mathbb{Z}/n\mathbb{Z})$.

By acting on each $A[\ell^k]$, the Frobenius endomorphism acts on the $\mathbb{Z}_\ell$-module $T_\ell(A)$, and we can extend its action to the $2g$-dimensional $\mathbb{Q}_\ell$-vector space $T_\ell(A) \otimes_{\mathbb{Z}_\ell} \mathbb{Q}_\ell$. This action can be represented by a square matrix of size $2g$ whose characteristic polynomial we denote χ_ℓ.

Theorem 1.37. *[34, Lem. 5.71] The polynomials χ_ℓ have integer coefficients which are independent from ℓ. Their common value χ is called the characteristic polynomial of the Frobenius endomorphism.*

Note that this section relies on some powerful theoretic results that we do not want to linger on. In what follows we will mostly consider actions of the Frobenius on subspaces such as the ℓ-torsion, on which there are more elementary definitions. We invite the interested reader to look for more detailed information on this subject in [111, Sec. 19].

1.1.6 Real multiplication

Definition 1.38. *We say that an Abelian variety is simple when it has no proper non-zero Abelian subvariety.*

Proposition 1.39. *[137] Denote $\mathrm{End}_K^0(A) = \mathrm{End}_K \otimes_{\mathbb{Z}} \mathbb{Q}$. If A is simple, then $\mathrm{End}_K^0(A)$ is a skew field.*

Definition 1.40. *Let F be a totally real number field, we say that A has real multiplication (RM) by F if there exists an embedding $F \hookrightarrow \mathrm{End}_K^0(A)$.*

Likewise, we say that A has RM by a subring R of a totally real number field if there exists an embedding $R \hookrightarrow \mathrm{End}_K(A)$.

Previous examples of endomorphisms highlighted the fact that if A is nonzero, then $\mathbb{Z}$ is always a subring of the ring $\mathrm{End}_K(A)$. When K is a finite field, this is also true for $\mathbb{Z}[\pi]$ with π the Frobenius endomorphism.

This definition may seem tautological as every Abelian variety has RM by $\mathbb{Z}$. In Section 3.1.2 and Chapter 6, we ask for hyperelliptic Jacobian with RM by an order $\mathbb{Z}[\eta]$ satisfying further constraints.

1.2 Point-counting

1.2.1 Definitions

Definition 1.41 (Local zeta function)**.** *Let $\mathcal{C}$ be a nonsingular projective algebraic curve over a finite field $\mathbb{F}_q$, the (local) zeta function of $\mathcal{C}$ is defined as the formal power series in $\mathbb{Q}[[t]]$:*

$$Z(t) = \exp\left(\sum_{k\geq 1} \#\mathcal{C}(\mathbb{F}_{q^k})\frac{t^k}{k}\right).$$

In what follows, most point-counting algorithms actually compute the whole zeta function of the input curve instead of simply computing $\#\mathcal{C}(\mathbb{F}_q)$ or $\#J_{\mathcal{C}}(\mathbb{F}_q)$. The reason is that these algorithms strongly rely on the fact that zeta functions satisfy remarkable properties, as conjectured by Weil in 1949 and later proved by Dwork, Grothendieck and Deligne.

The Weil conjectures can be summed up by the following three properties:

- $Z(t) \in \mathbb{Q}\,[[t]]$ is a rational function
- $Z(t)$ verifies a functional equation
- the numerator of $Z(t)$ is a polynomial in $\mathbb{Z}[t]$ whose roots are algebraic integers of norm $1/\sqrt{q}$.

For counting points, we use the following consequences of the Weil conjectures.

Proposition 1.42. *The zeta function $Z(t)$ of a nonsingular projective algebraic curve $\mathcal{C}$ is a rational fraction of the form*

$$Z(t) = \frac{L(t)}{(1-t)(1-qt)},$$

where $L = a_0 + \cdots + a_{2g}t^{2g}$ is a degree $2g$ polynomial whose coefficients a_i are integers such that $a_0 = 1$, $a_{2g} = q^g$ and

$$\forall i \leq g, a_{2g-i} = q^{g-i}a_i \text{ and } |a_i| \leq \binom{2g}{i} q^{i/2}.$$

We have reduced the problem to computing the polynomial L, but we can get even more information on this polynomial by relating it further to the curve. *In fine*, we can translate all these properties into efficient point-counting algorithms.

Lemma 1.43. *The polynomial L is the reciprocal polynomial of the characteristic polynomial χ of the Frobenius endomorphism, as defined in Definition 1.37.*

This lemma is the cornerstone to all the algorithms that we discuss next, each one follows the same principle indeed: deducing χ from the characteristic polynomial of the action of the Frobenius on some spaces, of course provided that we can recover the actual χ from the partial information obtained.

1.2.2 Algorithms

In this section, we review the main families of algorithms for counting points on hyperelliptic curves, as well as their complexities. Note that the input curve is given by a degree $2g+1$ polynomial in $\mathbb{F}_q[X]$, with $q = p^n$. This input has a bit-size in $O(ng \log p)$ which is why $ng \log p = g \log q$ is the reference when we give complexity estimates. Therefore, an algorithm in $O(p)$ will be called exponential. See for instance [56] for a survey on the subject, along with record computations.

Exhaustive search

Since by Proposition 1.42 we are looking for a finite number of bounded integers, an algorithm that comes to mind would be to simply try all possibilities. To do so, one can try all the finitely many possibilities for χ until the characteristic equation $\chi(\pi) = 0$ is satisfied in the Jacobian. This amounts to a searchspace of size determined by the Weil bounds, at least in $O(q^{g/2})$. The complexity is therefore both exponential in g and $\log q$.

The birthday-paradox approach

Replacing exhaustive search by a birthday-paradox approach, this running-time can be reduced to the square root of the size of the search space, but there is also a memory cost of similar magnitude. This was improved using distinguished points as in [99, 61], increasing the running time by a constant factor but making the memory requirements negligible.

This approach is exponential in both $\log q$ and g, but it has the major advantage of being massively parallelisable and having small memory requirements. This is the reason why, as we will see in Sections 3 and 6, it is still used in practice to finish computations. It can also benefit from previous knowledge on χ mod m as this reduces the size of the searchspace by a factor $m^{g/2}$.

Such information on χ can be gained using "polynomial-time" algorithms such as the ones we describe below. For simplicity, we only present these methods in genus 2 and 3, respectively in Chapters 3 and 6.

p-adic methods

Instead of considering the action of the Frobenius directly on $J_{\mathcal{C}}$, the p-adic approaches are based on computing (a p-adic approximation of) a lift of the Frobenius and its action on some differential forms. There are many algorithms following this philosophy, each using a different lift or different differential forms. For instance, Satoh's algorithm for elliptic curves [124] computes the canonical lift of both the curve and the (dual) Frobenius endomorphism, whose action on the lifted curve determines the trace of the Frobenius. Kedlaya's algorithm [80] just needs a monic lift but it acts on a larger space, namely a Monsky-Washnitzer cohomology group. Compared to Satoh's, this algorithm also has the advantage of working for hyperelliptic curves of arbitrary genera, with a complexity in $\widetilde{O}(pg^4n^3)$ bit-operations and $O(pg^3n^3)$ space. Note that Kedlaya's algorithm does not apply as such in characteristic 2, but this was fixed by Denef and Vercauteren in [41]. This was extended by Tuitman in [142] for (possibly non-hyperelliptic) curves with a "good" lift, where good hides various technical hypotheses that are expected to be satisfied in general.

All these approaches have polynomial complexities in both g and n, and despite an improvement by Harvey [69] reducing the dependency in p to $\sqrt{p}$, this is still exponential in $\log p$, which is why they are used for fields of small characteristic. However, when counting points over many fields, it is remarkable that an average polynomial-time complexity can be reached [71]. Indeed, given a curve over $\mathbb{Q}$, this algorithm computes the zeta function of its reduction modulo p on $\mathbb{F}_p$ for all the primes p of good reduction smaller than N in time $\widetilde{O}(N \log^3 N)$ and polynomial in g. Thus, on average, counting points on each curve amounts to a polynomial complexity in $\log p$, n and g. However, we still do not know any algorithm that has polynomial-time complexity in all these parameters for counting points on a single curve.

Schoof's algorithm and its extensions

input : An elliptic curve $E/\mathbb{F}_q$ given by the equation $y^2 = x^3 + ax + b$
output: $\#E(\mathbb{F}_q)$
$L \leftarrow 0$
$\ell \leftarrow 3$
while $L \leq 4\sqrt{q}$ **do**
  Compute ψ_ℓ
  Let $R = \mathbb{F}_q[X,Y]/\left(\psi_\ell(X), Y^2 - X^3 - aX - b\right)$ `/* this is` $E[\ell]$ `*/`
  In R, compute $F_0 = (X^{q^2}, Y^{q^2}) \oplus_E [q \bmod \ell](X,Y)$ and $F_1 = (X^q, Y^q)$
  Store t_ℓ the unique element of $\mathbb{Z}/\ell\mathbb{Z}$ such that $F_0 = [t_\ell]F_1$
  $L \leftarrow L \cdot \ell$
  $\ell \leftarrow \mathsf{NextPrime}(\ell)$
end
By CRT find t such that $t \equiv t_\ell \bmod \ell$ for all previous ℓ.
return $q - t + 1$

Algorithm 2: Schoof's algorithm

In [127], Schoof describes an algorithm to compute the zeta function of an elliptic curve, which amounts to computing the trace of the associated Frobenius endomorphism. The idea is to consider the action of the Frobenius on the ℓ-torsion subgroup to recover $\chi \bmod \ell$ for sufficiently many ℓ and, using Proposition 1.42, to recover χ by CRT.

Proposition 1.44. *Let $\mathcal{C}$ be a smooth projective curve over $\mathbb{F}_q$ and ℓ coprime to q, then the restriction of π to the ℓ-torsion subgroup $J_{\mathcal{C}}[\ell]$ has $\chi \bmod \ell$ for characteristic polynomial.*

Note that in Algorithm 2, $\oplus_E$ denotes the group law of the elliptic curve, which may lead to a division by zero in the algebra R. To avoid this problem, one can previously factor ψ_ℓ to perform operations in fields. This is a costly solution and we prefer to follow the approach of [40] and let the representations of elements of R evolve during the computations. In the unlikely event of a "forbidden" division, we can split ψ_ℓ as a product of two factors and pursue the computations in the algebras obtained by replacing ψ_ℓ by each of its factors, with no consequence on the complexity since each factor has a smaller degree. In this thesis, we sometimes reuse this method under the name of "D5 strategy". Another important aspect in practice is that we can modify the group law to avoid handling the ordinate, and work only in a univariate algebra. While this does not change the asymptotic complexity, it greatly reduces the running time.

Let us analyze the cost of one iteration of Schoof's algorithm for a fixed ℓ. First, ψ_ℓ can be obtained from the recurrence formulas on the ψ_i's. These formulas show that computing ψ_ℓ amounts to computing 5 ψ_k's with $k \simeq \ell/2$, which yields an overall complexity in $O(\ell^{\log_2 5})$ for computing Cantor's ℓ-division polynomials. The bottleneck is the computation of F_0 which requires $O(\log q)$ operations in R, each of them accounting for a bit-complexity in $\widetilde{O}(\ell^2 \log q)$ since ψ_ℓ has degree $(\ell^2-1)/2$. Likewise, computing F_1 is feasible within $O(\log q)$ operations in R, and recovering t_ℓ can be done by exhaustive search for ℓ additions in R.

Thus, for a fixed ℓ the loop costs $\widetilde{O}(\ell^2 \log q(\ell + \log q))$ bit operations. Using results of analytic number theory such as [140, Cor. 10.1], one sees that we have to repeat the loop about $O(\log q/\log\log q)$ times and the largest ℓ to consider has size $O(\log q)$. This proves that the complexity of Schoof's algorithm is in $\widetilde{O}(\log^5 q)$.

Schoof's algorithm was later improved by restricting to specific primes ℓ for which we can test the characteristic equation of the Frobenius in a proper subgroup of $E[\ell]$. This amounts to replacing ψ_ℓ by a factor of degree $O(\ell)$ in the definition of R, reducing the cost of each operation in R by a factor ℓ and therefore having an overall complexity in $\widetilde{O}(\log^4 q)$. We do not discuss these improvements further and refer to [128] for more information.

Schoof's algorithm relies on theoretical results such as Weil's conjecture and Proposition 1.44 which are still valid even in a much more general setting, and it was extended few years later by Pila [114] who proposed an algorithm to count points on Abelian varieties with time-complexity in $O((\log q)^{\Delta})$, where Δ depends on the dimension g of the input Abelian variety A, and its group law.

Although most of the theoretical background is still valid in this much more general context, to compute the action of π on $A[\ell]$ we need to find an explicit description $A[\ell]$ as a $2g$-dimensional vector space, so that given $e \in A[\ell]$ we can compute $\pi(e)$. This is the most difficult part and it constitutes the bottleneck of many if not all the ℓ-adic point-counting algorithms appearing in this thesis. Pila's approach to the problem is to view $A[\ell]$ as a zero-dimensional algebraic set, after getting a description for the maps $[n]$, with $n \leq \ell$. Applying straightforwardly the Frobenius map to an element would not give a polynomial-time algorithm, but using the description of $A[\ell]$ we can repeatedly square elements and reduce by the defining equations. The complexity result follows by bounding the number of monomials appearing in these equations, and applying various primitives such as ideal membership testing and monomial bases computations.

As in Schoof's algorithm, the complexity is polynomial in $\log q$ but the exponent Δ is actually exponential in the dimension g of A, so that the overall complexity is doubly exponential in g. The dependency in g of the exponent of $\log q$ has later been improved by [75] and [3].

In [75], Huang and Ierardi reduce the dependency in g of the exponent in the case of plane curves to a polynomial in g. This is achieved by using another way of representing $J_{\mathcal{C}}[\ell]$: by considering its elements as divisor classes and by using effective Riemann-Roch algorithms to get a semi-algebraic description with size polynomial in ℓ and exponential in the degree of $\mathcal{C}$. An important obstacle to overcome is the presence of singular points.

In [3], Adleman and Huang extend the result of [75] to Abelian varieties, with a more precise complexity bound in $(\log q)^{O(g^2 \log g)}$ for hyperelliptic curves of genus g. This time, the low-degree representation of the Frobenius is achieved through faster *ad hoc* algorithms on semi-algebraic sets.

1.3 Applications of point-counting

In this section, we review some of the various applications of point-counting. While some of them are based on slight variations of previous algorithms, others only need the output of point-counting algorithms. Some applications involve designing curves with special properties that are closely related to the number of points on the (Jacobian of the) curve. In this context, the so-called CM-method can be used to create curves with a prescribed number of points, instead of applying point-counting algorithms to random curves until we are satisfied with the result. For further information on this approach, we refer to [6].

1.3.1 Cryptographic use

Definition 1.45. *Let $(G, \oplus)$ be a cyclic group of order M and P a generator of G. The discrete logarithm problem (DLP) in the group G is the problem of recovering the integer $n \leq M$ from the element $nP = \underbrace{P \oplus \cdots \oplus P}_{n \text{ times}}$.*

This problem has lead to cryptographic applications taking advantage of the fact that the exponentiation $P \mapsto nP$ is a one-way function as long as the DLP is hard.

In [131], Shoup defined a concept of generic group and proved that in such a group, any algorithm must perform at least $\Omega(\sqrt{M})$ group operations in order to compute a discrete logarithm. There are many models for black box groups in the literature for which similar results were proven, such as [112], but we do not intend to review them all.

However, finding such generic groups in real life is not that easy: for instance if $G = \mathbb{Z}/M\mathbb{Z}$, the DLP can be solved in polynomial time by computing an XGCD. In real-life cryptography, G is either the multiplicative group of a finite field or (the Jacobian of) an elliptic curve. Note that the DLP in finite fields is much easier than in a generic group, as the complexity to solve it range from quasipolynomial to subexponential, depending on its characteristic.

Thus, surprisingly enough, Jacobians of curves of fixed genus are the only known examples of groups in which there is still no classical subexponential algorithm to solve the DLP. Yet, some subexponential algorithms exist when g grows asymptotically as fast as $\log q$ and some attacks like in [58], though still exponential, reduced the hardness of the DLP in genus strictly larger than 2, making genus 1 and 2 optimal in terms of keysize. For a more detailed survey on the subject, we refer to [53].

We emphasize that even if we consider a group G in which the DLP is hard, exponential algorithms may still be succesful in practice, for instance if $\#G$ is small. The following technique due to Pohlig and Hellman in [116] shows that considering G of large size is not sufficient since the difficulty of the DLP is entailed to the largest prime factor of $\#G$.

Let us assume that G has order $N = \prod_{i=1}^{r} p_i^{e_i}$, where the p_i's are distinct primes. Let $P_i = N_i P$ with $N_i = N/p_i^{e_i}$, then the subgroup G_i generated by P_i has order $p_i^{e_i}$, so that we can solve the DLP in G by solving it in each G_i and using the Chinese remainder theorem. Thus, the DLP in G is as hard as the DLP in the "hardest" G_i.

We can now assume that G has a prime-power order $N = p^e$. Given $Q = nP$, we want to find n. Since $n < N$, we decompose n in basis p as $n = \sum_{i=0}^{e-1} n_i p^i$. Multiplying this decomposition by p^{e-1}, we get

$$p^{e-1}n = p^{e-1}n_0 + p^e \sum_{i=1}^{e-1} n_i p^i.$$

Now since $Q = nP$, we have $p^{e-1}Q = np^{e-1}P$ so that $p^{e-1}Q = n_0 p^{e-1}P$. We can now recover n_0 by solving a DLP, but in a group of size p instead of p^e. Once done, we do the same for n_1 and so on by induction. Finally, the DLP in G is broken down into solving e DLPs in groups of order p.

To sum up, if we only focus on finding the smallest groups achieving a fixed security level, then we have to choose (Jacobians of) curves of genus 1 and 2. But then, we must find curves such that $\#J_{\mathcal{C}} = \chi(1)$ is prime (or actually almost prime for other cryptographic reasons). Because of Weil's bounds, we already know that our curves have to be defined over a large field. Although no practical attack against curves over fields of small characteristic has been published, standards seem to prefer curves defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$ with p a large prime, so that ℓ-adic methods are more adapted in this context.

For elliptic curves, Schoof's algorithm and its improvements based on Elkies and Atkin's work [128] are efficient enough to allow choosing random curves, counting points on them and retaining only those with an almost prime order. The same method was used in [62] to create a secure genus-2 curve, as we will detail in Chapter 3, but it involves much heavier computations.

1.3.2 Extensions of the Sato-Tate conjecture

Contrary to cryptographic applications for which only the size of the Jacobian is needed, we can use the fact that most of the point-counting algorithms actually compute the full zeta function of the curve. This has been used to study how the zeta functions (or the characteristic polynomials of the p-Frobenius) of a fixed curve $\mathcal{C}$ over $\mathbb{F}_p$ behaves when p varies. In genus 1, this was predicted by the Sato-Tate conjecture in 1948 and proven by Clozel, Harris, Shepherd-Barron and Taylor in [32, 67, 139].

Theorem 1.46. *Let E be an elliptic curve over $\mathbb{Q}$ and t_p the trace of the p-Frobenius of its reduction modulo a prime of good reduction p. If E does not have complex multiplication, then the normalized traces $t_p/2\sqrt{p}$ are equidistributed with respect to the measure $2\mathrm{d}t/\pi\sqrt{1-t^2}$.*

Note that the distribution of these quantities was also known since Deuring for curves with complex multiplication. A natural question is to ask for generalization of this statement in higher genera, both in the general case and in less likely cases analogous to the CM case in genus 1.

It is conjectured that given a curve $\mathcal{C}$, the normalized Weil polynomials $L_p(t)/\sqrt{p}$ of its reductions modulo primes of good reduction follow a distribution that matches that of the

characteristic polynomials of random matrices of a compact subgroup of $\mathrm{USp}(2g)$, called the Sato-Tate group of $\mathcal{C}$. We refer to [78] and [129] for more information on this subject.

In genus 2 and 3, this was investigated in [49] and [82]. Although the exponential generic approaches such as detailed in [134] are still faster than the p-adic ones for the range of curves and Jacobians studied, average polynomial-time algorithms perfectly fit such investigations.

1.3.3 Algorithmic applications

Shortly after Schoof's algorithm, Lenstra used elliptic curves to tackle the problem of factoring integers with the celebrated elliptic curve method (ECM). In this section, we detail two examples where curves, and more precisely point-counting on curves, are involved in designing deterministic algorithms for number theory or computer algebra. Although point counting is not involved in ECM, the number of rational points of the chosen elliptic curve plays an important role since it has to be smooth.

Primality proving

Given an integer N, we want an algorithm running in time polynomial in N that returns "yes" if N is prime and "no" if not, with a small probability of giving a wrong answer. We present two algorithms in which ℓ-adic methods play a central role, but let us first give an introductory example.

Assume that we can find another integer $m < N$ such that $m-1$ is coprime to N and such that $m^{(N-1)/2} \equiv 1 \bmod N$, then if $(N-1)/2$ is prime, N is prime as well and we repeat the process until the primality of N has been reduced to a number which is known to be prime (for instance any prime smaller than 100). Since it is quite easy to find a good m quickly by taking random integers, this would yield a probabilistic polynomial-time algorithm. But it has a fatal flaw: if $(N-1)/2$ is not prime we cannot draw any conclusion on the primality of N.

To deal with this obstacle, Goldwasser and Kilian [65] reduced the primality of N to that of another integer r which is roughly two times smaller than N but can be different from $(N-1)/2$. This is achieved by considering a random elliptic curve E and computing $m = \#E(\mathbb{Z}/N\mathbb{Z})$ using Schoof's algorithm. Then if m happens to be even, one can prove that the primality of $r = m/2$ entails that of N. Provided that there are sufficiently many "good" integers m occurring as cardinalities of random elliptic curves such that r is actually prime, this method achieves polynomial complexity. Unfortunately, this amounts to proving that there are sufficiently many primes between $N - \sqrt{N}$ and $N + \sqrt{N}$, but current knowledge on the distribution of primes is not even sufficient to even prove that there is a single prime in that interval.

Adleman and Huang found a workaround in [4] by devising two extensions of the previous algorithm, and combining them together. First, instead of only considering the case $m = 2r$, they reduced the primality of N to r such that $m = \lambda r$ with λ a small prime. This yields an algorithm terminating in polynomial time for integers smaller than x outside of a subset of size bounded by $x^{15/16}$. The other extension is to consider genus 2 curves instead of elliptic curves. Indeed, while there is still a polynomial-time analogue of Schoof's algorithm for counting points, the Hasse-Weil interval has a size larger than $N\sqrt{N}$. This "reduces" the primality of N to that of a larger integer, which could be a flaw at first sight. But they actually proved that after repeating this step three times at most, they obtained a candidate prime large enough for the first variant of the Kilian-Goldwasser algorithm to return the correct answer in polynomial time.

Agrawal, Kayal and Saxena later proposed a deterministic polynomial-time algorithm for primality proving. Although these algorithms answer a theoretical question, we also remark that

using elliptic curves for probabilistic primality testing is also competitive in practice thanks to work of Atkin and Morain [7, 109]. Indeed, a recent computation using ECPP gave a primality certificate for $2^{116224} - 15905$ in November 2017 by Peter Kaiser [76].

Deterministic factorisation of polynomials over finite fields

A recent paper by Poonen [117] highlights the potential of ℓ-adic methods to design a polynomial-time deterministic algorithm for factoring polynomials over finite fields. This is based on an idea by Kayal using Schoof's algorithm in the following way. Let us assume that we are given $P \in \mathbb{F}_p[t]$ such that $P = (t - r_1)(t - r_2)$, and we want to recover the two factors of P. By a result of Berlekamp [15], one can reduce the problem of factorization in $\mathbb{F}_q[t]$ to that of factoring polynomials in $\mathbb{F}_q[t]$ with distincts roots all in $\mathbb{F}_p$, and by induction on the degree of P, handling the case $P = (t-r_1)(t-r_2)$ is sufficient to perform factorization of any polynomial over a finite field.

Defining $B = \mathbb{F}_p[t]/P$, we consider an elliptic curve E on B. Actually, E splits as a cartesian product of two elliptic curves over $\mathbb{F}_p$, which we denote E_1 and E_2. Assuming $\#E_1(\mathbb{F}_p) \neq \#E_2(\mathbb{F}_p)$, the respective traces t_1 and t_2 of the Frobenius of E_1 and E_2 are also different. Therefore, there is a prime ℓ such that $t_1 \not\equiv t_2 \bmod \ell$. When applying Schoof's algorithm on E as if B were a field, we end up considering that ℓ and looking for a candidate t_0 for the trace of the Frobenius of E modulo ℓ. Doing so by exhaustive search we encounter special phenomena for $t_0 = t_1$ and $t_0 = t_2$, and eventually recover r_1 and r_2. Indeed, for $t_0 = t_1$, all the elements of the curve E over $\mathbb{F}_p[t]/(t - r_1)$ satisfy the characteristic equations $\phi_p^2 - t_0\phi_p + p = 0$, but not all the elements of E, since $t_1 \neq t_2$. This leads to a division by a non-invertible element in B, itself leading to a proper factor of P.

If $\#E_1(\mathbb{F}_p) = \#E_2(\mathbb{F}_p)$, however, this does not work so we have to choose another E and hope not to fall in the same pathologic case. It is reasonable to think that there is enough room for the choice of E to end up in a good situation after only a few attempts, but this is still unproved. To increase the chances of success, Poonen suggests to switch to higher-dimensional Abelian varieties and use Pila's algorithm instead of Schoof's, as their zeta functions have g degrees of freedom instead of one. Although it is even more convincing, the fact that we have "enough" different zeta functions remains unproved.

Other applications

Schoof's algorithm and its generalization all rely on having a nice representation of the ℓ-torsion, in a sense that we have already mentioned, and will make clearer in Chapters 3 to 5. An example is given in [122, Sec. 7.5] to compute all the ℓ-isogenies from an Abelian variety knowing its ℓ-torsion subgroups.

Last, some multiplication algorithms like [31] or algebro-geometric codes benefit from curves with many rational points [66, 107, 79]. We do not further develop these aspects since they use mostly non-hyperelliptic curves.

Chapter 2

Polynomial systems

The generalizations of Schoof's algorithm all rely on describing the ℓ-torsion in a way that allows to test ideal membership and perform group operations. In genus greater than 1, this step is the bottleneck of these algorithms, and therefore the step to improve in order to get better complexity estimates. The direction that we investigate in this thesis consists in formally multiplying a divisor D by ℓ and then solving the polynomial system obtained after equating $\ell D = 0$. The aim of this section is to define what we mean by polynomial system solving, to present the methods that we use to do so and to study their complexities. These methods and complexity results are used in Chapters 3 to 6.

Since all our systems will be designed to model (subsets of) the ℓ-torsion of Abelian varieties, they will all have dimension zero. Thus, all the definitions and statements of this section are given in the particular case of zero-dimensional systems.

In this chapter, we review three methods for solving polynomial systems along with complexity results that we reuse later. Section 2.2 recalls algorithms for computing Gröbner bases, but their complexities are hard to bound, so that they are only used for practical results in Chapter 6. Section 2.3 deals with resultants that provide a good alternative both in theory and in practice for bivariate systems, as detailed in Chapter 3. In the trivariate case, they are no longer competitive against algorithms like F4 but they can still be used to derive complexity bounds in Chapter 6. Lastly, Section 2.4 is dedicated to the geometric resolution, a method used in Chapter 5 to take advantage of structural properties of our polynomial systems.

2.1 Solving polynomial systems

Definition 2.1. *Let K be a field, and let $f_1, \dots, f_m$ be polynomials in $K[x_1, \dots, x_n]$. The solutions of the polynomial system $\{f_1, \dots, f_m\}$ are the tuples $(z_1, \dots, z_n) \in \bar{K}^n$ such that for all $i \in \{1, \dots, m\}$, $f_i(z_1, \dots, z_n) = 0$. When the set of solutions is finite, we say that the system is zero-dimensional (or has dimension zero). In that case, we refer to the number of solutions (in $\bar{K}$) counted with multiplicities as the degree of the system.*

The simplest possible case of operations in a quotient ring is the univariate case. For instance, given an elliptic curve E the ℓ-division polynomials allow us to reduce the computations of the Frobenius in $E[\ell]$ to exponentiation in the quotient ring $\mathbb{F}_q[X]/\psi_\ell(X)$. In more general cases, we always fall back to the univariate case using one of the following strategies. Either we eliminate variables one by one to end up with one univariate equation, or we parametrize all the variables by another one. More precisely, we say that we have solved a system when we have put it in one of the two following forms.

Definition 2.2 (Triangular form)**.** *We say that a zero-dimensional polynomial system is triangular if it has the form*

$$\begin{aligned} g_1(x_1, x_2, x_3, \ldots, x_n) \\ \vdots \\ g_{i_1}(x_1, x_2, x_3, \ldots, x_n) \\ g_{i_1+1}(x_2, x_3, \ldots, x_n) \\ \vdots \\ g_{i_2}(x_2, x_3, \ldots, x_n) \\ g_{i_2+1}(x_3, \cdots, x_n) \\ \vdots \\ g_{i_n}(x_n) \end{aligned}$$

Actually, we can often get an even simpler form like

$$\begin{aligned} x_1 - h_1(x_n) \\ \vdots \\ x_{n-1} - h_{n-1}(x_n) \\ h_n(x_n). \end{aligned}$$

When a system can be put in this form, we say that the system is in shape position. It has been proven in [14] that when the associated ideal is radical this is very likely after a random linear change of variables, provided that the field of definition is large enough.

Definition 2.3 (Geometric resolution)**.** *A geometric resolution of a zero-dimensional polynomial system is a linear combination x_0 of the variables x_i's and a system of the form*

$$\begin{aligned} h_0(x_0) = 0 \\ x_1 = h_1(x_0) \\ \vdots \\ x_n = h_n(x_0) \end{aligned}$$

where h_0 is a univariate polynomial whose degree D is the degree of the polynomial system, and the h_i are univariate polynomials of degrees smaller than D. The linear combination x_0 is called a separating variable or a primitive element.

To compute a triangular form of our system, a possible strategy is to eliminate one variable and then repeat the same procedure on the equations with $n-1$ variables. We made this precise by introducing the following definition

Definition 2.4. *[38, Sec. 3 Def. 1] Given $I = \langle f_1, \ldots f_m \rangle \subset K[x_1, \ldots, x_n]$, the k-th elimination ideal I_k is defined by*

$$I_k = I \cap K[x_{k+1}, \ldots, x_n].$$

An elimination scheme is an algorithm to compute a generating set of I_k, or at least a set of elements of I_k. A historical example of elimination is the Gauß-Jordan elimination for solving

linear systems of equations. This method can be seen as computing the row-reduced form of a matrix associated to the system, so it is no surprise that we end up with a system in triangular form. In Sections 2.2 and 2.3, we review two ways of performing elimination, respectively by computing a Gröbner basis or resultants.

2.2 Gröbner bases

This section presents properties of Gröbner bases and explains why they are a particularly convenient tool for solving polynomial systems. We briefly present known strategies to compute them and review complexity results.

2.2.1 Gröbner bases and elimination

Definition 2.5. *A monomial x^α in $K[x_1, \ldots, x_n]$ is an element $x_1^{\alpha_1} \ldots x_n^{\alpha_n}$ with $\alpha = (\alpha_1, \ldots, \alpha_n)$ a tuple of nonnegative integers. The total degree of such a monomial is the sum $|\alpha| = \sum_{i=1}^n \alpha_i$.*

Definition 2.6. *A monomial ordering on $K[x_1, \ldots, x_n]$ is a relation $\prec$ on $\mathbb{Z}^n_{\geq 0}$ such that*

- *if $\alpha \prec \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$, then $\alpha + \gamma \prec \beta + \gamma$,*
- *$\prec$ is a well-ordering, i.e. it is a strict total ordering such that every nonempty subset of $\mathbb{Z}^n_{\geq 0}$ has a smallest element under $\prec$.*

This gives an ordering on the set of monomials via $\alpha \mapsto x^\alpha$ which is compatible with the multiplication of monomials. Given an element $P = \sum_\alpha a_\alpha x^\alpha$ of $K[x_1, \ldots, x_n]$, it allows to define:

- the multidegree $\mathrm{mdeg}(P)$, the greatest α (for the monomial order) such that $a_\alpha \neq 0$,
- the leading monomial $\mathrm{LM}(P) = x^{\mathrm{mdeg}(P)}$, the greatest monomial appearing in P,
- the leading coefficient $\mathrm{LC}(P) = a_{\mathrm{mdeg}(P)}$, the coefficient of the leading monomial of P,
- the leading term $\mathrm{LT}(P) = \mathrm{LC}(P)\,\mathrm{LM}(P)$,

If $S \in K[x_1, \ldots, x_n]$ is a set of polynomials, we define $\mathrm{LT}(S) = \{\mathrm{LT}(P) \mid P \in S\}$.
In this thesis, we mostly encounter the following two monomial orderings:

Example 2.7. *(Lexicographic order). Let α and β be two elements in $\mathbb{Z}^n_{\geq 0}$, we write $\alpha \prec_{lex} \beta$ if there exists i such that $\alpha_j = \beta_j$ for any $j < i$ and $\alpha_i < \beta_i$.*

In other words, the monomials are ordered by lexicographic order using the order $x_n \prec x_{n-1} \prec \cdots \prec x_1$ for the variables.

Example 2.8. *(Graded reverse lex order). Let α and β be two elements in $\mathbb{Z}^n_{\geq 0}$, we write $\alpha \prec_{grevlex} \beta$ if $|\alpha| < |\beta|$ or if $|\alpha| = |\beta|$ and there exists i such that $\alpha_j = \beta_j$ for any $j > i$ and $\alpha_i > \beta_i$.*

In other words, grevlex orders first by total degree and then uses the reverse lexicographic order to compare in case of equality, using the order $x_1 \prec x_2 \prec \cdots \prec x_n$ for the variables.

Consider the case of a system of two univariate polynomials, $\{P(X), Q(X)\}$. A triangular form of this system is $\{\gcd(P(X), Q(X))\}$, and the GCD computation can be done using Euclid's algorithm, i.e. successively reducing one polynomial by the other. In what follows, we introduce definitions to extend the notion of reduction to the multivariate case.

Theorem 2.9. *[38, Chap. 2, Th. 3] Let us fix a monomial order and let $F = (f_1, \ldots, f_s)$ be an ordered tuple of polynomials in $K[x_1, \ldots, x_n]$. Every $f \in K[x_1, \ldots, x_n]$ can be written*

$$f = \sum_{i=1}^{s} a_i f_i + g,$$

with the a_i's and g in $K[x_1, \ldots, x_n]$ such that g is either 0 or a linear combination of monomials that are not divisible by any of the $LM(f_i)$'s. Furthermore, if $a_i f_i \neq 0$, then we have $\mathrm{mdeg}(f) \geq \mathrm{mdeg}(a_i f_i)$.

Definition 2.10. *In the setting of the previous theorem, we denote $g = \overline{f}^F$ and call it a remainder of f modulo F. Furthermore, if $\mathrm{LM}(g) \prec \mathrm{LM}(f)$, we say that f is top-reducible.*

These definitions generalize the univariate Euclidean division but in a much weaker sense: even given a fixed monomial ordering, there is no unicity of the remainder in general. For some well-chosen sets F, however, the remainder modulo F is unique and it is thus possible to perform an analogue of Euclid's algorithm. In the next section, we define Gröbner bases that are an example of such nice sets.

Definition 2.11 (Gröbner basis)**.** *Let I be an ideal of $K[x_1, \ldots, x_n]$, $\prec$ a monomial ordering and a finite subset $G \subset I$. Then G is a Gröbner basis of I for the order $\prec$ if $\langle LM(G) \rangle = \langle LM(I) \rangle$.*

Theorem 2.12. *[38, Chap. 2, §5 Cor. 6] Given a monomial ordering $\prec$, any nonzero ideal has a Gröbner basis for $\prec$.*

The previous theorem guarantees the existence of a Gröbner basis but there is no unicity: given G a Gröbner basis, the set G' obtained by adding any element in I is another Gröbner basis. In the following definition, this inconvenience is fixed by adding some minimality condition.

Definition 2.13. *A Gröbner basis G of I is said to be reduced if for all $h \in G$ we have $\mathrm{LC}(h) = 1$ and no monomial of h is in $\langle LM(G \backslash \{h\}) \rangle$.*

Proposition 2.14. *[38, Chap. 2, §7 Prop. 6] Let I be a non-zero ideal of $K[x_1, \ldots, x_n]$ and $\prec$ a monomial ordering. Then I has a unique reduced Gröbner basis G for $\prec$.*

Note that while the reduced Gröbner basis of I for a monomial order is unique, it may differ from the reduced Gröbner basis for a different monomial order. As announced previously, one of their essential features is the unicity of the reduction of a polynomial by a Gröbner basis, as defined in Definition 2.10.

Proposition 2.15. *Let $G = \{g_1, \ldots, g_k\}$ be a Gröbner basis of an ideal I and let $f \in K[x_1, \ldots x_n]$. Then there exists a unique $r \in K[x_1, \ldots x_n]$ such that:*

- *no monomial of r is divisible by any $\mathrm{LT}(g_i)$, i.e. r is in normal form modulo G,*
- *there exists $h \in I$ such that $f = h + r$.*

The unique r is called the normal form of f modulo G, still denoted $r = \overline{f}^G$.

The following proposition gives additional characterizations of Gröbner bases.

Proposition 2.16. *A finite set G is a Gröbner basis of an ideal I if one of the following equivalent properties is satisfied:*

- *for every $f \in I$, at least one of the reductions of f modulo G is zero.*
- *every non-zero $f \in I$ is top-reducible modulo G,*
- *for every $f \in I$, there exists $g \in G$ such that* $\mathrm{LM}(g)$ *divides* $\mathrm{LM}(f)$.

Theorem 2.17 (The elimination theorem)**.** *[38, Sec. 2, Th. 2] Let $I \subset K[x_1, \ldots, x_n]$ be an ideal and let G be a Gröbner basis of I with respect to the lexicographic order where $x_n \prec \ldots \prec x_1$, then for every $j \leq n$ the set*

$$G_j = G \cap K[x_{j+1}, \ldots, x_n]$$

is a Gröbner basis of the j-th elimination ideal $I_j = I \cap K[x_{j+1}, \ldots, x_n]$.

In particular, this shows that a Gröbner basis of an ideal for the lexicographic order is in triangular form.

2.2.2 Computing Gröbner bases

The first algorithm to compute Gröbner bases was introduced by Buchberger in 1965. Like Gaussian elimination, it relies on cancelling the leading monomials of two polynomials by combining them.

Definition 2.18. *Let P and Q be two polynomials in $K[x_1, \ldots, x_n]$, the S-polynomial of P and Q with respect to the monomial ordering $\prec$ is the combination*

$$S(P,Q) = \frac{\mathrm{lcm}(\mathrm{LM}(P), \mathrm{LM}(Q))}{\mathrm{LT}(P)} P - \frac{\mathrm{lcm}(\mathrm{LM}(P), \mathrm{LM}(Q))}{\mathrm{LT}(Q)} Q.$$

Then $\mathrm{LM}(S(P,Q))$ *is strictly smaller (for $\prec$) than* $\mathrm{lcm}(\mathrm{LM}(P), \mathrm{LM}(Q)))$. *We call (P,Q) a critical pair and $S(P,Q)$ the S-polynomial associated to the critical pair.*

This gives another characterization of Gröbner bases.

Proposition 2.19. *Let $G = \{g_1, \ldots, g_k\}$ be a subset of $K[x_1, \ldots, x_n]$ not containing 0 and let $S_{ij} = S(g_i, g_j)$ be the S-polynomials for the monomial ordering $\prec$. Then G is a Gröbner basis of $\langle g_1, \ldots, g_k \rangle$ if and only if for any i, j, at least one of the reductions modulo G of S_{ij} is zero.*

Buchberger's algorithm constructs Gröbner bases by forcing this proposition to be satisfied: starting from a set $F = \{f_1, \ldots, f_m\}$, compute all the S-polynomials, reduce them by F and repeat the operation to the union of F and all the non-zero remainders modulo F.

Note that we still have to worry about the termination of Buchberger's algorithm. Dickson's lemma states that any monomial ideal has a finite basis, which is equivalent to the fact that there is no infinite increasing sequence of monomial ideals. See for instance [38, Chap. 2, Sec. 4 & 5] for statements and proofs.

There is an extensive literature on improvements to Buchberger's algorithm, but we mainly focus on two types of improvement that we used in practice in Chapter 6. The first idea is to compute Gröbner bases for the grevlex order as their computations involve polynomials of smaller degrees. This is noticeable both in practice and in the complexity bounds given in [91]. However, grevlex bases are often not sufficient to directly solve a system, contrary to their lex counterparts. We can circumvent this difficulty by a change of ordering using either the FGLM algorithm [47] in dimension zero, or a Gröbner walk [35] in positive dimension.

Since most of the running time of Buchberger's algorithm is spent computing reductions of critical pairs, the choice of the order in which we reduce them plays a prominent role. There is

```
input  : F = {f_1, ..., f_m} and ≺ a monomial ordering
output: A Gröbner basis G for ≺
G ← F, G' ← ∅
while G ≠ G' do
    G' ← G
    for each critical pair (P, Q) with P, Q in G' and P ≠ Q do
        S ← S(P, Q)^{G'}
        if S ≠ 0 then
            G ← G ∪ {S}
        end
    end
end
return G
```

Algorithm 3: Buchberger's algorithm as in [38, Chap. 2, Th. 2].

no absolute answer to this question, but practical experiments allow to compare the efficiency of different choices. The so-called normal strategy consists of reducing first by pairs of small degrees and seems to be quite efficient. Another improvement was brought by the F4 algorithm [45], using linear algebra to perform reductions much faster. The link between Gröbner bases and linear algebra will be detailed in the next section, as it is also helpful to prove complexity bounds.

Further improvements on the reduction step can be designed, for instance by anticipating and avoiding reductions of some critical pairs to zero. This idea was introduced by Buchberger's criteria (see [38, Sec. 9]) and later improved in the F5 algorithm [46].

2.2.3 Complexity results

Let us consider an ideal I generated by m homogeneous polynomials $f_i \in K[x_1, \ldots, x_n]$ of respective degrees d_i. We introduce $I_d = \{f \in I \mid \deg f = d\}$ and point out that it is a vector space of finite dimension. Since any element of I_d can be decomposed in the basis of degree-d monomials of $K[x_1, \ldots, x_n]$, $\dim I_d \leq \binom{n+d-1}{d}$.

Definition 2.20. *Let $I = \langle f_1, \ldots, f_m \rangle$ be a homogeneous ideal. A finite set G is a d-Gröbner basis of I if it generates I and if any of the following equivalent statements hold:*

- $\forall g_1, g_2 \in G$, $\overline{S(g_1, g_2)}^{G} = 0$ *as long as* $\deg S(g_1, g_2) \leq d$,
- *every $f \in I$ with $\deg(f) \leq d$ is top-reducible by G.*

Note that a d-Gröbner basis is also a k-Gröbner basis for $k \leq d$, so that a sequence G_i of i-Gröbner bases is increasing. Thus, the ascending chain condition implies that G_i is stationnary, i.e. there is a D such that $G_k = G_D$ for any $k \geq D$. Hence, for $k \geq D$, a k-Gröbner basis is a Gröbner basis. An algorithm computing d-Gröbner basis can be derived from Buchberger's algorithm by only considering S-polynomials of degrees $\leq d$.

To make the link between Gröbner bases and Gaussian elimination even clearer, one can represent the vector space I_d by a matrix whose columns are indexed by the degree-d monomials of $K[x_1, \ldots, x_n]$ (in decreasing order for $\prec$) and the rows by the degree-d multiples of the m generators of I. This was introduced by Macaulay in [97] and the matrix is named the degree-d Macaulay matrix of I. It enables to perform operations in I by using linear algebra, and generalizes the notion of Sylvester matrix (see Definition 2.30, below).

Definition 2.21. *A homogeneous polynomial f of degree d is generic if it can be written as*

$$f = \sum_{i_1+\cdots+i_n=d} U_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n},$$

where the $U_{i_1,\ldots,i_n}$ are variables.

Definition 2.22. *Let $f_1, \ldots f_n$ be generic homogeneous polynomials of respective degrees d_i in the variables $x_1, \ldots, x_n$ such that the variables appearing as coefficients of the f_i are all distinct. The Macaulay resultant of the f_i's is the GCD of all the minors of maximal size of the degree-d Macaulay matrix, with $d = \sum_{i=1}^{n}(d_i - 1) + 1$.*

Proposition 2.23. *[97] The Macaulay resultant R is a homogeneous polynomial, irreducible and of degree $D_i = \prod_{j \neq i} d_j$ in the coefficients of f_i. The system $\{f_1, \ldots, f_n\}$ has a non-trivial solution if and only if R vanishes.*

In [91, 92], Lazard performs Gaussian elimination to the degree-d Macaulay matrices for $d \leq D$. This yields a D-Gröbner basis, which is a Gröbner basis for D large enough. From the maximal degree of polynomials appearing in the computations and the size of the Macaulay matrix of that degree, one can deduce a complexity bound for Lazard's algorithm. However, as in Buchberger's algorithm, we expect this algorithm to perform many unnecessary reductions to zero since the matrix has a rank much smaller than its size. Using the F5 criteria, it is possible to consider a much smaller matrix and design a matrix-based counterpart to the F5 algorithm [46].

To construct the degree-d Macaulay matrix in the affine case, the columns are indexed by all the monomials of degree $\leq d$ and the rows by multiples of the f_i's by monomials such that the product has degree $\leq d$. As previously, a d-Gröbner basis is computed by reducing the affine degree-d Macaulay matrix.

A similar method to find a solution of a polynomial system is to solve the linear system $MX = 0$ with M the Macaulay matrix of the system. We do not detail it further and refer to [36] for the introduction of XL and its application to cryptanalysis.

In the worst case, the cost of computing a Gröbner basis is doubly exponential in the number of variables (see [100]), but this bound was reached using tailored systems and is pessimistic even for random inputs. In fact, most systems that we encounter have particularities that make them even simpler to solve. Indeed, we seldom want to solve random polynomial systems but rather focus on examples coming from specific contexts. In our case, the ideals that we consider are zero-dimensional by nature, a special case for which the complexity drops to simply exponential.

In the overdetermined case, the following theorem states that the maximal degree is at least twice smaller than the Macaulay bound.

Theorem 2.24. *[136] Let $f_1, \ldots, f_{n+1}$ be a generic system (i.e. the coefficients of the f_i's are parameters) of respective degrees $d_1, \ldots, d_{n+1}$ in $K[x_1, \ldots, x_n]$. Then the Macaulay resultant of the homogeneized system can be computed from a Macaulay matrix of degree at most $\left(\sum_{i=1}^{n+1}(d_i - 1) + 1\right)/2$.*

The Macaulay bound gives a dependency of the complexity in the arithmetic mean of the degrees of the equations. On the other hand, one can wish for results involving their geometric mean given by the following theorem due to Lakshman and Lazard.

Theorem 2.25. *[89, Th. 1] There exists a probabilistic algorithm which, given a zero-dimensional system, computes a Gröbner basis of its radical over the field of coefficients in time polynomial in the Bézout bound $\prod i = 1^n d_i$.*

The remainder of the section presents a tighter complexity estimate from [11], when further assumptions are made on the system. These assumptions guarantee that all the trivial reductions in Buchberger's algorithm are avoided using the F5 criteria.

Definition 2.26 (Regular sequence)**.** *[11, Def. 1.7.1] Let $f_1, \ldots, f_m$ be a sequence of homogeneous polynomials in $K[x_1, \ldots, x_n]$. We say that the sequence is regular if the following conditions hold*

- $\langle f_1, \ldots, f_m \rangle \neq K[x_1, \ldots, x_n]$
- *for $2 \leq i \leq m$, if $g_i f_i \in \langle f_1, \ldots, f_{i-1} \rangle$, then $g_i \in \langle f_1, \ldots, f_{i-1} \rangle$.*

Definition 2.27. *[11, Def. 1.7.2] Let $f_1, \ldots, f_m$ be a sequence of polynomials in $K[x_1, \ldots, x_n]$. Denote by f_i^h the homogeneous part of highest degree in f_i. We say that the sequence $f_1, \ldots, f_m$ is regular if the sequence of homogeneous polynomials $f_1^h, \ldots, f_m^h$ is regular.*

Definition 2.28 (Noether position)**.** *A homogeneous ideal I of $K[x_1, \ldots, x_n]$ is in Noether position if there exists $r \leq n$ such that $I \cap K[x_1, \ldots, x_r] = (0)$ and $K[x_1, \ldots, x_n]/I$ is an integral extension of $K[x_1, \ldots, x_r]$.*

Let $f_1, \ldots, f_m$ be a sequence of elements in $K[x_1, \ldots, x_n]$ with $m \leq n$ and d_i the degree of f_i. Let us fix the monomial ordering grevlex with $x_n \prec \cdots \prec x_1$ and make the following hypotheses:

(Hyp. 1) the sequence $f_1, \ldots, f_m$ is regular

(Hyp. 2) for any $1 \leq i \leq m$, the ideal $\langle f_1, \ldots, f_i \rangle$ is in Noether position.

Define $g_{d,i}(n)$ as the coefficient of z^d in the expansion of

$$\frac{z^{d_i}}{(1-z)^{i-1}} \prod_{k=1}^{i-1} (1 - z^{d_k}).$$

The $g_{d,i}(n)$ bound the number of degree-d polynomials in the Gröbner basis of $\langle f_1, \ldots, f_i \rangle$ for the grevlex ordering [11, Th. 3.4.1] . The previous expression is in fact a polynomial whose degree equals the Macaulay bound $\sum_{j=1}^{i} (d_j - 1) + 1$.

Theorem 2.29. *[11, Th. 3.4.2] Under Hypotheses 1 and 2, there exists an algorithm to compute the Gröbner basis of $\langle f_1, \ldots, f_m \rangle$ which performs a total number of elementary operations bounded by*

$$\sum_{i=1}^{m-1} \sum_{d=0}^{\infty} g_{d+d_{i+1},i+1}(n) \binom{i+d+d_{i+1}}{d+d_{i+1}} \binom{n+d+d_{i+1}-1}{d+d_{i+1}}.$$

This result is achieved using a variant of the F5 algorithm, and although it is not easily compared to other complexity bounds, it is instantiated in particular cases in [11, 12] in which a simpler complexity bound is derived and indeed yields an improvement over that of Lazard's algorithm. As explained in [11], these bounds where notably used for solving equations over $\mathbb{F}_2$ and in particular attacking the public-key system HFE, as well as decoding codes. In our setting however, the regularity hypotheses fail and the complexity bound is not tight enough, just as the Bézout bound.

Although Gröbner bases are a powerful tool for polynomial system solving, we cannot use them to derive asymptotic complexity estimates because only the most general and pessimistic

complexity bounds apply to our setting. In particular, the Bézout bound is much too large for our purpose as it does not take into account the fact that most variables come with a small degree, contrary to its multihomogeneous counterpart. In practice however, we will see in Chapter 6 that they are particularly efficient to solve trivariate systems. The next two sections review alternative options for which we have tighter complexity estimates, and Section 2.4 presents a method to achieve a polynomial-time complexity in the multihomogeneous Bézout bound, which is the cornerstone of Chapter 5.

2.3 Resultant-based approaches

2.3.1 Resultants and elimination

Definition 2.30. *Let m and n be two positive integers, and $P = a_0X^m + \cdots + a_m$ and $Q = b_0X^n + \cdots + b_n$ be two polynomials in $K[X]$. We define* $\mathrm{Syl}(P,Q)$ *the Sylvester matrix of P and Q as*

$$\begin{pmatrix} a_0 & a_1 & \cdots & a_m & & & \\ & a_0 & a_1 & \cdots & a_m & & \\ & & \ddots & \cdots & & \ddots & \\ & & & a_0 & a_1 & \cdots & a_m \\ b_0 & b_1 & \cdots & b_n & & & \\ & b_0 & b_1 & \cdots & b_n & & \\ & & \ddots & \cdots & & \ddots & \\ & & & b_0 & b_1 & \cdots & b_n \end{pmatrix}.$$

Definition 2.31. *The resultant of P and Q, denoted* $\mathrm{Res}(P,Q)$ *or* $\mathrm{Res}_X(P,Q)$ *is the determinant of the Sylvester matrix defined above.*

There is a strong link between resultants and GCD, as the last non-zero row of the row-echelon form of $\mathrm{Syl}(P,Q)$ contains the coefficients of a GCD of P and Q. In particular, the degree of $\mathrm{GCD}(P,Q)$ is the corank of $\mathrm{Syl}(P,Q)$ and we have the following result:

Proposition 2.32. *The polynomials P and Q are coprime if and only if* $\mathrm{Res}(P,Q) \neq 0$.

There is an extensive literature on the numerous properties of the resultant, but we only detail those we will reuse. For more information on the subject, we refer for instance to [5].

Let us now remark that it is possible to define resultants even when the coefficient ring is not a field, and in particular when P and Q live in $R[X]$ with $R = K[Y]$. In that case, $\mathrm{Res}_X(P,Q)$ is a polynomial in Y but no longer in X and we expect that the previous proposition still holds, i.e. that the solutions of $P(X,Y) = Q(X,Y) = 0$ satisfy $\mathrm{Res}_X(P,Q)(Y) = 0$. If so, by adding either the equation $P(X,Y) = 0$ or $Q(X,Y) = 0$ we have put the system in triangular form. However, the following proposition shows that those two systems are not equivalent since we may have added additional solutions.

Proposition 2.33 (Prop 6.4, [21])**.** *Let m and n be two positive integers, $P = a_0X^m + \cdots + a_m$ and $Q = b_0X^n + \cdots + b_n$ be two polynomials in $(K[Y])[X]$, with K an algebraically closed field. Then the roots of* $\mathrm{Res}_X(P,Q) \in K[Y]$ *are the y-coordinates of the solutions of the system $P = Q = 0$ and the common roots of the leading coefficients $a_0(Y)$ and $b_0(Y)$.*

In our bivariate example, we eliminated a variable using a resultant and then carried on one of the initial equations to have a triangular system. In many cases and in particular in

the one encountered in Chapter 3, the additional equation can be "nicer", i.e. of the form $S_1(Y)X - S_0(Y)$. Such an equation is given by one of the subresultants defined below.

Definition 2.34 (Subresultant matrix [141]). *Let m, n and j be three positive integers, and $P = a_0X^m + \cdots + a_m$ and $Q = b_0X^n + \cdots + b_n$ be two polynomials in $K[X]$. We define the j-th subresultant matrix of P and Q as the $(n+m-2j) \times (n+m-j)$ submatrix $N^{(j)}(P,Q)$ of the Sylvester matrix by taking the top $m-j$ rows of coefficients of P and the top $n-j$ rows of coefficients of Q.*

$$\left.\begin{pmatrix} a_0 & a_1 & \cdots & a_n & & \\ & a_0 & a_1 & \cdots & a_n & \\ & & \ddots & \ddots & \ddots & \\ & & & a_0 & \cdots & a_n \\ b_0 & b_1 & \cdots & b_m & & \\ & b_0 & b_1 & \cdots & b_m & \\ & & \ddots & \ddots & \ddots & \\ & & & b_0 & \cdots & b_m \end{pmatrix}\right\} \begin{matrix} m-j \\ \\ \\ \\ n-j \\ \\ \\ \end{matrix} .$$

Definition 2.35 (Subresultant [141]). *Keeping the notation of the previous definition, for $k \leq j$, we further define $N_k^{(j)}(P,Q)$ as the (square) submatrix of $N^{(j)}(P,Q)$ obtained by taking only its rightmost $m+n-2k-1$ columns and its $(m+n-j-k)$-th column. The j-th subresultant of P and Q is then defined as the polynomial*

$$\sum_{k=0}^{j} \det N_k^{(j)} X^k.$$

Considering P and Q in $K[X,Y]$, one can define the bivariate resultants and subresultants except that the coefficients a_i and b_j are now polynomials in (say) Y. Thus, the bivariate resultant $R(Y) = \mathrm{Res}_X(P,Q)$ is now a univariate polynomial and the j-th subresultant is a bivariate polynomial of degree at most j in X. If the first subresultant of P and Q is non-zero, then it has degree 1 in X so that we can write it $S_1(Y)X + S_0(Y)$ with S_1 a non-zero univariate polynomial. Since the resultant and subresultants of P and Q are all in the ideal generated by P and Q, the system $P(X,Y) = Q(X,Y) = 0$ is equivalent to the system $\{R(Y) = 0, S_1(Y)X + S_0(Y) = 0\}$.

Let us consider P and Q in $K[x_1, \ldots, x_n]$, which we view as $R[X] = K[x_2, \ldots, x_n][X]$. We can likewise define the Sylvester matrix and resultant $\mathrm{Res}_{x_1}(P,Q) \in K[x_2, \ldots, x_n]$. As in the bivariate case, we eliminate the variable x_1 but the resultant is not necessarily a generator of the first elimination ideal. However, we will see in Chapter 6 that successive elimination by resultants is still accurate enough for us to use it in the trivariate case with an asymptotic complexity that matches that of more sophisticated methods. In the remainder of the section, we give more details about the complexity of computing resultants of polynomials in up to three variables.

2.3.2 Computing univariate resultants

Consider P and Q two univariate polynomials over K. To compute $\mathrm{Res}(P,Q)$, an algorithm that comes in mind would be computing the determinant of the Sylvester matrix. This can be

done in $O(n^\omega)$, where n is a bound on the degrees of P and Q and $\omega < 2.38$ the exponent of linear algebra. Using the fact that $\mathrm{Res}(P,Q) = (-1)^{mn} b_0^{m-r} \mathrm{Res}(Q,R)$ with R the remainder of the Euclidean division of P by Q, one can design a algorithm that returns $\mathrm{Res}(P,Q)$ in time quadratic in n. The subresultants can similarly be related to (variations of) Euclid's algorithm by the fundamental theorem of subresultants (see for instance [141, Th. 3.4.]).

In general, by following Euclid's algorithm, we have a sequence of polynomials whose degrees decrease by one at each step so that n steps are needed and the complexity is indeed quadratic. However, one can imitate the half-GCD algorithm to halve the degree at each step. This yields a quasi-optimal algorithm for computing the resultant and the last non-zero subresultant of two univariate polynomials P and Q.

Properly presenting a fast algorithm for computing the resultant of two polynomials along with their last non-zero subresultant is not a challenge that we want to take in this thesis, insomuch as we did not contribute on this aspect. We therefore limit ourselves to stating the following theorem, which is the only statement about (sub)resultants that will be needed throughout this thesis.

Theorem 2.36 (Computing resultants and subresultants)**.** *[21, Prop. 6.15 & Thm. 6.16] Let P and Q be two univariate polynomials in $K[X]$ of degrees bounded by $n > 0$. Then $\mathrm{Res}(P,Q)$ can be computed in time and space $\widetilde{O}(n)$, and so can any subresultant of P and Q.*

For our purpose, we will also need to compute bivariate and trivariate (sub)resultants, for which the existence of a quasi-optimal algorithm is still an open problem. In the next section, we give complexity bounds for computing these resultants by using evaluation / interpolation schemes to reduce to the univariate case.

2.3.3 Bivariate and trivariate resultants

In Chapters 3 and 6, we put the equations of the ℓ-torsion ideal in triangular form by successively eliminating variables using resultants. In this section, we bound the complexity of computing bivariate and trivariate resultants. We also provide bounds on the degrees of the resultants, either because they intervene in another complexity result or for the following reason: we have seen that the resultant of two polynomials belongs to the elimination ideal but there is no guarantee that it is a generator, and it can even be zero. This gives rise to extraneous solutions of our system that we are not interested in and that we call parasites. When the resultants are not zero, bounding their degrees is a way of controlling the number of parasites. In our case, we can use this to ensure that parasites do not harm the asymptotic complexity. On the other hand, they are not innocuous in practice and part of Chapter 3 is dedicated to reducing their number.

Definition 2.37 (Evaluation-Interpolation)**.** *Given n distinct elements $a_0, \ldots, a_{n-1}$ in a field K, and $P \in K[X]$ a polynomial of degree $< n$, we call (multipoint) evaluation the computation of $P(a_0), \ldots, P(a_{n-1})$.*

Conversely, given b_0, …, b_{n-1}, n additional elements of K we call interpolation the computation of a polynomial P of degree $< n$ such that $P(a_0) = b_0, \ldots, P(a_{n-1}) = b_{n-1}$.

Theorem 2.38. *[21, Th. 5.1] Given n distinct field elements $a_0, \ldots, a_{n-1}$, one can perform the multipoint evaluation or the interpolation in $\widetilde{O}(n)$ field operations.*

Note that when $K = \mathbb{F}_q$, we may not have enough distinct points to perform evaluation or interpolation of a polynomial of large degree. However, when it is the case, we can take a field

extension $\mathbb{F}_{q^\delta}$ of $\mathbb{F}_q$, and that will add a factor $\widetilde{O}(\delta)$ to the complexity. The complexity of the algorithms will be polynomial in the number of evaluation points, therefore, the final complexity will be logarithmic in δ, so that the cost of taking a field extension will be hidden in the $\widetilde{O}()$ notation. We will therefore not mention this potential complication further.

Another difficulty is that an evaluation / interpolation strategy assumes that the points of evaluation are generic enough, so that all the degrees after evaluation are generic. This is again guaranteed by taking a large enough base field. Still, the algorithm remains a Monte-Carlo one. In Chapters 3 and 6, the final results of our algorithms can readily be tested, which is why they are Las Vegas even though they involve resultants-based primitives that are not.

Proposition 2.39. *[54, Thm. 6.22 and Cor. 11.21] Let $P(x, y)$ and $Q(x, y)$ be two polynomials whose degrees in x and y are bounded by d_x and d_y respectively. Then, $R(y) = \mathrm{Res}_x(P, Q)$ can be computed in $\widetilde{O}(d_x^2 d_y)$ field operations, and the degree of R is bounded by $2d_x d_y$.*

Proposition 2.40. *Let $P(x, y, z)$ and $Q(x, y, z)$ be two polynomials whose degrees in each variable are bounded by d. Then, $R(y, z) = \mathrm{Res}_x(P, Q)$ can be computed in $\widetilde{O}(d^5)$ field operations, and the degree of R in each variable is bounded by $2d^2$.*

Proof. The Sylvester matrix has at most $2d$ columns and its entries are bivariate polynomials whose degrees in y and z are bounded by d. Thus, its determinant is a polynomial whose degrees in y and z are bounded by $2d^2$.

We first perform a Kronecker substitution by considering $\tilde{P}(x, y) = P(x, y, y^{2d^2+1})$ and $\tilde{Q}(x, y) = Q(x, y, y^{2d^2+1})$, which are polynomials of degrees $\leq d$ in x and $\leq 2d^3 + d$ in y. Note that the choice to replace z by y^{2d^2+1} is made to be able to invert the Kronecker substitution after the resultant computation.

Next, we compute $\tilde{R}(y) = \mathrm{Res}_x(\tilde{P}(x, y), \tilde{Q}(x, y))$. By Lemma 2.39, it is a univariate polynomial of degree at most $4d^4 + 2d^2$ and can be computed in $\widetilde{O}(d^5)$ operations. We can then invert the Kronecker substitution to get $R(y, z)$, which can be done in time linear in the number of monomials, that is in $O(d^4)$. □

Proposition 2.39 has remained unimproved for several decades, however Villard has recently announced [143] that given two generic bivariate polynomials P and Q in $K[x, y]$, the bivariate resultant $\mathrm{Res}_x(P, Q)$ can be computed in $O\left((d_x^{(2-1/\omega)} d_y)^{1+o(1)}\right)$ field operations. Since the computations of resultants are the bottleneck of the algorithms presented in Chapters 3 and 6, this new algorithm may have a direct impact on their complexity bounds. We discuss this in the dedicated chapters and sum up the impact of these new bounds in the conclusion.

2.4 Geometric resolution

2.4.1 Bézout bound and multihomogeneity

We expect the complexity of solving polynomial systems to depend on n the number of variables, m the number of equations, their respective degrees and possibly also their number of solutions. The number of solutions greatly depends on the system itself and cannot be predicted. However, it is possible to bound it by the previous data, which is the point of the following definition. This bound was introduced by Bézout to study the number of points of intersection between two curves, but it can be generalized as follows:

Theorem 2.41 (Bézout's theorem)**.** *Let $f_1, \ldots, f_n$ be n homogeneous polynomials in $n+1$ variables of respective degrees $d_1, \ldots, d_n$. Then either the number of projective solutions counted with multiplicities (in the algebraic closure) is infinite or equal to the product $d_1 \cdots d_n$.*

We will see later that this bound plays a role in complexity results, but we anticipate that it will not satisfy our needs and we therefore introduce a sharper bound for more structured systems.

Definition 2.42 (Multihomogeneous system)**.** *A multihomogeneous polynomial f is a polynomial such that there exists a partition of the variables in subsets on which the polynomial is homogeneous. If d_i is the degree of the homogeneous polynomial with respect to the i-th subset of variables, the sequence (d_i) is called the multi-degree of f.*

For example, on $K[x_1, \ldots, x_{n_x}, y_1, \ldots, y_{n_y}]$ a bihomogeneous polynomial f of bidegree d_1, d_2 is such that

$$\forall \lambda, \mu \in K,\ f(\lambda x_1, \ldots, \lambda x_{n_x}, \mu y_1, \ldots, \mu y_{n_y}) = \lambda^{d_1} \mu^{d_2} f(x_1, \ldots, x_{n_x}, y_1, \ldots, y_{n_y}).$$

Definition 2.43 (Multi-degree)**.** *We extend this notion to non-homogeneous polynomials by defining the multi-degree of a polynomial f with respect to a partition of the variables as the tuple (d_i) where d_i is the degree of f in the variables of the i-th block, when all the other variables are evaluated at a generic value.*

Definition 2.44. *[110] Let $F = \{f_1, \ldots, f_m\}$ be a non-homogeneous system on $K[X_1, \ldots, X_n]$, where each $X_i = (x_{i,1}, \ldots, x_{i,n_i})$ is a tuple of variables and let $d_{j,1}, \ldots, d_{j,n}$ be the multi-degree of f_j with respect to the X_i's. The multihomogeneous Bézout number of F is defined as the coefficient of $\prod_{i=1}^{n} T_i^{n_i}$ in the product*

$$\prod_{j=1}^{m} \sum_{i=1}^{n} d_{j,i} T_i.$$

Theorem 2.45. *[110] Consider F as above, then it has no more isolated solutions than its multihomogeneous Bézout number d.*

This bound is much more convenient than the original Bézout bound when dealing with a system which has a large number of variables appearing with small degree and a small number of variables appearing with large degree. In Chapter 5 we encounter systems of $O(g^2)$ variables with only g variables of "large" degree δ. In this context, the Bézout bound is in $\delta^{O(g^2)}$ versus $\delta^{O(g)}$ for its multihomogeneous counterpart. The reason why these bounds appear in the complexity is detailed later on: we will see in Section 2.4 that the cost of computing a geometric resolution is polynomial in the maximum of the degrees of intermediate ideals, as defined below. However, contrary to the Bézout bound which is readily computable from the input system, computing the degree of an ideal is not straightforward. When the input system is generic enough (i.e. when it is a regular sequence as in Definition 2.26), the degrees of the intermediate ideals can be bounded by the (multihomogeneous) Bézout bound.

Definition 2.46 (Degree of an ideal)**.** *By identifying a point $(\lambda_0, \ldots, \lambda_n) \in \overline{K}^{n+1}$ with the polynomial $\lambda_0 + \lambda_1 X_1 + \cdots + \lambda_n X_n \in \overline{K}[X_1, \ldots, X_n]$, there is a dense Zariski open subset $\mathcal{O} \subset (\overline{K}^{n+1})^{\dim V(I)}$ such that for any $(\ell_1, \ldots, \ell_{\dim V(I)}) \in \mathcal{O}$, the algebra $\overline{K}[X_1, \ldots, X_n]/(I + \langle \ell_1, \ldots, \ell_{\dim V(I)} \rangle)$ is a finite dimensional $\overline{K}$-vector space of constant dimension, which is called the degree of I.*

Definition 2.47 (Reduced sequence)**.** *The sequence $(f_1, \ldots, f_i)$ is reduced if every intermediate ideal $\langle f_1, \ldots, f_j \rangle$ with $j \in [1, i]$ is radical.*

Proposition 2.48. *Let $f_1, \ldots, f_m$ be a regular sequence in $\mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]$ and $d_x, d_y \in \mathbb{Z}_{\geq 0}$ be such that for any $i \in [1, m]$, $\deg_x(f_i) \leq d_x$ and $\deg_y(f_i) \leq d_y$. Then the degree of the ideal $\langle f_1, \ldots, f_m \rangle$ is at most*

$$\sum_{\substack{j_1+j_2=m \\ 0\leq j_1\leq n_x \\ 0\leq j_2\leq n_y}} \binom{m}{j_1} d_x^{j_1} d_y^{j_2}. \tag{2.1}$$

Moreover, this degree is bounded above by $2^{n_x+n_y} d_x^{n_x} d_y^{n_y}$.

Proof. This is a direct consequence of [123, Prop. I.1] using, with the notation of [123, Prop. I.1], $k = 1$, $e = 0$, $P = m$, $D_{i,0} = d_x$, $D_{i,1} = d_y$, $n = n_x$, $n_1 = n_y$. Note that [123, Prop. I.1] is stated when the base field is $\mathbb{C}$, but the proof works without any major modification when the base field is a finite field. The last sentence of the statement follows from the fact that the regularity assumption implies that $m \leq n_x + n_y$, and hence the sum of the binomial coefficients is bounded above by $2^m \leq 2^{n_x+n_y}$. □

2.4.2 Geometric resolutions

Contrary to the previous two sections, we no longer work with multivariate polynomials in dense representation but as programs describing which operations to perform to evaluate them. We call that a straight line program (SLP) and give a more precise definition of particular instances of SLP.

Definition 2.49 (Division-free SLP)**.** *A division-free SLP (DFSLP) defined over a field K is a sequence of polynomials $h_1, h_2, \ldots, h_\ell \in K[X_1, \ldots, X_n]$ such that each polynomial h_i is either a variable X_t with $t \in [1, n]$, an element in K, or $h_i = h_j \circ h_{j'}$, where $j, j' < i$ and $\circ \in \{+, -, \times\}$ is an arithmetic operation. The time of a DFSLP is the total number of arithmetic operations, and its space is the minimal number of arithmetic registers required to evaluate it. A polynomial system $f_1, \ldots, f_m$ is said to be represented by a DFSLP $h_1, \ldots, h_\ell$ if $\{f_1, \ldots, f_m\} \subset \{h_1, \ldots, h_\ell\}$.*

The following lemma gives a bound on the size of a DFSLP needed to represent a bihomogeneous polynomial:

Lemma 2.50. *Let $d_x, d_y \in \mathbb{Z}_{>0}$ be two positive integers. A polynomial system $f_1, \ldots, f_m \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]$ such that for all $i \in [1, m]$, $\deg_x(f_i) \leq d_x$ and $\deg_y(f_i) \leq d_y$ can be represented by a DFSLP with time and space $O\left((d_x + d_y + m)\binom{n_x+d_x}{n_x}\binom{n_y+d_y}{n_y}\right)$.*

Proof. There are $\binom{n_x+d_x}{n_x}\binom{n_y+d_y}{n_y}$ monomials μ in $\mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]$ such that $\deg_x(\mu) \leq d_x$ and $\deg_y(\mu) \leq d_y$. We consider the DFSLP which starts by evaluating these monomials. This costs less than $\binom{n_x+d_x}{n_x}\binom{n_y+d_y}{n_y}(d_x + d_y - 1)$ multiplications, using a naive algorithm. Then we multiply each of these monomials by the corresponding coefficients, and we sum. This costs $m\binom{n_x+d_x}{n_x}\binom{n_y+d_y}{n_y}$ multiplications and $m(\binom{n_x+d_x}{n_x}\binom{n_y+d_y}{n_y} - 1)$ additions. □

For describing 0-dimensional (i.e. finite) sets $V \subset \overline{\mathbb{F}_q}^n$ where V is defined over $\mathbb{F}_q$, we use a data structure called a geometric resolution of V. The terminology here is borrowed from [25],

see also [64]. The following definition is slightly simpler than the one in [25, Sec. 2.1] because we restrict ourselves to the 0-dimensional case in the whole thesis (in [25, Sec. 2.1], the definition is also valid for equidimensional varieties with positive dimension).

Definition 2.51 (Geometric resolution)**.** *We say that an $\mathbb{F}_{q^e}$-geometric resolution of V is a tuple $((\ell_1, \ldots, \ell_n), Q, (Q_1, \ldots, Q_n))$ where:*

- *The vector $(\ell_1, \ldots, \ell_n) \in \mathbb{F}_{q^e}^n$ is such that the linear form*

$$\begin{array}{rccc} \ell : & \overline{\mathbb{F}_q}^n & \to & \overline{\mathbb{F}_q} \\ & (x_1, \ldots, x_n) & \mapsto & \sum_{i=1}^n \ell_i x_i \end{array}$$

  takes distinct values at all points in V. The linear form ℓ is called the primitive element of the geometric resolution;

- *The polynomial $Q \in \mathbb{F}_{q^e}[T]$ equals $\prod_{\mathbf{x} \in V}(T - \ell(\mathbf{x}))$;*

- *The polynomials $Q_1, \ldots, Q_n \in \mathbb{F}_{q^e}[T]$ parametrize V by the roots of the polynomial Q, i.e.*

$$V = \{(Q_1(t), \ldots, Q_n(t)) \mid t \in \overline{\mathbb{F}_q}, Q(t) = 0\}.$$

2.4.3 Computing geometric resolutions

Following [64], we present the main aspects of the computation of a geometric resolution. Let us consider $\{f_1, \ldots, f_n\}$ a system of homogeneous polynomials in $K[X_0, \ldots, X_n]$ such that the f_i's form a reduced regular sequence, along with an inequation $g \neq 0$. The general idea is to take the equations into account one by one, deducing a geometric resolution of the ideal $I_{i+1} = \langle f_1, \ldots, f_{i+1} \rangle$ from a geometric resolution of $I_i = \langle f_1, \ldots, f_i \rangle$.

Let us assume that we already have a geometric resolution for I_i, that is a description of the system $S_i = \{x_1, \ldots, x_{n-i}, f_1, \ldots, f_i\}$ in the form

$$q(T) = 0, \quad \begin{cases} x_{n-i+1} = T \\ x_{n-i+2} = v_{n-i+2}(T), \\ \quad \vdots \\ x_n = v_n(T) \end{cases}$$

The lifting step consists of computing a description of the system

$$x_1 = \cdots = x_{n-i-1} = f_1 = \cdots = f_i = 0, \quad g \neq 0,$$

in the form

$$Q(x_{n-i}, T) = 0, \quad \begin{cases} x_{n-i+1} = T \\ x_{n-i+2} = W_{n-i+2}(x_{n-i}, T), \\ \quad \vdots \\ x_n = W_n(x_{n-i}, T) \end{cases}$$

This step can be seen as seeing the variable x_{n-i} as a parameter of the geometric resolution, and a solution of S_i can be seen as an approximated solution of the above system at precision $O(x_{n-i})$. By the Newton method, this solution can be lifted at precision $O(x_{n-i}^2)$, and the

process can be repeated until the precision is sufficient to have an exact resolution. This is achieved when the precision becomes greater than the degree of the variety.

At the end of the lifting step, the equation $f_{i+1} = 0$ is still not taken into account. This is the point of the so-called intersection step and it is achieved as follows. First, introduce a new variable X and perform the change of variable in $K[[t]]$

$$x_{n-i} = X - tx_{n-i+1} + O(t^2),$$

in the previous system. This yields

$$Q_t(X,T) = 0, \quad \begin{cases} x_{n-i+1} = T \\ x_{n-i+2} = V_{t,n-i+2}(X,T), \\ \qquad \vdots \\ x_n = V_{t,n}(X,T) \end{cases}$$

With Q_t a polynomial in X and T and the $V_{t,j}$'s are polynomials in T and rational fractions in X with coefficients in $K[[t]]$ at precision $O(t^2)$. Let us now compute

$$A(X) = \mathrm{Res}_T(Q_t(X,T), f_{i+1}(0,\ldots,0,X-tT,T,V_{t,n-i+2}(X,T),\ldots,V_{t,n}(X,T))).$$

This resultant is in $K[X][[t]]$ and substituting $X = x_{n-i} + tx_{n-i+1}$ in $A(X) = a_0(X) + ta_1(X) + O(t^2)$, we get

$$a_0(x_{n-1}) = 0, \quad a_0'(x_{n-i})x_{n-i+1} + a_1(x_{n-i}) = 0.$$

Therefore, we have the following geometric resolution for $S_i \cup \{f_{i+1} = 0\}$:

$$a_0(T) = 0, \quad \begin{cases} x_{n-i} = T, \\ x_{n-i+1} = -\dfrac{a_1(T)}{a_0'(T)} = V_{n-i+1}(T), \\ \qquad \vdots \\ x_n = W_n\left(-\dfrac{a_1(T)}{a_0'(T)}, T\right) = V_n(T). \end{cases}$$

This is not completely satisfying as we must still remove the potential solutions contained in the hypersurface $g \neq 0$. This is the cleaning step and consists essentially of replacing $a_0(T)$ by $a_0(T)/c(T)$, where

$$c(T) = \mathrm{GCD}_T(a_0, g(0,\ldots,0,T,v_{n-i+1},\ldots,v_n)).$$

2.4.4 Complexity bounds

Theorem 2.52. *[64, Th. 1] Let K be a field of characteristic 0 and let $f_1,\ldots,f_n,g$ be polynomials in $K[x_1,\ldots,x_n]$ of degree bounded by d and given in SLP representation of size at most L. Assume furthermore that the f_i's define a reduced regular sequence in the open subset $\{g \neq 0\}$. The geometric resolution of the variety $V(\langle f_1,\ldots,f_n\rangle) \setminus V(\langle g\rangle)$ can be computed with $O\left(n(nL+n^\omega)\mathbf{M}(d\delta)^2\right)$ field operations, where $\omega < 2.38$ is the exponent of linear algebra, $\delta = \max_{i=1,\ldots,n} \deg(\langle f_1,\ldots,f_i\rangle)$ and $\mathbf{M}(N) = O(N\log^2 N \log\log N)$.*

For our purposes, the main result to remember is that one can compute a geometric resolution in time polynomial (actually quadratic) in the (multi-homogeneous) Bézout bound. Note that this result does not apply to our setting, but the following theorem gives a similar statement for finite fields of sufficiently large size.

Theorem 2.53. *[25, Thm. 4.8] Let $f_1, \ldots, f_n \in \mathbb{F}_{q^e}[x_1, \ldots, x_n]$ be a reduced regular sequence, where the polynomials are represented by a DFSLP with space $\mathcal{S}'$ and time $\mathcal{T}'$. Set the following notation:*

- *The integer d is $\max_{i\in[1,n]}(\deg f_i)$;*
- *For any real number $x \geq \exp(1)$, $\mathcal{U}(x) = x(\log x)^2 \log\log x$;*
- *Let $\delta \in \mathbb{Z}_{\geq 0}$ be an integer larger than the degrees of the ideals $\langle f_1\rangle, \langle f_1, f_2\rangle, \ldots, \langle f_1, \ldots, f_n\rangle$.*

Assume further that $q^e \geq 60\, n^4 d\delta^4$. There is a probabilistic Turing machine using space $O((\mathcal{S}' + n + d)\delta^2 \log(q^e\delta))$ and time $O((n\mathcal{T}' + n^5)\mathcal{U}(\delta)(\mathcal{U}(d\delta) + \log(q^e\delta))\mathcal{U}(\log(q^e\delta)))$ which takes such polynomial systems as input and which outputs an $\mathbb{F}_{q^e}$-geometric resolution of the algebraic set $\{\mathbf{x} \in \overline{\mathbb{F}_{q^e}}^n \mid f_1^{(M)}(\mathbf{x}) = \cdots = f_n^{(M)}(\mathbf{x}) = 0\}$ with probability at least $11/12$.

The above complexity estimates derive from two costly steps: the lifting and the intersection. The former's complexity is essentially due to the cost of computing a Newton lift at precision δ and the latter's bottleneck is the computation of the resultant A. The regularity assumption on the input system ensures that we have an invertible Jacobian matrix to perform the Newton iterations. We do not investigate further and refer the interested reader to [64, 25] for more details.

Chapter 3

Counting points on genus-2 curves

In this chapter, we investigate genus-2 extensions of Schoof's algorithm both in theory and practice, along with their applications in cryptography. Like elliptic curves, Jacobians of hyperelliptic curves are ideal candidates for cryptographic groups. However, some attacks were designed for Jacobians of curves of genus ≥ 3 and while these attacks remain exponential, they imply a less advantageous ratio between key-length and security level. For genus-2 curves, this ratio is comparable to elliptic curves and by using the Kummer surface associated to the curve rather than its Jacobian, a genus-2 Diffie-Hellman protocol detailed in [119] can be made faster than its elliptic analogue [105, 17] thanks to more efficient arithmetic operations designed in [55]. More recently, a signature scheme based on Kummer surfaces of genus-2 curves was designed in [120]. Almost all the results presented here were already known before the beginning of this thesis, so this section can be considered as a warm-up for Chapters 5 and 6 as we focus on parts of the algorithm that we extend later to hyperelliptic curves of larger genera.

Although Pila's algorithm [115] already yields a polynomial-time algorithm for counting points on genus-2 curves, the first practical attempt was made in 2000 [57] by combining three different methods. First, using the Cartier-Manin operator, $\chi \bmod p$ can be computed provided that the characteristic p is not too large. Since this relates to p-adic methods and will not be used in this thesis, we do not explore this approach and focus on the two other points: the computation of $\chi \bmod \ell$ for small primes ℓ in the spirit of Schoof's algorithm, and the reconstruction of χ exploiting previous modular knowledge by a baby-step giant-step (BSGS) algorithm. By then, it was already possible to count points on a curve over a 63-bit prime field (i.e. in a 126-bit Jacobian) in about two CPU-months. However, generating a Jacobian of cryptographic size requires much heavier computations that were made possible in [62] by introducing numerous practical optimizations.

In Section 3.1, we first give an overview of these algorithms along with their complexity estimates. Section 3.2 reviews practical improvements taken mostly from [62] and how they were used to compute a cryptographic Jacobian of size 256 bits, i.e. with a 128-bit security level. A recent note by the NSA [113] advised to upgrade the security level of curve-based protocols to 192 bits, casting doubt about possibly more efficient yet still exponential attacks on ECDLP. While finding elliptic curves with this security level is not a problem, it seems quite a challenge in genus 2. With this motivation in mind, Section 3.1.2 focuses on genus-2 curves with real multiplication (RM) and how this property is used in [59] to speed-up point-counting. For non-RM curves, Section 3.3 surveys prospective improvements and ongoing research that could make it possible to design genus-2 curves that offer a 192-bit security level.

3.1 Genus-2 extensions of Schoof's algorithm

In this section, $\mathcal{C}$ is a genus-2 hyperelliptic curve over a finite field $\mathbb{F}_q$ of characteristic $p > 2$ given by an equation $y^2 = f(x)$, with f monic squarefree of degree 5. The Jacobian of $\mathcal{C}$ is denoted by $J_\mathcal{C}$ or simply by J when there is no ambiguity. The Frobenius endomorphism is denoted by π and the characteristic polynomial of its action by either χ or χ_π when there is need for disambiguation.

3.1.1 The Gaudry-Harley-Schost algorithms

We now briefly instantiate properties of hyperelliptic curves from Chapter 1 in genus 2. First, recall that the characteristic polynomial of π has the form $\chi(t) = t^4 - s_1t^3 + s_2t^2 - s_1qt + q^2$. Hence, by the Weil bounds, we are looking for the integers s_1 and s_2 which respectively satisfy $|s_1| \leq 4\sqrt{q}$ and $|s_2| \leq 6q$. To recover (s_1, s_2), we compute them modulo ℓ for sufficiently many primes ℓ. Given a fixed ℓ, we compute an ℓ-torsion divisor $D \in J[\ell]$ and test whether $-s_1(\pi^3(D) + q\pi(D))$ and $\pi^4(D) + s_2\pi^2(D) + (q^2 \bmod \ell)D$ coincide. If there is only one couple (s_1, s_2) satisfying this condition, then we can deduce $\chi \bmod \ell$ and move to the next ℓ. Were it not the case, we apply the same procedure to another torsion divisor D' to further reduce the number of candidates for (s_1, s_2) until only one remains.

Let us now switch to the problem of computing a torsion element. In genus 2, an element of J is either the neutral element P_∞, the image $P - P_\infty$ of a point on $\mathcal{C}$, twice the image of a point $2(P - P_\infty)$ or, in most cases, a divisor $D = P_1 + P_2 - 2P_\infty$ with $P_1 \neq \pm P_2$. In the latter case, we call such a divisor a "generic" divisor. In general, $\mathcal{C}$ does not have a rational torsion point so that we only look for torsion elements of the last form. In fact, even if there were non generic divisors other than the neutral element in $J[\ell]$, Kampkötter showed in [77] that generic divisors generate $J[\ell]$ so that it does no harm to miss potential non-generic elements. In genus larger than 2, we do not know of any similar result and although we still expect torsion divisors to be generic, we will have to consider some degenerate cases.

Let us consider a generic divisor $D = P_1 + P_2 - 2P_\infty$, with $P_i = (x_i, y_i)$ and write a polynomial system enforcing the fact that D is in $J[\ell]$. If $\ell D = 0$, then $\ell(P_1 - P_\infty) = -\ell(P_2 - P_\infty)$. We denote by $\langle u_i, v_i \rangle$ the respective Mumford forms of both terms, then $\ell D = 0$ is equivalent to $u_1 = u_2$ and $v_1 = -v_2$. To develop further, we give a description of the u_i and v_i, which is a genus-2 version of the division polynomials defined in [28].

Proposition 3.1. *Using the above notation and setting $D_i = P_i - P_\infty$, there exist univariate polynomials $d_0, d_1, d_2, e_0, e_1, e_2$ in $\mathbb{F}_q[x]$ such that for $\ell \geq 3$, the Mumford form $\langle u_i, v_i \rangle$ of ℓD_i is given by*

$$u_i(X) = X^2 + \frac{d_1(x_i)}{d_2(x_i)}X + \frac{d_0(x_i)}{d_2(x_i)},$$

$$v_i(X) = \frac{y_i}{e_2(x_i)}\left(e_1(x_i)X + e_0(x_i)\right).$$

In the particular case of genus-2 curves, it is known that the respective degrees of these polynomials are $2\ell^2 - 1$, $2\ell^2 - 2$, $2\ell^2 - 3$, $3\ell^2 - 1$, $3\ell^2 - 2$ and $3\ell^2 - 3$.

Rewriting the equality of the u_i, we get the following system in the variables x_1, x_2:

$$\begin{aligned} E_1(x_1, x_2) &= d_1(x_1)d_2(x_2) - d_1(x_2)d_2(x_1) = 0, \\ E_2(x_1, x_2) &= d_0(x_1)d_2(x_2) - d_0(x_2)d_2(x_1) = 0. \end{aligned} \tag{3.1}$$

This system is put in triangular form by computing $R(x_1) = \mathrm{Res}_{x_2}(E_1, E_2)$ and replacing one of the equations by $R(x_1) = 0$. Before doing so, one must actually remove a factor $(x_1 - x_2)$ that appears in both E_1 and E_2 to avoid having $R = 0$. This factor is due to the fact that if $x_1 = x_2$, then we have $P_1 = \pm P_2$, thus $\ell(P_1 - P_\infty) = \pm\ell(P_2 - P_\infty)$ and therefore $u_1 = u_2$. This is an example of solutions to our system that do not yield useful information on $J[\ell]$, we call them parasites and investigate them later on. Apart from that factor which threatened the validity of our algorithm, other parasites only increase the complexity by a constant factor. In larger genus, many more degenerate cases can occur so that a thorough analysis of those parasites is required.

Once the resultant is computed, a torsion point can be reconstructed as follows: find a root x_1 of R, possibly in an extension of $\mathbb{F}_q$. From the other equation in x_1 and x_2, deduce a value for x_2. Then, there are only four possibilities for (y_1, y_2), pick one of them to deduce two points P_1 and P_2 and finally test whether any of the combinations $P_1 \pm P_2$ leads to a torsion divisor. If it is not the case, then x_1 was a root of a parasite factor of R, so we have to consider another root. The same can be done if x_1 leads to a torsion divisor for which there are still several candidates for $(s_1, s_2) \bmod \ell$.

Actually, it may happen that even after checking the whole ℓ-torsion we may end up with more than one candidate for (s_1, s_2). When this is the case, one must remember that the (s_1, s_2) correspond to polynomials that annihilates the Frobenius action. Computing their GCD, we can first deduce a multiple of its minimal polynomial. Luckily, the degree and roots of that polynomial are enough information to recover the actual characteristic polynomial $\chi \bmod \ell$. Since this is unlikely, we do not detail that subtlety and refer to [62, Sec. 3.4] for that matter.

It is possible to eliminate all the parasites by taking into account all the equations and not only the first two. Writing that the v-coordinates of $\ell(P_1 - P_\infty)$ and $\ell(P_2 - P_\infty)$ have to be opposite amounts to the third equation

$$E_3(x_1, x_2) = e_1(x_1)e_2(x_1) - e_1(x_2)e_2(x_1). \tag{3.2}$$

Then, one can compute $R_1 = \mathrm{Res}_{x_2}(E_1, E_3)$ and apply the previous method to $\tilde{R} = \gcd(R, R_1)$ instead of R.

Following the D5 strategy of [40] mentioned in Section 1.2.2.0, one could actually recover a triangular form of the ℓ-torsion ideal I_ℓ, and perform operations in the quotient ring while handling the potential "forbidden divisions" by removing the vanishing factor from the univariate polynomial of the lex Gröbner basis of I_ℓ. This approach was considered but not used in [60] as the first strategy seems more efficient for primes smaller than 19. In [62], the D5 strategy is preferred as ℓ goes up to 31.

To do so, we compute both the resultant and subresultant of the equations E_1 and E_2 to put the system in the form

$$\begin{aligned} S_0(x_1) + x_2 S_1(x_1) &= 0, \\ R(x_1) &= 0. \end{aligned}$$

Then, taking into account the equation E_3, we clean up the parasites by computing $\tilde{R}$ and the modular inverse $\tilde{S} = S_0/S_1 \bmod \tilde{R}$. We can therefore represent the ℓ-torsion ideal by the base

$$\begin{aligned} y_2^2 &- f(x_2) \\ y_1^2 &- f(x_1) \\ x_2 &+ S(x_1) \\ &\tilde{R}(x_1). \end{aligned}$$

Once given such a representation, it is no longer necessary to factor R, as one can consider a generic $D_\ell = (x_1, y_1) + (x_2, y_2) - 2P_\infty$ in $\mathbb{F}_q[x_1, x_2, y_1, y_2]$ and test the equation $\chi(D_\ell) = 0$ in $\mathbb{F}_q[x_1, x_2, y_1, y_2]/I_\ell$.

input : A genus-2 hyperelliptic curve $\mathcal{C}$ given by a monic squarefree $f \in \mathbb{F}_q[x]$ of degree 5
output: The characteristic polynomial of the Frobenius
$w \leftarrow 1$;
while $w \leq 12q$ **do**
  $\ell \leftarrow \mathsf{NextPrime}(\ell)$;
  $w \leftarrow w \cdot \ell$;
  Compute $R(x_1) = \mathrm{Res}_{x_2}(E_1, E_2)$ and $S_0(x_1) + x_2 S_1(x_1) = \mathrm{Subres}_{x_2}(E_1, E_2)$;
  $\tilde{R} \leftarrow \mathrm{GCD}(R, \mathrm{Res}_{x_2}(E_1, E_3))$;
  $S \leftarrow S_0/S_1 \bmod \tilde{R}$;
  Deduce a basis for I_ℓ ;
  Set D_ℓ a generic divisor in $\mathbb{F}_q[x_1, x_2, y_1, y_2]/I_\ell$;
  Eliminate candidates $(s_1, s_2) \in (\mathbb{Z}/\ell\mathbb{Z})^2$ for which $\chi(D_\ell) \neq 0$;
  Deduce $(s_1, s_2) \bmod \ell$;
end
Perform a CRT to recover the actual (s_1, s_2) ;
return $\chi(t) = t^4 - s_1 t^3 + s_2 t^2 - s_1 q t + q^2$

Algorithm 4: Genus-2 point counting algorithm from [57, 62, 60]

We now follow the complexity analysis of [57, 62]. Note that [57] originally proves a complexity in $\widetilde{O}(\log^9 q)$ because it does not make use of fast arithmetic in $\mathbb{F}_q$.

Theorem 3.2. *[57, Sec. 5.4] Algorithm 4 has a complexity in $\widetilde{O}(\log^8 q)$ bit operations and a memory requirement in $O(\log^5 q)$.*

Proof. Compared to the rest of the algorithm, computing the genus-2 division polynomials takes negligible time and memory in practice: even a naive approach using the recurrence formulas of [28, Eq. (1.8)] and storing each ℓ-division polynomials yield a complexity in $O(\ell^3 \log q)$ memory bits and $O(\ell^3 \log q)$ binary operations (each step requires $O(1)$ operations on polynomials over $\mathbb{F}_q$ with degree in $O(\ell^2)$) which is within the complexity bounds we aim for.

Computing the bivariate resultant R is done by an evaluation / interpolation scheme. The degrees of E_1 and E_2 in the x_i are in $O(\ell^2)$ so that by Proposition 2.39 the polynomials R, S_1, S_2 can be computed in $\widetilde{O}(\ell^6)$ field operations using $O(\ell^4)$ interpolation points, i.e. $O(\ell^4 \log q)$ bits of memory. Since we consider polynomials of degrees in $O(\ell^4)$, the GCD computations also fit within $\widetilde{O}(\ell^4)$ field operations.

In the algebra $\mathbb{F}_q[x_1, x_2, y_1, y_2]/I_\ell$, each operation costs $\widetilde{O}(\ell^4)$ field operations and each element is stored on $O(\ell^4 \log q)$ memory bits. Finding $\chi \bmod \ell$ costs at most $O(\ell)$ operations in the algebra $\mathbb{F}_q[x_1, x_2, y_1, y_2]/I_\ell$ and a constant number of Frobenius computations, which amounts to $\widetilde{O}(\ell^4(\log q + \ell))$ field operations. During this step, only a fixed (i.e. independent of ℓ) number of elements needs to be stored, hence a memory requirement in $O(\ell^4 \log q)$ bits.

Since both the number of primes ℓ and the size of the largest ℓ to consider are in $O(\log q)$ and that each operation in $\mathbb{F}_q$ has a bit complexity in $\widetilde{O}(\log q)$, we deduce the final bit complexity in $\widetilde{O}(\log^8 q)$ and a memory requirement of $O(\log^5 q)$ bits. □

The complexity in $\widetilde{O}(\log^8 q)$ bit operations is much larger than that of Schoof's algorithm in $\widetilde{O}(\log^5 q)$ and the exponent is twice larger than that of the SEA algorithm. It is very challenging to get modular information on χ for prime numbers above 30, which is the reason why other strategies are used in practice to terminate the computations. This complexity analysis also reveals an interesting phenomenon: compared to Schoof's algorithm, applying powers of π to a generic torsion element is no longer the bottleneck in the genus-2 case. Indeed, the most costly step is the computation of a triangular form for the ℓ-torsion ideal. When g grows, it is even more conspicuous that this step is also the bottleneck of our generalizations of Schoof's algorithm.

Confronted with such a complexity bound, one may look for more favorable instances of the problem in which the bounds are more reasonable. For example, one would like to find Jacobians with "a smaller torsion". Unfortunately, such Jacobians cannot exist as they must satisfy Proposition 1.35. However, we will see in the next section that there exist families of curves whose torsion can be split into a direct sum of subspaces which are similar in size to the ℓ-torsion subgroup of an elliptic curve. Such subspaces correspond to ideals of smaller degrees than the ℓ-torsion ideal, and therefore putting them in triangular form is less costly than doing the same to the full ℓ-torsion.

3.1.2 The case of RM curves

In this section, we review how the previous point-counting algorithm can be adapted into a faster one when applied to families of curves that are equipped with a particular endomorphism. This is work of Gaudry, Kohel and Smith in [59] and it will be extended to genus-3 curves in Chapter 6.

Since we consider hyperelliptic curves defined over $\mathbb{F}_q$, the Frobenius yields an endomorphism of their Jacobians, so that $\mathbb{Z}[\pi] \subset \mathrm{End}(J)$. Furthermore, the dual $\pi^\vee$ of the Frobenius endomorphism is also in $\mathrm{End}(J)$ hence any curve has RM by $F = \mathbb{Q}(\pi + \pi^\vee)$ in the sense of Definition 1.40. However, this RM is not efficient because applying the endomorphism $\psi = \pi + \pi^\vee$ to a point of $\mathcal{C}$ costs $O(\log q)$ operations in J as it involves q-th powers. Worse, this RM is not explicit in the sense that we do not have formulas to describe the action of ψ on the curve. For our purpose, we ask for curves with an additional endomorphism that is easy to compute in the sense of the following definition.

Definition 3.3. *Let η be a real element of a number field, and let $\mathcal{C}$ be a hyperelliptic curve with RM by $\mathbb{Z}[\eta]$. We say that the real multiplication is explicit if we have explicit formulas to compute the Mumford form $\eta(P - P_\infty)$ for $P = (x, y)$ the generic point on the curve $\mathcal{C}$.*

Remark Consider $\mathbb{Q}(\pi)$ the so-called CM-field of J, then the intersection $\mathbb{Q}(\pi) \cap \mathrm{End}_{\mathbb{F}_q}(A)$ is an order $\mathcal{O}$ of $\mathbb{Q}(\pi)$ and hence it is a subring of the maximal order $\mathcal{O}_{\mathbb{Q}(\pi)}$. By a result from [145], $\mathcal{O}$ also contains a "minimal order" as it has to contain $\mathbb{Z}[\pi, \pi^\vee]$.

Let us consider a genus-2 curve $\mathcal{C}$ with explicit RM by $\mathbb{Z}[\eta]$ as in Definition 3.3. Examples of such curves are given by the family $\mathcal{C}_t : Y^2 = X^5 - 5X^3 + 5X + t$ from [138] with RM by $\mathbb{Z}[\zeta_5 + \zeta_5^{-1}]$, as well as other families due to Humbert and Mestre [101]. They are detailed in [87] along with examples of RM in higher genus.

In what follows, we assume that the curve $\mathcal{C}$ has explicit RM by $\mathbb{Z}[\eta]$. Let us denote $\psi = \pi + \pi^\vee$ and recall the expression of $\chi_\pi(t) = t^4 - s_1 t^3 + s_2 t^2 - s_1 q t + q^2$, from which we deduce the characteristic polynomial of ψ,

$$\chi_\psi(t) = t^2 - s_1 t + s_2.$$

By the previous remark, $\mathbb{Z}[\psi] \subset \mathbb{Z}[\eta]$ hence there exist two integers a and b such that $\psi = a + b\eta$. They are uniquely determined by s_1 and s_2 because

$$s_1 = \mathrm{Tr}(\psi) = 2a + b\,\mathrm{Tr}(\eta), \quad \text{and} \quad s_2 = N(\psi) = a^2 + ab\,\mathrm{Tr}(\eta) + b^2 N(\eta). \tag{3.3}$$

Contrary to Section 3.1 we do not test directly the characteristic equation of π but the equality between the last two members of $\psi\pi = \pi^2 + q = a\pi + b\eta\pi$. In other terms, we compute $a \bmod \ell$ and $b \bmod \ell$ by finding $\bar{a}$ and $\bar{b}$ in $(\mathbb{Z}/\ell\mathbb{Z})$ such that for any torsion divisor D we have

$$\bar{a}\pi(D) + \bar{b}\eta\pi(D) = \pi^2(D) + (q \bmod \ell)D.$$

This brings two advantages over the general case: first, we only have to apply powers of π up to π^2 instead of π^4, and more importantly, [59, Eq. 10] shows that both a and b are in $O(\sqrt{q})$ while s_2 is in $O(q)$. More precisely, one can prove that $|a| \leq 4\sqrt{q}$ and $|b| \leq 2(|\,\mathrm{Tr}(\eta)| + 1)\sqrt{q}$.

Yet, this improvement only reduces the number (and size) of primes ℓ to consider by a constant factor since it depends logarithmically on the width of the Hasse-Weil interval. The most significant gain lies in the structure of the ℓ-torsion, which allows us to find torsion divisors D more easily.

Let us consider a prime ℓ that splits in $\mathbb{Z}[\eta]$ into the product $\mathfrak{p}_1\,\mathfrak{p}_2$. We first detail how this can be used to split $J[\ell] \simeq (\mathbb{Z}/\ell\mathbb{Z})^4$ into a direct sum of two subspaces isomorphic to $(\mathbb{Z}/\ell\mathbb{Z})^2$.

In [59], the ideals $\mathfrak{p}_i$ are assumed to be principal because the order $\mathbb{Z}[\eta]$ has class number 1 in all the examples of RM families. This assumption is not necessary and will be removed in Chapter 6 although it still holds in the genus-3 RM family that we used for practical experiments. For simplicity, we follow [59] and make the same assumption in this chapter.

Lemma 3.4. *[59, Lemma 1] If $\mathfrak{p}$ is a principal ideal of norm ℓ in a real quadratic order $\mathbb{Z}[\eta]$, then there exists an effectively computable generator $\alpha = a + b\eta$ of $\mathfrak{p}$ with both a and b in $O(\sqrt{\ell})$.*

Computing small generators α_1 and α_2 of $\mathfrak{p}_1$ and $\mathfrak{p}_2$, we have $J[\ell] = J[\alpha_1] \oplus J[\alpha_2]$ so that any torsion divisor D can be written $D_1 + D_2$ with $D_i \in J[\alpha_i]$. We have therefore transformed the problem of finding a generic element of ℓ-torsion into finding a generic element of α_i-torsion.

To do so, we proceed exactly as in Subsection 3.1.1 but with the equation $\alpha_i(D_i) = 0$ instead of $\ell D_i = 0$. Once found D_i a generic element of $J[\alpha_i]$, we compute $\pi^2(D_i) + (q \bmod \ell)D_i$ and $k\pi(D_i)$ for any $k \in \mathbb{Z}/\ell\mathbb{Z}$ to find k_i the only value of k such that these two quantities are equal. This is summed up in Algorithm 5.

Theorem 3.5. *[59, Th. 1] Algorithm 5 has a complexity in $\widetilde{O}(\log^5 q)$ bit operations.*

Proof. Let us consider a fixed prime ℓ that splits in $\mathbb{Z}[\eta]$. For each $\mathfrak{p}_i$, we compute a small generator $\alpha_i = \beta_i + \gamma_i\eta$ as in Lemma 3.4. Since we know there is one of size $O(\sqrt{\ell})$, it is possible to find it by exhaustive search for $O(\ell)$ field operations.

We now compute a generic element $D_i \in J[\alpha_i]$ using the strategy of Section 3.1.1 except that we write $\alpha_i(P_1 - P_\infty) = -\alpha_i(P_2 - P_\infty)$ instead of $\ell(P_1 - P_\infty) = -\ell(P_2 - P_\infty)$. Since β_i and γ_i are in $O(\sqrt{\ell})$, the Mumford form of $\alpha_i(P_i)$ have coefficients whose degrees in the abscissa x_i is in $O(\ell)$. Then, following the analysis in the proof of Theorem 3.2 with equations of degree ℓ instead of ℓ^2, we prove that a generic element of $\mathrm{Ker}\,\alpha_i$ can be computed in $\widetilde{O}(\ell^3)$ field operations.

Then, finding the k_i's require two applications of π and hence $\widetilde{O}(\log q)$ field operations and at most $O(\ell)$ field operations for the exhaustive search. Deducing (a, b) from k_1 and k_2 is linear algebra in $\mathbb{F}_\ell$, which is negligible, and therefore each step in the main loop of Algorithm 5 has an overall cost of $\widetilde{O}(\ell^3)$ field operations i.e. $\widetilde{O}(\ell^3 \log q)$ bit operations.

input : q an odd prime power, and $f \in \mathbb{F}_q[X]$ a monic squarefree polynomial of degree 5 such that the curve $Y^2 = f(X)$ has explicit RM by $\mathbb{Z}[\eta]$.
output: The characteristic polynomial $\chi_\pi \in \mathbb{Z}[T]$ of the Frobenius endomorphism on the Jacobian J of the curve.

$w \leftarrow 1$;
$\ell \leftarrow 2$;
while $w < \max(|2\,\mathrm{Tr}(\eta)| + 1, 4)\sqrt{q}$ **do**
  Pick the next prime ℓ that splits in $\mathbb{Z}[\eta]$;
  Compute the ideal decomposition $\ell\,\mathbb{Z}[\eta] = \mathfrak{p}_1\,\mathfrak{p}_2$, corresponding to the eigenvalues λ_1, λ_2 of η in $J[\ell]$;
  for $i \leftarrow 1$ **to** 2 **do**
    Compute a small generator α_i of $\mathfrak{p}_i$ with coefficients in $O(\sqrt{\ell})$;
    Compute a generic element D_i in $J[\alpha_i]$;
    Find the unique $k_i \in \mathbb{Z}/\ell\,\mathbb{Z}$ such that $k_i\pi(D_i) = \pi^2(D_i) + qD_i$;
  end
  Find the unique (a, b) in $(\mathbb{Z}/\ell\,\mathbb{Z})^2$ such that $a + b\lambda_i = k_i$, for i in $\{1, 2\}$;
  $w \leftarrow w \cdot \ell$;
end
Reconstruct (a, b) using the Chinese Remainder Theorem ;
Deduce χ_π from Equations (3.3).

Algorithm 5: Overview of the genus-2 RM point-counting algorithm from [59]

Since the RM field is fixed, by Chebotarev's density theorem, half of the primes split in $\mathbb{Z}[\eta]$ and thus both the number of primes ℓ to consider and the size of the largest one are still in $O(\log q)$. Replacing ℓ by $\log q$ and adding a factor $\log q$ for the number of primes proves the theorem. □

In this section, we have actually given a simplified version of point-counting algorithms to simplify the exposition and the complexity analysis. In practice, more optimizations have to be done to achieve significant practical results, in particular when using Algorithm 4. These modifications are the focus of the following section.

3.2 Practical improvements and past results

In this section, we review practical improvements to Algorithm 4. None of them change the complexity in $\widetilde{O}(\log^8 q)$ bit operations but some enabled to halve the number of operations. We conclude by reviewing the efforts made to design a genus-2 curve with 128-bit security level, which took great advantage of the practical improvements. Both the improvements and computations were made prior to this thesis, in the early 2010's so we also give a panorama of what has changed since then in the next section.

3.2.1 Sharper modelling

Eliminating "parasites"

In Section 3.1.1, we already mentioned the possibility of parasites, i.e. extraneous factors of the resultant coming not from torsion points but from particularities of the input system. More

than the factor $(x_1 - x_2)$, it is pointed out in [57] that $d_2(x_1)^{2\ell^2-2}$ divides the resultant $R(x_1)$. Indeed, recall the equations

$$\begin{aligned} E_1(x_1, x_2) &= d_1(x_1)d_2(x_2) - d_1(x_2)d_2(x_1) = 0, \\ E_2(x_1, x_2) &= d_0(x_1)d_2(x_2) - d_0(x_2)d_2(x_1) = 0, \end{aligned} \tag{3.4}$$

and remark that if $d_2(x_1)$ vanishes, then any root of d_2 is also a common root of $E_1(x_1, \cdot)$ and $E_2(x_1, \cdot)$.

Using evaluation / interpolation techniques, we can directly avoid these parasites by evaluating $\tilde{R}(a_1) = \mathrm{Res}_{x_2}(E_1(a_1, x_2), E_2(a_1, x_2))/d_2(a_1)^{2\ell^2-2}$ and then reconstruct $\tilde{R}$ by interpolation. Knowing the degrees of the d_i, we see that $\deg \tilde{R} = 4\ell^4 - 10\ell^2 + 6$ is about twice smaller than $\deg R$. The computation of R representing most of the time spent by the algorithm, this trick almost halves the running time.

In [62], it was further noticed that d_2 has the form $f^3\delta^2$, with f the defining polynomial of the hyperelliptic curve. This also results in parasites that can be eliminated, albeit with a less spectacular decrease of $\deg(\tilde{R})$ by roughly $\ell^4/4$.

Resymmetrization

Since the ℓ-torsion has size ℓ^4, we know that $\tilde{R}$ still has about three times more parasitic factor than factors coming from actual torsion points. Notice that System 3.1 is symmetric in the variables x_1 and x_2. In [60] the symmetry is used and the change of variables $U_1 = -(x_1 + x_2)$ and $U_0 = x_1x_2$ allows to halve the degrees of the input system. Moreover, the parasites of the previous paragraph can still be tracked after resymmetrization so that an additional factor 2 is gained in the degree of $\tilde{R}$.

3.2.2 Further optimization

Lifting the ℓ-torsion

The cost of computing χ mod ℓ for an additional ℓ being in $\widetilde{O}(\ell^6 \log q)$, one may wish that there were "more small primes" to retrieve modular information. The idea behind Algorithm 4 still works when replacing ℓ by ℓ^k. Indeed, for $k > 1$ we may try to find a generic ℓ^k-torsion divisor D_k and test whether $\chi(D_k) = 0$. The modelling is done iteratively by solving systems of the form $\ell D = D_{k-1}$ with D_{k-1} an ℓ^{k-1}-torsion divisor found previously and D a divisor whose coordinates are the indeterminates. This was introduced in [57] for $\ell = 2$ and then extended for primes up to 7 in [60, 62]. In [62], the largest ℓ was constrained by memory and powers of small ℓ were used until the computations became too heavy. Today, the memory is no longer a problem but running practical tests seems to be the only way of making one's mind on the subject as it is hard to assess the difference between going for ℓ^k or trying an additional prime ℓ' of similar size. Since we do not use the torsion lifting technique thoroughly in the remainder of this thesis, we do not detail this further and refer to [62, Sec. 4] for more information.

Eliminating unpredictable parasites

Even after resymmetrization and removal of predictable parasites, $\deg \tilde{R}$ is still almost twice larger than what it should be. To eliminate the remaining parasites, it is possible to compute a third equation E_3 involving the v-coordinates of the respective Mumford forms of $\ell(P_i - P_\infty)$ as in [57], compute its resultant R_1 with either E_1 or E_2 and then the GCD of R and R_1, which is of the right degree $(\ell^4 - 1)$ in practice. Back then, it was unclear whether this step

was preferable to searching directly factors in $\tilde{R}$ with risks of useless computations leading to parasite solutions, or "cleaning" $\tilde{R}$ before factorization to ensure that no parasite remains, at the cost of almost doubling the running time. This was ruled out in [62, Sec. 3.4] by using an alternative strategy involving the recovery of a Gröbner basis for the whole symmetrized torsion ideal, and by designing another way of cleaning the parasites.

Let $S = S_0/S_1 \bmod \tilde{R}$, and let us consider the algebra $B = \mathbb{F}_q[U_1, X]/\langle \tilde{R}(U_1), X^2 + U_1 X + S(U_1)\rangle$. Let P_1 and P_2 be points of respective abscissae $X_1 = X$ and $X_2 = -U_1 - X$, and compute $\ell(P_1 - P_\infty)$ and $\ell(P_2 - P_\infty)$ in the algebra B. Note that we can handle these scalar multiplications without having to worry about the ordinates and only dealing with their respective squares $f(X_1)$ and $f(X_2)$. Ultimately, we want $\ell(P_1 - P_\infty) = -\ell(P_2 - P_\infty)$ so that the v coordinates of their Mumford forms have to be opposite. Denote $v_{1i}X + v_{0i}$ such Mumford forms. We must have $v_{11}^2 = v_{12}^2$ for $D_\ell = P_1 + P_2 - 2P_\infty$ to be an ℓ-torsion element. Experimentally, $R = \gcd(\tilde{R}, v_{11}^2 - v_{12}^2)$ has no remaining parasite factor.

Let us now explain how to recover χ as in the non-resymmetrized case. Let us consider $D_\ell = \langle X^2 + U_1 X + U_0, V_1 X + V_0\rangle$, a generic divisor in $\mathbb{F}_q[U_1, U_0, V_1, V_0]$. The first equations that D_ℓ must satisfy to be an ℓ-torsion divisor are $\tilde{R}(U_1) = 0$ and $U_0 = S(U_1)$, which determine its u-coordinate. Now, remark that writing $D_\ell = P_1 + P_2$ as above, the coordinate V_1 must be $(Y_1 - Y_2)/(X_1 - X_2)$ and the quotient V_0/V_1 must be $(X_1 Y_1 Y_2 - X_2 f(X_1))/(Y_1 Y_2 - f(X_2))$. We can actually find expressions involving only the X-coordinates for $Y_1 Y_2$ and for $Y_i^2 = f(X_i)$. For the V_1-coordinate, however, we have to consider its square which we express as

$$\frac{f(X_1) + f(X_2) - 2Y_1 Y_2}{(X_1 - X_2)^2}.$$

Plugging back the expressions of the X_i's in terms of the U_i's, we end up with expressions of the coordinates of D_ℓ involving only U_1 and U_2, which yield the following Gröbner basis for the torsion ideal $I_\ell \subset \mathbb{F}_q[X_1, X_2, V_1, V_0]$:

$$\begin{aligned} &V_0 - V_1 Z(U_1)\\ &V_1^2 - W(U_1)\\ &U_0 - S(U_1)\\ &\tilde{R}(U_1) \end{aligned}$$

with all the polynomials $\tilde{R}, S, W, Z$ of degrees $\leq (\ell^4 - 1)/2$. We refer to [62, Sec. 3] for more information about this process.

Once given such a representation, we likewise avoid the factorization of $\tilde{R}$, and directly test the equation $\chi(D_\ell) = 0$ in $\mathbb{F}_q[U_1, U_0, V_1, V_0]/I_\ell$, using the D5 strategy to avoid division by any non-invertible element in that algebra.

3.2.3 Final collision search

In Section 1.2.2, we mentioned the fact that it is possible to directly test the characteristic equation of the Frobenius in the whole Jacobian to compute (s_1, s_2). Although the complexity of such an approach is exponential, it is still very efficient in practice. Moreover, it can benefit from the knowledge of (s_1, s_2) modulo an integer m and it can also be run in parallel. A one-dimensional approach when looking only for $\#J$ is to take random divisors $D \in J$ and compute their orders to deduce factors of $\#J$ until only one possibility remains in the Hasse-Weil interval. Since Jacobians of hyperelliptic curves are often "almost cyclic", a random D has a large order

and only a few random D are expected to be necessary to determine $\#J$. This method *a priori* gives the exponent of the group rather than its order, but it can be adjusted to deduce the actual order of the group, as presented in [33][Algorithm 5.4.1]. Note that this requires bounds on the cardinal of the input group, which is not a problem in our case since they are provided by the Weil bounds.

Using a birthday paradox approach, one expect to find (s_1, s_2) with running time and memory requirement in $O(q^{3/4})$. If s_1 and s_2 are already known modulo an integer m, then the search-space is reduced by a factor m^2 and the complexity by a factor m. An idea introduced in [99] is to split the characteristic equation into two parts: one depending only on a parameter derived from s_2 and the other depending on two parameters derived from s_1 and s_2. Then, instead of directly trying random values for these 3 parameters, one stores all the possible values for the first part, and deduces bounds for the two other before performing a random collision search to determine the remaining two parameters. The main drawback of these methods is the storage requirement.

The key to avoiding storage is to look for collision of deterministic sequences which are assumed to behave as pseudo-random sequences in the complexity analysis (an assumption that is backed by practical evidence). This is inspired by Pollard's kangaroos method. To simplify the exposition, let us first assume that $m \geq 8\sqrt{q}$, so that s_1 is already completely determined and we only look for the right value of s_2. Let us split $s_2 = \bar{s_2} + m\tilde{s_2}$ with $\bar{s_2}$ already known. Denote by $K = q^2 + 1 - s_1(q+1) + \bar{s_2}$ so that $\#J = K + m\tilde{s_2}$. From the bounds on s_2 we deduce bounds on $\tilde{s_2}$ and we actually substract from K some multiple of m and $\tilde{s_2}$ by some constant to make these bounds of the form $|\tilde{s_2}| \leq B$. Let us pick a random $D \in J$ and define the "wild" and "tamed kangaroos" as

$$W = \{(K + m\sigma_2)D \mid |\sigma_2| \leq B\} \text{ and } T = \{m\sigma_2 D \mid |\sigma_2| \leq B\}.$$

When an element of $W \cap T$ is found, we are able to compute $\tilde{s_2}$. Using a birthday paradox approach, we want to compute random elements in each set until we find an element in the intersection. But we need a way of storing some elements in order to detect such a collision. Storing any of the two sets in totality is excluded since it would entail memory requirements comparable to the algorithm of [99]. A workaround is to use distinguished points, i.e. elements of J with a particular feature, such has having the 20 last bits of their u_0 coordinate equal to zero. We now fix a hash function on divisors and perform pseudo-random walks (D_i) such that D_{i+1} is determined by the hash of D_i and decide to stop the pseudo-random walk whenever it hits a distinguished divisor. We perform many such walks in W and T and only store one element per walk i.e. the final distinguished divisor. Due to the deterministic design of the pseudo-random walks, if a walk in W collides with a walk in T, then they keep colliding until the end. Hence, the distinguished divisor that is stored is also a point in $W \cap T$, which is the reason why only the last element of the walk has to be stored. We redirect to [60] for complexity analysis and optimization of the parameters. Note that similarly to the one-dimensional approach we *a priori* only get s_2 modulo the order of D, but once again this can be fixed using [33][Algorithm 5.4.1].

In the general case, however, both s_1 and s_2 are not completely known and the previous collision has to be sought in intersecting rectangles instead of intervals. Each step is made in a plane instead of a line although in practice it is better to impose a fixed proportion of one-dimensional steps in the direction corresponding to s_2, as it is much larger than s_1. Indeed, our rectangles are really flat because of the Weil bounds in $O(\sqrt{q})$ for s_1 and in $O(q)$ for s_2. To perform the pseudo-random walk, D_{i+1} is computed from D_i by adding an offset of the form $(-1)^b\alpha(q+1)mD + \beta mD$, where b, α and β depend on the hash of D_i. These quantities are

initially taken uniformly at random respectively in $\{0,1\}$, $[0,2L_1]$ and $[0,2L_2]$ with the L_i's being parameters. Note that once each tuple (b,α,β) is associated to a value of the hash function, it remains the same in order to keep our walks deterministic.

Let us now discuss on the parameters involved. First, let N be the number of points in the rectangle $\{(s_1,s_2) \mid b_1 \leq s_1 \leq B_1, b_2 \leq s_2 \leq B_2\}$. Let C be the number of chains to create: this is fixed by the user and must be large enough to avoid each chain being too long but small enough not to require too much memory. We expect to construct $O(\sqrt{N})$ points before a collision, and the user can estimate an actual value λ for this quantity. We must now decide of the probability p_D for a random divisor to be distinguished: too small will imply a larger running time while too large will be too demanding on memory. We can actually relate it to C: if we are too compute about λ points divided into C chains, then we expect each chain to have a length about λ/C. Since a chain ends when it reaches a distinguished point, its expected length is $1/p_D$. Equating the two quantities yields $p_D = C/\lambda$. To fix the parameters L_1 and L_2, let us observe that we do not want the chains to leave the intersection of the rectangles, because there is no hope to find a collision outside. On average, each chain goes a distance L_2/p_D from the center along the s_2-axis so we want L_2/p_D to be small enough compared to $B_2 - b_2$, for example one tenth of it. We do the same for L_1 but warn that along the s_1-axis, the chains can move in both directions because of the sign $(-1)^b$. In that case, by the central limit theorem we expect the chains to be at distance $2\sqrt{2/3\pi}L_1/\sqrt{p_D}$ of the center along the s_1-axis. Setting L_1 to be about one tenth of the limit and approximating the previous term, one can set $L_1 = (B_1 - b_1)\sqrt{p_D}/9$. Once again, we refer to [60] for a heuristic complexity analysis and discussion on the choice of parameters. We give more details about that in Chapter 6 in the tridimensional case.

Note that the running time of Gaudry and Schost's algorithm for one- and two-dimensional collision search depends on the overlap between the sets T and W. In [52], Galbraith and Ruprai propose a more detailed complexity analysis as well as an improved version of the algorithm in which the size of $W \cap T$ is constant.

3.2.4 A cryptographic genus-2 curve

In [62, Sec. 5], a cryptographic genus-2 curve is found after large scale computations involving a million CPU hours. More precisely, this curve is defined over $\mathbb{F}_{2^{127}-1}$ and both its Jacobian and that of its quadratic twist have order 16 times a large prime. Further properties ensuring efficiency of the group law of the associated Kummer surface are imposed but we do not describe all the details here. In particular, these conditions entail some rationality conditions that are responsible for the factor 16 in the cardinality of the Jacobian. To find a "random" curve in the sense that it has no additional remarkable property which could decrease its security level, many curves are generated and those among them who do not satisfy the requirements are discarded until one suitable curve is found. The Kummer surfaces associated to these curves can be parametrized by four parameters called theta constants. The starting set was chosen to be the set of curves whose associated theta constants have squares between -40 and 40. Among them, 83639 lead to Kummer surfaces with nice arithmetic properties.

Then, Schoof's algorithm is applied to all these curves but with an early-abort strategy ensuring that we first compute the order of J modulo small primes ℓ and discard curves such that $\#J \bmod \ell = 0$. Filtering out with $\ell = 3$ and $\ell = 5$ leaves "only" 21201 candidates. Computing $\#J \bmod 32$ and $\#J \bmod 7$ reduce that number to 3608. Schoof's algorithm was continued using this early abort approach until $\ell = 31$, for which $\#J$ is known modulo $\simeq 2^{30}$. Back then, memory requirements were too high to go further and powers of primes were used up to 2^{17}, 3^7, 5^4 and 7^2, allowing knowledge of $\#J$ modulo $N \simeq 2^{77}$. In the end, the actual (s_1, s_2)

of 586 curves were computed for a total time of roughly 1000 CPU hours per curve, using the collision search algorithm described in Section 3.2.3. The 128-bit security level Jacobian that was retained corresponds to a hyperelliptic curve defined over $\mathbb{F}_{2^{127}-1}$ by the equation

$$\begin{aligned} y^2 =& x^5 + 64408548613810695909971240431892164827x^4 \\ &+ 76637216448498510246042731975843417626\,x^3 \\ &+ 154735094972565041023366918099598639851\,x^2 \\ &+ 98557324435909905133349189668472772 22\,x \\ &+ 81689052950067229064357938692912969725. \end{aligned}$$

It has since been used in various cryptographic implementations and records such as [119, 18, 120].

To our knowledge, this example is still the only random 128-bit secure genus-2 curve in the literature and this is no wonder because of the efforts required to achieve it. Worse, to hope for a higher security level, one needs to compute modular information for larger ℓ, for a complexity in $\widetilde{O}(\ell^6 \log q)$. The goal of the next section is to survey the prospects for larger cryptographic genus-2 Jacobians. Note that there are other ways of finding such Jacobians by using the CM method or by restricting to curves with RM, but one could prefer a less structured curve as additional properties might well lead to faster attacks on the DLP, although none have been published yet.

3.3 Prospective improvements

3.3.1 Feasability of a cryptographic 384-bit Jacobian

In [62], when looking for a 256-bit Jacobian of a genus-2 curve, the Schoof-like part had to be halted at $\ell = 31$ and further modular information up to 77 bits was extracted from non-prime torsion. Counting points on a single curve over a 192-bit field without using additional prime ℓ would require a collision search over a space of size about 2^{140} and thus about 2^{70} operations. As this task has to be repeated over several hundred curves, this would not be reasonable.

On the other hand, counting points on a curve over a 192-bit field without using an exponential step would require to perform Schoof's algorithm up to $\ell = 149$, or $\ell = 109$ if we manage to recover the same amount of information as [62] by lifting the ℓ-torsion. Assuming that the collision search algorithm brings about 2^{50} bits of information on (s_1, s_2), we can decrease the largest ℓ to 79 or 73.

The previous limit was set at $\ell = 31$ because of a lack of memory. Since time complexity grows in $\widetilde{O}(\ell^6 \log q)$ while memory grows in $\widetilde{O}(\ell^4 \log q)$ when ℓ grows, we expect the memory requirements to be less of a concern. It seems to be the case indeed as our simulations lead to estimate the memory requirements to be under 500 GB even for an ℓ as large as 83. The running time, however, quickly grows beyond control as, for instance, we expect that the computation of (s_1, s_2) modulo primes ℓ up to 73 would require about 10000 CPU days.

Even up to $\ell = 53$, such computations would take about 1000 CPU days, more than 80 times the whole time needed to compute the previous cryptographic Jacobian. Worse, even assuming the modular knowledge up to $\ell = 53$ and torsion lifting identical to [62], the collision search would still have to cope with a search space of size $\simeq 2^{95}$.

In practice, almost all the running time would be spent doing either evaluations using resultant computations or collision search. These two steps being parallelizable, such computations

may not be completely impossible. However, we question the point of spending conspicuous amounts of computational power that might even not be negligible compared to the cost of discrete logarithm computations in the secure curve.

Unless further improvements are made, it seems that the only plausible alternatives for safe genus-2 curves come from RM curves or from the CM-method. Indeed, in [59] counting points on a RM-curve defined over a 512-bit prime field is done in about 80 CPU days. When using the CM-method, the order of the Jacobian is almost already determined and the bulk of the computations is actually to find suitable fields K and $\mathbb{F}_p$ and recover an equation of a curve $\mathcal{C}$ over $\mathbb{F}_p$ with CM by the ring of integers of K. When the CM-field K has a small class number as in [146], Jacobians of genus-2 curves offering a 128-bit security level can be computed in a matter of minutes. Later on, further examples with fields of larger class numbers were constructed in [44], the largest one being the field $K = \mathbb{Q}[X]/(X^4 + 1357X^2 + 3299)$, with class number 40032. In the next two subsections, we discuss research areas that could help make random genus-2 curves competitive again.

3.3.2 Further improvements

For prime numbers ℓ larger than 30, we observe that computing the bivariate resultant represents more than 90% of the running time of Algorithm 4, which is no surprise since we expected it to be the bottleneck asymptotically. Therefore, to improve the running-time of this algorithm, one must either reduce the size of the input system or find a faster way of computing the resultants. Efficient computation of resultants is a problem that has drawn a lot of attention in the past decades and for which there has been no significant improvement in the previous years. Recently, however, Villard proposed a faster algorithm [143] for computing bivariate resultants that we already mentioned at the end of Section 2.3. Using this algorithm, the cost of the computation of bivariate resultants could be decreased to $O(\ell^{6-2/\omega+o(1)})$, which represents an improvement by a factor at least $(\log q)^{2/3}$ in the final complexity. Recall that ω is the exponent from linear algebra, which was proven to be smaller than 2.38. In practice however, we expect to be using Strassen's algorithm for matrix multiplication and thus have a value of approximately 2.8 for ω. Also note that Villard's algorithm relies on some genericity assumption which may prevent us from using it in our case.

In order to reduce the size of the bivariate equations, a possibility could be to forecast and remove additional parasites. This approach, however, seems to have been fully explored in [62]. Note that since $\deg \tilde{R}$ is reduced to about $2\ell^4$ in [62] and has to be at least $\ell^4/2$ to encode the whole ℓ-torsion, no more than a factor 4 can be saved anyway. Another way to reduce the degrees of our equations could be to consider less than the full torsion, as in the SEA algorithm for which factors of division polynomials of degree $O(\ell)$ are considered instead. While we still lack the tools to make this technique a reality in genus 2, the next section reviews ongoing research in that direction.

3.3.3 Generalization of the Elkies-Atkin improvements

In the elliptic case, Schoof's algorithm has been improved by Elkies and Atkin, as detailed in [128]. Both improvements involve the so-called modular polynomial $\Phi_\ell(X, Y)$ which is a bivariate polynomial defined by the property $\Phi_\ell(j(\mathcal{E}_1), j(\mathcal{E}_2)) = 0$ if and only if the curves $\mathcal{E}_1$ and $\mathcal{E}_2$ are ℓ-isogenous.

Given a curve $\mathcal{E}$, the univariate polynomial $\Phi_\ell(j(\mathcal{E}), X)$ has a very constrained factorization pattern in $\mathbb{F}_q[X]$. Indeed, only three possibilities occur for the degrees of the irreducible factor-

ization $\Phi_\ell(j(\mathcal{E}), X) = f_1 \cdots f_s$. We denote by $(\delta_1, \ldots, \delta_s)$ the tuple formed by the degrees of the f_i's rearranged in non-decreasing order and we use the terminology of [8] to classify the primes ℓ according to the tuple associated to $\Phi_\ell(j(\mathcal{E}), X)$:

- if it is $(1, \ell)$, we say that ℓ is a volcanic prime
- if it is $(1, 1, r, \ldots, r)$, we say that ℓ is an Elkies prime
- if it is $(r, r, \ldots, r)$, we say that ℓ is an Atkin prime

The improvement by Atkin allows to deduce information on $\chi \bmod \ell$ from this factorization pattern: it does not change the asymptotic complexity of Schoof's algorithm, but provides a significant speed-up. Indeed, we have $\chi(X) = X^2 - tX + q$ and Atkin proved that $t^2 \bmod \ell$ is either $4q \bmod \ell$ in the volcanic case or $(\zeta + \zeta^{-1} + 2)q \bmod \ell$ in the other two cases, with ζ a primitive e-root of unity, for e dividing either $\ell + 1$ if ℓ is an Atkin prime or $\ell - 1$ if ℓ is an Elkies prime.

The improvement due to Elkies consists of determining $t \bmod \ell$ by replacing the test $\chi(P) = 0$ in $\mathcal{E}[\ell]$ by the test $\pi(P) = \lambda P$ in the kernel of an ℓ-isogeny determined by the factorization of $\Phi_\ell(j(\mathcal{E}), X)$. Since the kernel is given by a polynomial of degree $(\ell + 1)/2$ versus $(\ell^2 - 1)/2$ for the ℓ-division polynomial, this decreases the complexity of computing $\chi \bmod \ell$ by a factor $O(\ell)$ provided that there exists an ℓ-isogeny. This is the case when ℓ is either a volcanic or an Elkies prime but in the first case we already know much about $\chi \bmod \ell$. Heuristically, we expect Elkies and Atkin primes to represent both about 50% of all primes, but we cannot invoke the Chebotarev density theorem since we do not work in a fixed number field. Under this heuristic, by considering only Elkies primes, we expect the largest ℓ to be in $\widetilde{O}(\log q)$. Therefore, the SEA algorithm has a heuristic complexity of $\widetilde{O}(\log^4 q)$. However, although this heuristic complexity is backed by numerical experiments, Satoh and Galbraith showed in [125, Appendix A] that under GRH, the largest ℓ to consider in the SEA algorithm is in $O((\log q)^{2+\varepsilon})$.

In order to extend these improvements to point-counting in genus 2, analogues of modular polynomials were introduced in [60] along with an algorithm to compute them and experiments on their factorization patterns. Unfortunalety, the complexity estimate to obtain these polynomials is in $O(\ell^8 \log q)$ bit operations, which is more costly than the natural extension of Schoof's algorithm. In some favorable cases, i.e. when the curve has RM by a small quadratic order, Milio and Martindale [103, 98] have computed analogues of modular polynomials which could be exploited to mimic the Atkin improvement. By computing modular correspondences between Abelian varieties equipped with a theta-structure, Faugère, Lubicz and Robert propose another extension of modular polynomials in higher dimension [48]. In order to extend the Elkies improvement to the genus-2 case, current work by Couveignes and Ezome and implementations by Milio [104, 37] involve computing (ℓ, ℓ)-isogenies from their kernels, which solves a part of the problem, but we still lack an algorithm to compute the kernel itself. We also refer to the AVIsogenies software [20] for ongoing work in that direction although it requires hypotheses on the rationality of 2- and 4-torsion, and therefore in most cases to accept working in a significant extension of the base field.

Part II

Contributions

Chapter 4

Cantor's division polynomials

As explained in Chapter 1, we compute a low-degree representation of the Frobenius endomorphism by successive squarings and reductions in the ℓ-torsion ideal. To make these reductions possible, we compute the equations of the torsion ideal by formally equating $\ell D = 0$ for D a generic divisor and put them in a "nice" form by solving a polynomial system derived from $\ell D = 0$. This has a complexity cost which has to be controlled, and which depends on parameters such as the degree of the ideal, the number of variables and the degrees of the equations, as detailed in Chapter 2.

Given a generic point $P = (x, y)$ on an imaginary hyperelliptic curve, we show in this chapter that the divisor ℓP has coordinates that are rational fractions in x and y and bound their degrees. This is used in Chapters 3 to 6 to bound the degrees of polynomial systems involved in the modelling of the torsion subgroups.

These hyperelliptic counterparts to division polynomials were first described by Cantor in [28], although an alternative strategy was suggested at the same time in Kampkötter's thesis [77] to compute scalar multiplications. Cantor's paper is quite long and technical, so we do not attempt to make this chapter self-contained and advise the meticulous reader to keep it handy while browsing through our proofs. Still, in Section 4.1 we give details on how Cantor's paper works as well as some intuition behind the objects that we define to make this chapter as understandable as possible without previous knowledge of Cantor's polynomials. More precisely, we have tried to make this whole chapter self contained for the reader willing to accept statements from [28] without proof but unwilling to read the paper. Lastly, we emphasize that this section is purely technical and can easily be skipped without jeopardizing the reader's understanding of this thesis, as we only reuse the main statements proven in Sections 4.2 and 4.3.

While the description of ℓP using $2g + 2$ polynomials was established in Cantor's original paper [28], the degrees of only two of these polynomials were actually computed, whereas the polynomial systems of Chapters 3 to 6 involve all the $2g+2$ polynomials. In genus 2, the degrees of all these polynomials were computed precisely as seen in Proposition 3.1 but no result was published in larger genus, although numerical evidence suggests that the degrees are quadratic in ℓ. In Section 4.2, we prove a bound in $O_g(\ell^3)$ for the degrees of Cantor's ℓ-division polynomials in arbitrary genus. In Section 4.3, we prove that in genus 3, Cantor's polynomials have degrees in $O(\ell^2)$. These two sections are joint work with Pierrick Gaudry and Pierre-Jean Spaenlehauer and are to appear as [1, Sec. 6] and [2, Sec. 6].

4.1 Overview on division polynomials

As seen in Section 1.1.4, given a point of coordinates (x, y) on an elliptic curve and an integer $\ell > 1$, explicit formulas for the coordinates of the point ℓP have been known for long and can be described using the so-called division polynomials ψ_ℓ. Recall that these polynomials follow a recurrence formula which we restate in a different form:

$$\forall\, s \geq r \geq 1, \quad \psi_{s-r}\psi_{s+r} = \det \begin{pmatrix} \psi_{s-1}\psi_r & \psi_s\psi_{r-1} \\ \psi_s\psi_{r-1} & \psi_{s+1}\psi_r \end{pmatrix},$$

and that $\deg \psi_\ell = (\ell^2 - 1)/2$.

We can already illustrate why those degrees are important: in Schoof's algorithm they determine the size of the quotient ring in which we compute the Frobenius, and therefore the cost of each operation.

Theorem 4.1 ([28], Th. 8.35)**.** *Let $\mathcal{C}$ be a hyperelliptic curve given of genus g by an equation of the form $Y^2 = F(X)$ with F monic of degree $2g+1$. Let P be the generic point on $\mathcal{C}$, (x, y) be its coordinates and let $D = P - P_\infty$ be the associated divisor.*

For $\ell \geq g$, there exists two polynomials $\delta_\ell(X)$ and $\varepsilon_\ell(X)$ of respective degrees g and $g-1$ such that the non-normalized Mumford form of ℓD is

$$\left\langle \delta_\ell\left(\frac{x-X}{4y^2}\right), \varepsilon_\ell\left(\frac{x-X}{4y^2}\right)\right\rangle.$$

Furthermore, the coefficients of δ_ℓ are polynomials in x. And those of ε_ℓ/y are rational fractions whose numerators and common denominator are also polynomials in x.

By non-normalized Mumford form, we mean that the polynomial δ_ℓ is not monic, contrary to Definition 1.29. This is the only difference and it allows us to have polynomials as coefficients of δ_ℓ.

Definition 4.2. *Let $\ell \geq g$, the $g+1$ coefficients of the polynomials δ_ℓ, the g numerators and the common denominator of the coefficients of ε_ℓ/y are called Cantor's ℓ-division polynomials, and we omit the ℓ when there is no ambiguity on it.*

In this chapter, we study the degrees in x of these polynomials, and notably their dependency in ℓ. For a polynomial P whose coefficients are rational fractions, we denote by $\mathrm{degmax}(P)$ the maximum of the degrees of the numerators and denominators of its coefficients. In the remainder of the chapter, we aim to bound $\mathrm{degmax}(\delta_\ell)$ and $\mathrm{degmax}(\varepsilon_\ell/y)$.

Warning: Instead of the coefficients of δ_ℓ, we may also consider those of $\delta_\ell((x-X)/(4y^2))$ or more often the $2g+2$ polynomials $(d_i)_{0\leq i\leq g}$ and $(e_i)_{0\leq i\leq g}$ such that

$$\ell D = \left\langle X^g + \sum_{i=0}^{g-1} \frac{d_i(x)}{d_g(x)} X^i, y \sum_{i=0}^{g-1} \frac{e_i(x)}{e_g(x)} X^i \right\rangle.$$

The second family of polynomials is deduced from the first after developping $(x-X)/(4y^2)$, and the third comes from the second after simplifying the rational fractions. For simplicity, all of them are called division polynomials, but there is little ambiguity on their respective occurrences: the last one is the only form appearing in our systems and in practice, while we mostly focus on

the first one when proving bounds on degrees. However, the difference of degrees between them only depends on g and can readily be computed.

To simplify the exposition, the first step is a change of variable $X = x - z$ from the point $P = (x, y)$ on the curve $Y^2 = F(X)$ to the point $P_0 = (0, (-1)^{g+1}y)$ on the curve $\mathcal{C}'$ of equation $Y'^2 = E(z)$ with $E(z) = F(x - z)$. The choice of the sign of the ordinate of P_0 is well-motivated in [28], but since we ultimately only focus on the degrees of Cantor's polynomials, we prefer not to linger on signs. Denote $\sqrt{E(z)}$ the formal power series which is the Taylor series of the square root of E around $z = 0$ with constant term $(-1)^{g+1}y$. Following Cantor, we first define unnormalized division polynomials as A_ℓ, B_ℓ, C_ℓ and D_ℓ, then we normalize them by the right power of $2y$ and we invert the change of variables to recover the normalized polynomials α_ℓ, β_ℓ, γ_ℓ and δ_ℓ. Lastly, the polynomial ε_ℓ is deduced from δ_ℓ, $\delta_{\ell-1}$ and $\delta_{\ell+1}$.

Let us now consider the curve $\mathcal{C}'$, mapped in its Jacobian J' by $P \mapsto P - P_\infty$. Let $A_\ell(z)$ and $B_\ell(z)$ be polynomials such that

- z^ℓ divides $A_\ell(z) - B_\ell(z)\sqrt{E(z)}$,
- $2 \deg A_\ell \leq \ell + g$ and $2 \deg B_\ell + 2g + 1 \leq \ell + g$.

We are not sure yet whether they exist and how to compute them, but this will be dealt with once their definition becomes more natural and relevant to the initial problem.

Indeed, the function on the curve $\mathcal{C}'$ given by $A_\ell(z) - Y'B_\ell(z)$ has $\ell + h$ poles at infinity with $h \leq g$. Then, denote D the associated principal divisor, we have $D = D' + \ell P_0 - (\ell + h)P_\infty$ where D' is an effective degree-h divisor, since z^ℓ divides $A_\ell(z) - Y'B_\ell(z)$. Now, this principal divisor has to be zero in the Jacobian J', so we end up with $D' - hP_\infty = -\ell(P_0 - P_\infty)$.

For $\ell > g$ define

$$D_\ell(z) = -(A_\ell(z)^2 - B_\ell(z)^2 E(z))/z^\ell \tag{4.1}$$

as in [28, 2.3]. This definition is natural in the sense that D_ℓ is the Mumford u-coordinate of D'. Then, we define $\bar{E}_\ell(z)$ to be the corresponding v-coordinate of the Mumford form, that is $\deg \bar{E}_\ell < \deg D_\ell$ and $\bar{E}_\ell(z)^2 - E(z) \equiv 0 \bmod D_\ell$. This gives the intuition on the construction of the non-normalized division polynomials, but deeper understanding is required to define them rigorously, which actually comes with the existence and definition of A_ℓ and B_ℓ.

The first condition on these polynomials amounts to ℓ homogeneous linear conditions on their (unknown) coefficients. The degree conditions only allows for $\deg A_\ell + \deg B_\ell$ to be $\ell - 1$ so there is a total of exactly $\ell + 1$ coefficients to be determined to completely fix those two polynomials. In other terms, A_ℓ and B_ℓ are defined as Padé-Hermite approximants of the series $\sqrt{E(z)}$ modulo z^ℓ. Thus, by unicity of the Padé-Hermite approximants, either there is no solution, or there is a unique solution for A_ℓ and B_ℓ up to multiplication by a scalar. There exist algorithms to compute these Padé approximants, and the condition for their existence is the non-nullity of some Hankel determinants.

For brevity, let us define the power series $S(z) = \sqrt{E(z)}$, denote s_j being either the j-th coefficient of S or 0 if $j \leq 0$. For $m \geq 0$ and $n \geq 1$ let us define the following Hankel matrix as

$$H_{mn}(S) = \begin{pmatrix} s_{m-n+1} & s_{m-n+2} & \cdots & s_m \\ s_{m-n+2} & s_{m-n+3} & \cdots & s_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m-1} & s_m & \cdots & s_{m+n-2} \\ s_m & s_{m+1} & \cdots & s_{m+n-1} \end{pmatrix},$$

and $h_{mn}(S)$ its determinant with the convention $h_{mn}(S) = 1$ if $n = 0$ and $h_{mn}(S) = 0$ if $n \leq -1$. The non-nullity of these h_{mn} guarantees the existence of solutions to the Padé approximation problem, as stated in [28, Th. (3.5)]. These solutions are actually (up to a constant) determinants of matrices similar to H_{mn}. We do not restate them since that would not be enlightening, but properties of Padé approximants allow to define the polynomials $A_\ell(z)$, $B_\ell(z)$ as well as two other quantities which will play a crucial role: the series $C_\ell(z)$ such that

$$A_\ell(z) - B_\ell(z)S(z) = -z^\ell C_\ell(z),$$

and the f_ℓ which are polynomials in x, defined as $h_{n_{\ell+1}m_{\ell+1}}$ where n_ℓ and m_ℓ are some indices depending on ℓ and g which we do not detail. We will see later that these f_ℓ are actually non normalized versions of polynomials ψ_ℓ that extend the elliptic division polynomials in some natural way [28, Cor. 8.34].

Using properties of Padé approximants called Frobenius identities in [74, Eq. (2.5)], one can derive the following recurrence formulas:

Proposition 4.3 ([28], 3.14)**.** *For $\ell \geq g + 1$,*

$$\begin{aligned} f_{\ell-1}A_{\ell+1}(z) &= f_\ell A_\ell(z) - z f_{\ell+1}A_{\ell-1}(z), \\ f_{\ell-1}B_{\ell+1}(z) &= f_\ell B_\ell(z) - z f_{\ell+1}B_{\ell-1}(z), \\ f_{\ell-1}C_{\ell+1}(z) &= (f_\ell C_\ell(z) - f_{\ell+1}C_{\ell-1}(z))/z. \end{aligned}$$

Along with initial values given in [28, 3.10], these identities allow to compute the A, B, C and f inductively without having to compute the determinants. More importantly for our purpose, they allow to inductively bound $\mathrm{degmax}(A)$, $\mathrm{degmax}(B)$ and $\mathrm{degmax}(C)$ once the degrees of the f_i are known. In Section 4.2, we transcript them into recurrence relations involving the normalized counterparts α, β, γ and ψ of Definition 4.4 instead of directly studying the non normalized objects. This is done for a pragmatic reason: to avoid duplicating proofs and results in Cantor's paper, the non-normalized objects are used to simplify technical proofs but final results are only given in normalized form.

Let us introduce some notation: for $S(z)$ a formal power series, we denote $S[\iota_n]$ the polynomial of degree $\leq n$ obtained by truncation, i.e. $\sum_{k=0}^{n} s_k z^k$ and $S[n]$ the n-th term of the series, that is $s_n z^n$.

We restate the dictionary to switch from the non-normalized to the normalized world.

Definition 4.4 ([28], 8.7)**.** *Let $\ell \geq g + 1$ and $\nu_\ell = (\ell^2 - \ell - g^2 + g)/2$, we define*

$$\begin{aligned} \psi_\ell &= (2y)^{\nu_\ell} f_\ell, \\ \alpha_\ell(z) &= 2(2y)^{\nu_{\ell-1}-1} f_\ell A_\ell(4y^2 z)\{\iota_g\}, \\ \beta_\ell(z) &= (2y)^{\nu_{\ell-1}} f_\ell B_\ell(4y^2 z)\{\iota_g\}, \\ \gamma_\ell(z) &= (2y)^{\nu_{\ell+1}} f_\ell C_\ell(4y^2 z)\{\iota_g\}, \\ \delta_\ell(z) &= (2y)^{2\nu_\ell} D_\ell(4y^2 z), \\ \varepsilon_\ell(z) &= \bar{E}_\ell(4y^2 z). \end{aligned}$$

From the non-normalized conditions of the Mumford form, we get the following alternative expression for ε_ℓ, which allows us to focus on ψ_ℓ and $\delta_\ell(z)$.

Proposition 4.5 ([28], 8.13)**.** *For $\ell > g$,*

$$\varepsilon_\ell(z) = y \frac{z(\psi_{\ell-1}^2 \delta_{\ell+1}^2(z) - \psi_{\ell+1}^2 \delta_{\ell-1}(z))}{\psi_{\ell-1}\psi_\ell^2\psi_{\ell+1}} \mod \delta_\ell(z).$$

The five polynomials α to ε have degrees at most g in z, but their coefficients are *a priori* rational fractions in x and y. The following theorem clarifies the situation:

Theorem 4.6 ([28], 8.15)**.** *If $\ell - g$ is even, then ψ_ℓ is a polynomial in x, and $\alpha_\ell(z)/(2y)^g$, $\beta_\ell(z)/(2y)^g$ and $\gamma_\ell(z)/(2y)^g$ are polynomials in z with coefficients that are polynomials in x. If $\ell - g$ is odd, then $\psi_\ell/(2y)^g$ is a polynomial in x, and $\alpha_\ell(z)$, $\beta_\ell(z)$ and $\gamma_\ell(z)$ are polynomials in z with coefficients that are polynomials in x.*

Definition 4.7 ([28], 8.16)**.** *Let us now define P_ℓ as ψ_ℓ if $\ell - g$ is even, and $\psi_\ell/(2y)^g$ otherwise, so that P_ℓ is always a polynomial in x.*

Both its degree and leading coefficient are given but we only restate the result on the degree.

Theorem 4.8 ([28], 8.17)**.** *The polynomial P_ℓ has degree*

$$\begin{cases} \dfrac{g(\ell^2-g^2)}{2} & \text{if } \ell-g \text{ is even,} \\ \dfrac{g(\ell^2-g^2)-g(2g+1)}{2} & \text{if } \ell-g \text{ is odd.} \end{cases}$$

Using the fact that $y^2 = F(x)$, we can rephrase this by a formula which we often use to prove the following theorems: $\deg(\psi_\ell^2) = g(\ell^2 - g^2)$. We now have all the necessary ingredients for Section 4.2 but we present alternative recurrence formulas that are more similar to the elliptic case, and that also allow for sharper bounds in the genus-3 case.

This is studied in [28, Sec. 6 & 8], respectively in the non-normalized and normalized cases. Once again, the idea is to use properties of Padé approximants and translate them into recurrence relations. We focus on the results and refer to Cantor's paper for proofs, hence we restrict to the normalized case. We restate the relations [28, 8.31 to 8.33] that allow to express α_ℓ, γ_ℓ and ψ_ℓ in terms of determinants involving the polynomials α_r, γ_r and ψ_r for several values of r that are close to $\ell/2$.

Definition 4.9 ([28], 8.30)**.** *Let $s \geq r \geq 2g-1$ and $h \leq g$, define the $(g+1)\times(g+1)$ matrix*

$$\mathcal{E}_{rs}[h] = \begin{pmatrix} \alpha_{r-g}(z)\alpha_s(z)[\iota_{g-2}] & \psi_{r-g}\psi_s & \gamma_{r-s}(z)\gamma_s(z)[h] \\ \alpha_{r-g+1}(z)\alpha_{s-1}(z)[\iota_{g-2}] & \psi_{r-g+1}\psi_{s-1} & \gamma_{r-g+1}(z)\gamma_{s-1}(z)[h] \\ \vdots & \vdots & \vdots \\ \alpha_r(z)\alpha_{s-g}(z)[\iota_{g-2}] & \psi_r\psi_{s-g} & \gamma_r(z)\gamma_{s-g}(z)[h] \end{pmatrix}.$$

Definition 4.10 ([28], 8.32)**.** *Let $s \geq r \geq 2g-1$ and $h \leq g$, define the $(g+1)\times(g+1)$ matrix*

$$\mathcal{F}_{rs}[h] = \begin{pmatrix} \alpha_{r-g}(z)\alpha_s(z)[h] & \psi_{r-g}\psi_s & \gamma_{r-s}(z)\gamma_s(z)[\iota_{g-2}] \\ \alpha_{r-g+1}(z)\alpha_{s-1}(z)[h] & \psi_{r-g+1}\psi_{s-1} & \gamma_{r-g+1}(z)\gamma_{s-1}(z)[\iota_{g-2}] \\ \vdots & \vdots & \vdots \\ \alpha_r(z)\alpha_{s-g}(z)[h] & \psi_r\psi_{s-g} & \gamma_r(z)\gamma_{s-g}(z)[\iota_{g-2}] \end{pmatrix}.$$

Recall that, using previous notation, the first matrix involves $g-1$ terms in α and one term in γ while the second one involves $g-1$ terms in γ and one in α. We restate recurrence relations based on the determinants of these matrices:

Proposition 4.11 ([28], 8.31)**.** *For $s \geq r \geq 2g-1$ and $h \leq g$, we have*

$$\det \mathcal{E}_{rs}[h] = (-1)^{\binom{g+1}{2}}\gamma_{r+s-2g+1}[h]\psi_{s-r}\prod_{k=1}^{g-1}\psi_{r-g+k}\psi_{s-g+k}.$$

Proposition 4.12 ([28], 8.33)**.** *For $s \geq r \geq 2g-1$ and $h \leq g$, we have*

$$\det \mathcal{F}_{rs}[h] = (-1)^{\binom{g+1}{2}} \alpha_{r+s-2g+1}[h] \psi_{s-r} \prod_{k=1}^{g-1} \psi_{r-g+k} \psi_{s-g+k}.$$

When $g = 2$, and for $s \geq r \geq 3$ these formulas yield

$$\psi_s \psi_r \psi_{s+r} \psi_{s-r} = \det \begin{pmatrix} \psi_{s-2}\psi_r & \psi_{s-1}\psi_{r+1} & \psi_s\psi_{r+2} \\ \psi_{s-1}\psi_{r+1} & \psi_s\psi_r & \psi_{s+1}\psi_{r+1} \\ \psi_s\psi_{r+2} & \psi_{s+1}\psi_{r-1} & \psi_{s+2}\psi_r \end{pmatrix}.$$

This can be used to compute the exact degrees of Cantor's division polynomials in genus 2, which is used in Chapter 3. Another important remark is that when $g = 1$ they give exactly the same recurrence as the one satisfied by the division polynomials, and the immediate corollary is that in genus 1 the ψ_ℓ coincide with the previously known division polynomials.

To our knowledge, apart from the leading and constant coefficients of δ_ℓ, which Cantor proved to be respectively $-(4y^2)^g \psi_\ell^2$ and $(-1)^{g+1}\psi_{\ell-1}\psi_{\ell+1}$ even in arbitrary genus, no additional proven result was published for $g \geq 3$. In Section 4.3, we instantiate these recurrence formulas in genus 3 and use them to prove a quadratic bound in ℓ on the degrees of all the 8 analogues of division polynomials. We also explain why we are not very optimistic about that approach compared to the first one when g is larger.

4.2 A cubic bound in any genus

We prove the following:

Theorem 4.13. *For any integer $\ell > g$, the polynomial $\delta_\ell(X)$ of degree g in X has coefficients in $\mathbb{F}_q[x]$ whose degrees in x are bounded by $g\ell^3/3 + O_g(\ell^2)$; the polynomial $\varepsilon_\ell(X)/y$ has coefficients in $\mathbb{F}_q(x)$ such that the degrees of the numerators and the denominators have degrees bounded by $2g\ell^3/3 + O_g(\ell^2)$. Furthermore, the roots of the denominators are roots of the leading coefficient of $\delta_\ell(X)$.*

Proof. Technicalities arise from the normalizations required to manipulate entities that are polynomials in x (and not rational fractions), without odd power of y involved. In Cantor's article, this normalization often depends on the parity of $\ell - g$. We will concentrate on the case where g is even; for the other case some formulas must be adapted, multiplying or dividing by $2y$ at various places.

We recall that $\nu_\ell = (\ell^2 - \ell - g^2 + g)/2$ as defined in (8.7), so that $\nu_\ell = \nu_{\ell-1} + \ell - 1$. By combining Definition 4.4 and 4.1, we obtain

$$\delta_\ell(z) = \frac{(2y)^{2\nu_\ell}}{(4y^2 z)^\ell} \left(A_\ell(4y^2 z)^2 - B_\ell(4y^2 z)^2 E(4y^2 z) \right),$$

where A_ℓ and B_ℓ are unnormalized versions of α_ℓ and β_ℓ given in Definition 4.4 and $E(z)$ is defined by $E(z) = f(x - z)$. For our purpose, it is easier to deal with non-truncated versions of α_ℓ and β_ℓ. Let us then introduce the following quantities, inspired by Definition 4.4:

$$\bar{\alpha}_\ell(z) = 2(2y)^{\nu_{\ell-1}-1} A_\ell(4y^2 z), \quad \text{and} \quad \bar{\beta}_\ell(z) = (2y)^{\nu_{\ell-1}} B_\ell(4y^2 z),$$

so that δ_ℓ can be rewritten as

$$\delta_\ell(z) = \frac{1}{4z^\ell}\left(\bar{\alpha}_\ell(z)^2 - \frac{1}{y^2}\bar{\beta}_\ell(z)^2 E(4y^2 z)\right).$$

By Theorem 4.6, the coefficients of $\alpha_\ell(z)$ and $\beta_\ell(z)$ are polynomials in $\mathbb{F}_q[x]$, and the proof is also valid for the non-truncated versions $\bar{\alpha}_\ell(z)$ and $\bar{\beta}_\ell(z)$. Note that here we use the fact that g is even, so that the potential adjusting factor $(2y)^g$ is an even power of y that can be rewritten in terms of $F(x)$. The polynomial $E(4y^2z)$ has coefficients which are polynomials in x of degree bounded by $(2g+1)^2$. Therefore, in order to obtain a degree bound for the coefficients of $\delta_\ell(z)$, it is sufficient to bound the coefficients of $\bar{\alpha}_\ell(z)$ and $\bar{\beta}_\ell(z)$.

We are interested in a bound for fixed genus g and when ℓ grows to infinity and we use the O_g notation as a O notation which also hides factor depending only on g (and not on ℓ). For k in $[1,\ell]$, we will use an induction to bound $\mathrm{degmax}(\bar{\alpha}_k(z))$ and $\mathrm{degmax}(\bar{\beta}_k(z))$. For $k \leq g+1$, none of these quantities depends on ℓ, so that all the degrees can be bounded by an expression in g only, *i.e.* in $O_g(1)$. For $k \geq g+1$, we start from Proposition 4.3 where we substitute k for ℓ, we evaluate it at $4y^2z$, and we multiply by $(2y)^{2\nu_k-1}$, so that we obtain:

$$(2y)^{\nu_k} f_{k-1}\bar{\alpha}_{k+1}(z) = (2y)^{\nu_k+k-1} f_k \bar{\alpha}_k(z) - (2y)^{\nu_k+2k-1} f_{k+1} z \bar{\alpha}_{k-1}(z),$$

where all the polynomials have coefficients in $\mathbb{F}_q[x]$. The expression for $\bar{\beta}_k$ is exactly the same, but we have to multiply the expression in Proposition 4.3 by $(2y)^{2\nu_k}$ in that case. By Definitions 4.4 and 4.7 plus Theorems 4.6 and 4.8, for any k, the quantity $(2y)^{\nu_k} f_k$ is a polynomial in x of degree $g(k^2-g^2)/2$. Therefore the right-hand-side of the recurrence relation has coefficients with degrees bounded by an expression of the form $\max\big(\mathrm{degmax}(\bar{\alpha}_k(z)), \mathrm{degmax}(\bar{\alpha}_{k-1}(z))\big) + gk^2/2$, up to a term linear in k and cubic in g. We finally get:

$$\mathrm{degmax}(\bar{\alpha}_{k+1}(z)) \leq \max\big(\mathrm{degmax}(\bar{\alpha}_k(z)), \mathrm{degmax}(\bar{\alpha}_{k-1}(z))\big) + gk^2/2 + \mathrm{Err}_g(k),$$

where $\mathrm{Err}_g(k)$ is a polynomial linear in k and cubic in g. Again, this inequality is also valid for $\bar{\beta}_k$. By induction, we then get the following bounds:

$$\mathrm{degmax}(\bar{\alpha}_\ell(z)) \leq \frac{g\ell^3}{6} + O_g(\ell^2), \quad \text{and} \quad \mathrm{degmax}(\bar{\beta}_\ell(z)) \leq \frac{g\ell^3}{6} + O_g(\ell^2).$$

We can then propagate these bounds in the expression of δ_ℓ and we get $\mathrm{degmax}(\delta_\ell(z)) \leq \max(2\,\mathrm{degmax}(\bar{\alpha}_\ell(z)), 2\,\mathrm{degmax}(\bar{\beta}_\ell(z)) + \mathrm{degmax}(E(4y^2z))$, so that we get the claimed result concerning δ_ℓ.

The fact that $\varepsilon_\ell(z)/y$ has coefficients in $\mathbb{F}_q(x)$ follows directly from Proposition 4.5 that we recall here:

$$\varepsilon_\ell(z) = y\frac{z\left(\psi_{\ell-1}^2\delta_{\ell+1}(z) - \psi_{\ell+1}^2\delta_{\ell-1}(z)\right)}{\psi_{\ell-1}\psi_\ell^2\psi_{\ell+1}} \bmod \delta_\ell(z).$$

As stated in [28, 8.11], the leading coefficient of $\delta_\ell(z)$ is $-(4y^2)^g\psi_\ell^2$, so that the property on the denominator of ε_ℓ can not be easily deduced from this equation, due to the presence of $\psi_{\ell-1}$ and $\psi_{\ell+1}$ before the reduction modulo $\delta_\ell(z)$ occurs. We will prove it below, with a direct geometric argument, but we first give bounds on the degrees of the coefficients of the numerator and the denominator.

The polynomial $\delta_\ell(z)$ is of degree g in z, so that at most two steps of reduction are required to reduce the degree of ε_ℓ to strictly less than g. In fact, it can be checked that $\mathrm{LT}(\psi_{\ell-1}^2\delta_{\ell+1}(z)) =$

$\mathrm{LT}(\psi_{\ell+1}^2\delta_{\ell-1}(z))$, so that there is at most only one reduction step. This reduction accounts for an increase of the coefficients' degrees in x by at most $\mathrm{degmax}(\delta_\ell)$ in the numerator and an increase of the degree of the leading coefficient of δ_ℓ in the denominator. Since $\deg\psi_\ell = g\ell^2/2 + O_g(\ell)$, the degrees of the coefficients of the numerator of $\varepsilon_\ell(z)$ are bounded by $\frac{2}{3}g\ell^3 + O_g(\ell^2)$, and the degree of the denominator is bounded by $3g\ell^2 + O_g(\ell)$.

It remains to prove the claim on the roots of the denominator of the coefficients of $\varepsilon_\ell(z)/y$. For this, we consider the map from the affine part of the curve $\mathcal{C}_{\mathrm{aff}}$ to J seen as a projective Abelian variety, that sends a point (x, y) to $[\ell]((x, y) - \infty)$. One of the main points of Cantor's article is that if $\psi_\ell(x) \neq 0$, then the image by this map is in $J \setminus \Theta$, where $\Theta \subset J$ is the subvariety of elements of weight less than g (i.e. divisors that are sums of less than g points). On this open subset, Mumford coordinates with a monic u of degree g and v of degree at most $g-1$ give a local set of coordinates that we use to describe the map. The i-th coefficient of v is y times a rational fraction c_i in x that gives a finite value at any x for which $\psi_\ell(x) \neq 0$. Therefore, any root of the denominator of c_i is a root of ψ_ℓ. By Theorem 4.1, the Mumford v-polynomial that we are considering is ε_ℓ up to a renormalization that will only introduce additional powers of $4y^2$ in the denominator. Therefore, any root of the denominator of the coefficients of ε_ℓ is a root of ψ_ℓ or of $4y^2$, and both divide the leading coefficient of δ_ℓ, which is $-(4y^2)^g\psi_\ell^2$. □

Remark. The bounds that we obtain are not tight: from [28], we know that the leading and constant coefficients are in $O_g(\ell^2)$ instead of $O_g(\ell^3)$. We ran experiments that allow us to conjecture precise degrees for the other coefficients. In these experiments, instead of developing $\delta_\ell\left(\frac{x-X}{4y^2}\right)$ and $\varepsilon_\ell\left(\frac{x-X}{4y^2}\right)$ to compute the d_i's and e_i's, we computed $\ell((x, y)-\infty)$ over the function field of the curve. This does not exactly yield the d_i's and e_i's because we actually get d_i/d_g and e_i/e_g, thus possibly missing a common factor in all the d_i's and e_i's. We denote $\tilde{d}_i$ and $\tilde{e}_i$ the numerators and denominators of the aforementioned fractions, and we compute their degrees for each pair (g, ℓ) with $g \leq 8$ and $g < \ell \leq g + 20$ (which includes non prime values of ℓ). We found that the degrees of the $\tilde{d}_i$ are consecutive from $\deg(\tilde{d}_g)$ up to $\deg(\tilde{d}_0) = \deg(\tilde{d}_g) + g$, with the following values for $\deg(\tilde{d}_0)$.

$$\begin{cases} g\ell^2 - g^3 + g & \text{if } g-\ell \text{ is even} \\ g\ell^2 - g^3 + 2g^2 - 1 & \text{if } g-\ell \text{ is odd} \end{cases}$$

Concerning the $\tilde{e}_i$, the degrees are consecutive from $\deg(\tilde{e}_{g-1})$ up to $\deg(\tilde{e}_0) = \deg(\tilde{e}_g)$, the latter being equal to

$$\begin{cases} 3(g\ell^2 - g^3)/2 + 2g^2 - g - 1 & \text{if } g-\ell \text{ is even} \\ 3(g\ell^2 - g^3)/2 + 3g^2 - g/2 - 1 & \text{if } g-\ell \text{ is odd} \end{cases}$$

Cantor [28] gave simple expressions for the leading term and constant term of δ_ℓ (respectively $-(4y^2)^g\psi_\ell^2$ and $(-1)^{g+1}\psi_{\ell-1}\psi_{\ell+1}$), from which we can deduce the degrees of d_0 and d_g by evaluating δ_ℓ at $(x - X)/4y^2$. Assuming that there is no common factor to all the d_i's when $g-\ell$ is even, while the GCD of all the d_i's is f^{g-1} when $g-\ell$ is odd, these theoretical degrees are consistent with our experiments.

4.3 A quadratic bound in genus 3

The previous cubic bound was sufficient because its error only affects the final complexity bound by a constant in some $O()$. In genus 3, however, we want to compute the exact exponent of $\log q$

in the complexity so we need a bound that is quadratic in ℓ. We do so by using other induction formulas.

Theorem 4.14. *In genus 3, the degrees of Cantor's ℓ-division polynomials are bounded by $O(\ell^2)$.*

We first prove a bound on the degrees of the coefficients of the quantities α_r and γ_r defined in [28], from which the wanted bounds will follow. The key tools are Propositions 4.11 and 4.12 that relate quantities at index ℓ to quantities at index around $\ell/2$, in a similar fashion as for the division polynomials of elliptic curves. More precisely, the following lemma shows that when the index ℓ is (roughly) doubled, $\operatorname{degmax}\alpha_\ell$ and $\operatorname{degmax}\gamma_\ell$ are roughly multiplied by 4, which leads to the expected quadratic growth.

Lemma 4.15. *Let $\ell \geq 10$, and assume that for all $i \leq (\ell+9)/2$ the degrees $\operatorname{degmax}\alpha_i$ and $\operatorname{degmax}\gamma_i$ are bounded by C, then $\operatorname{degmax}\alpha_\ell$ and $\operatorname{degmax}\gamma_\ell$ are bounded by $4C + 36\ell + 108$.*

Proof. We first deal with the bound on $\operatorname{degmax}\gamma_\ell$. Let us consider r and s around $\ell/2$ such that $\ell = r + s - 5$: we take either $r = s - 3 = \ell/2 + 1$ if ℓ is even, or $r = s - 4 = (\ell+1)/2$ otherwise.

From Definition 4.9 and Proposition 4.11, the degree of $\gamma_\ell[h]\psi_{s-r}\psi_{r-2}\psi_{s-2}\psi_{r-1}\psi_{s-1}$ is that of the determinant of the matrix $\mathcal{E}_{rs}[h]$ defined by:

$$\mathcal{E}_{rs}[h] = \begin{pmatrix} \alpha_{r-3}\alpha_s[0] & \alpha_{r-3}\alpha_s[1] & \psi_{r-3}\psi_s & \gamma_{r-3}\gamma_s[h] \\ \alpha_{r-2}\alpha_{s-1}[0] & \alpha_{r-2}\alpha_{s-1}[1] & \psi_{r-2}\psi_{s-1} & \gamma_{r-2}\gamma_{s-1}[h] \\ \alpha_{r-1}\alpha_{s-2}[0] & \alpha_{r-1}\alpha_{s-2}[1] & \psi_{r-1}\psi_{s-2} & \gamma_{r-1}\gamma_{s-2}[h] \\ \alpha_r\alpha_{s-3}[0] & \alpha_r\alpha_{s-3}[1] & \psi_r\psi_{s-3} & \gamma_r\gamma_{s-3}[h] \end{pmatrix}.$$

Therefore we have an expression for the degrees of the coefficients of γ_ℓ in terms of objects at index around r and s:

$$\deg\gamma_\ell[h] \leq \deg\det\mathcal{E}_{rs}[h] - \deg(\psi_{r-2}\psi_{s-2}\psi_{r-1}\psi_{s-1}).$$

In this last formula, the factor ψ_{s-r} has been omitted, because $s-r$ is either 3 or 4, and by Theorem 4.8 this has non-negative degree in any case. Thus, we simply bounded it below by 0 in the previous inequality. Before entering a more detailed analysis, we use the fact that $\alpha_k(0) = \psi_{k-1}$ and $\gamma_k(0) = \psi_{k+1}$ (this is [28, (8.8)]) to rewrite the first column with expressions for which we have exact formulas for the degree:

$$\mathcal{E}_{rs}[h] = \begin{pmatrix} \psi_{r-4}\psi_{s-1} & \alpha_{r-3}\alpha_s[1] & \psi_{r-3}\psi_s & \gamma_{r-3}\gamma_s[h] \\ \psi_{r-3}\psi_{s-2} & \alpha_{r-2}\alpha_{s-1}[1] & \psi_{r-2}\psi_{s-1} & \gamma_{r-2}\gamma_{s-1}[h] \\ \psi_{r-2}\psi_{s-3} & \alpha_{r-1}\alpha_{s-2}[1] & \psi_{r-1}\psi_{s-2} & \gamma_{r-1}\gamma_{s-2}[h] \\ \psi_{r-1}\psi_{s-4} & \alpha_r\alpha_{s-3}[1] & \psi_r\psi_{s-3} & \gamma_r\gamma_{s-3}[h] \end{pmatrix}.$$

The determinant of $\mathcal{E}_{rs}[h]$ is the sum of products of 4 ψ factors and 4 α or γ factors. The degrees of the former are explicitly known, while by hypothesis we have upper bounds on the latter, since all the indices are at most $(\ell+9)/2$. We can then deduce an upper bound on the degree of this determinant. All the ψ_i have indices with i in the range $[r-4, s]$ (remember that $r \leq s$), and since their degrees increase with the indices, we can upper bound the degree of the products of the four ψ factors by $4\deg\psi_s$. Therefore we have

$$\deg\det\mathcal{E}_{rs}[h] \leq 4(\deg\psi_s + C).$$

In order to deduce an upper bound on $\operatorname{degmax}\gamma_\ell$, it remains to get a lower bound on the degree of the $\deg(\psi_{r-2}\psi_{s-2}\psi_{r-1}\psi_{s-1})$ term, and again by monotonicity of the degree in the index, it is bounded below by $4\deg\psi_{r-2}$. So finally, we get

$$\operatorname{degmax}\gamma_\ell \leq 4C + (\deg\psi_s^4 - \deg\psi_{r-2}^4).$$

Using Definition 4.7 and Theorem 4.8, we deduce that for all k, we have $\deg(\psi_k^2) = 3(k^2-9)$ and substituting this value and the expression of $r-2$ and s in term of ℓ, we obtain

$$\deg\psi_s^4 - \deg\psi_{r-2}^4 = \begin{cases} 30\ell + 90 & \text{if } \ell \text{ is even,} \\ 36\ell + 108 & \text{if } \ell \text{ is odd,} \end{cases}$$

and the result follows for $\operatorname{degmax}\gamma_\ell$.

The proof for $\operatorname{degmax}\alpha_\ell$ follows the same line. Using the matrix $\mathcal{F}_{rs}[h]$ of Definition 4.10 in a similar way as we used the matrix $\mathcal{E}_{rs}[h]$ and with the help of Proposition 4.12, we end up with the following bounds

$$\operatorname{degmax}\alpha_\ell \leq \begin{cases} 4C + 30\ell - 30 & \text{if } \ell \text{ is even,} \\ 4C + 36\ell - 36 & \text{if } \ell \text{ is odd,} \end{cases}$$

which are stricter than our target.

Finally, the bound $\ell \geq 10$ is sufficient to ensure that the quantities r and s are at least 5, as required to apply Propositions 4.11 and 4.12. This condition would still hold for ℓ as small as 8 but our recurrence needs $\ell > 9$ to propagate, or else $(\ell+9)/2$ would be greater or equal to ℓ. □

We can now finish the proof of Theorem 4.14. We define two sequences $(\ell_i)_{i\geq 0}$ and $(C_i)_{i\geq 0}$ as follows: let $\ell_0 = 10$ and let C_0 be a bound on the degrees of the coefficients of all the α_i and γ_i for $i \leq \ell_0$. Then for all $i \geq 1$, we define the sequences inductively by

$$\begin{cases} \ell_{i+1} = 2\ell_i - 9 \\ C_{i+1} = 4C_i + 36\ell_{i+1} + 108. \end{cases}$$

By Lemma 4.15, for all i, and all $\ell \leq \ell_i$, the degrees $\operatorname{degmax}\alpha_\ell$ and $\operatorname{degmax}\gamma_\ell$ are bounded by C_i. The expression $\ell_i = (\ell_0 - 9)2^i + 9 = 3\cdot 2^i + 9$ can be derived directly from the definition and substituted in the recurrence formula of C_{i+1} to get $C_{i+1} = 4C_i + 216\cdot 2^i + 432$. This recurrence can be solved by setting $\Gamma_i = C_i + 108\cdot 2^i + 144$, so that $\Gamma_{i+1} = 4\Gamma_i$, and we obtain $C_i = (C_0 + 252)\,4^i - 108\cdot 2^i - 144$. Finally, for any ℓ, we select the smallest i such that $\ell \leq \ell_i$. This value of i is $\lceil \log_2((\ell-9)/3)\rceil$. The corresponding bound for $\operatorname{degmax}\alpha_\ell$ and $\operatorname{degmax}\gamma_\ell$ is then C_i, which grows like $O(\ell^2)$ (and we remark that the effect of the ceiling can make the constant hidden in the $O()$ expression grow by a factor at most 3).

Using [28, Eq. 8.10], i.e. $\delta_\ell(z) = \alpha_\ell(z)\gamma_\ell(z)\{\iota_g\}$, we have $\operatorname{degmax}\delta_\ell \leq \operatorname{degmax}\alpha_\ell + \operatorname{degmax}\gamma_\ell$, and therefore the bound $O(\ell^2)$ also applies to the degrees of the coefficients of δ_ℓ. And by Proposition 4.5, the same holds as well for the coefficients of ϵ_ℓ/y.

This concludes the proof of Theorem 4.14.

Remark. One could try to extend this method to larger g in the hope of getting a better bound than the cubic one proven in the previous section. In a nutshell, the quadratic bound was achieved because the 4×4 determinants involve 2 terms in either α^2 or γ^2 and 2 terms in

ψ^2. This led to a bound on the degrees of α and γ that was mutiplied by 4 each time ℓ was multiplied by 2. In larger genus, however, the balance between the two types of terms is broken because the $(g+1) \times (g+1)$ determinant is made up of $(g-1)$ terms in α^2 and β^2. A direct generalization of our method would therefore give a bound B_ℓ on the degrees of α and β that is multiplied by $2(g-1)$ each time ℓ is multiplied by 2. In particular, for $g \geq 5$, the growth of B_ℓ seems already worse than cubic.

Chapter 5

Asymptotic complexity bounds in arbitrary genus

Let $\mathcal{C}$ be a hyperelliptic curve of genus g over a finite field $\mathbb{F}_q$ of characteristic p and denote by J its Jacobian. In this chapter, we present a Las Vegas algorithm derived from Schoof and Pila's approaches to count points on hyperelliptic curves that achieves a time complexity in $O((\log q)^{cg})$ for c a constant and g fixed with q growing and p large enough. This is joint work with Pierrick Gaudry and Pierre-Jean Spaenlehauer and most of this chapter is to appear as [1].

Organization of the chapter. Section 5.1 describes a general algorithm for point-counting on Abelian varieties along with its complexity, assuming that the ℓ-torsion can be efficiently computed. Section 5.2 establishes the complexity result for multi-homogeneous polynomial systems that is required to obtain our claimed complexity bound. Section 5.3 contains the modelling of the ℓ-torsion under some mild assumptions on its structure. Finally, Section 5.4 describes the complete modelling of the ℓ-torsion, which is faithful even if the assumptions required in Section 5.3 are not satisfied.

5.1 Overview

This chapter aims to give a proof of the following result:

Theorem 5.1. *There exists an explicitly computable constant c such that for all genus g, there exists an integer $q_0(g)$ such that for all prime power $q = p^n$ larger than $q_0(g)$ with $p \geq (\log q)^{cg}$ and for all imaginary hyperelliptic curves $\mathcal{C}$ of genus g defined over $\mathbb{F}_q$, the numerator L of the local zeta function of $\mathcal{C}$ from Proposition 1.42 can be computed with a probabilistic algorithm in expected time bounded by $(\log q)^{cg}$.*

This complexity result is summarized by the notation $O_g((\log q)^{O(g)})$, keeping in mind that g is fixed and q grows to infinity. Indeed, such a complexity statement can hide any factor that depends only on g: a running time in $f(g)(\log q)^{cg}$ can be transformed into $(\log q)^{c'g}$ by taking a value c' larger than c and adjusting $q_0(g)$, so that $|f(g)| \leq (\log q_0(g))^{(c'-c)g}$.

A typical example is the multiplication of two polynomials of degree $d = (\log q)^{O(g)}$. Using FFT-based techniques, this can be done in $\widetilde{O}(d)$ operations, which can be rewritten as $(\log q)^{O(g)}(\log((\log q)^{O(g)}))^k$ for some constant k and is therefore again in $O_g(\log(q)^{O(g)})$. Here the function $f(g)$ that has been hidden in the operation is polynomial in g, but we will have

cases where it is a combinatorial factor that grows very quickly with g and we make no effort to optimize it.

The algorithm that allows to prove the theorem is essentially the same as the one proposed by Pila for Abelian varieties, which is itself inspired by Schoof's algorithm for counting points on elliptic curves. Pila's algorithm reconstructs the numerator of the local zeta function of $\mathcal{C}$ by computing the action of the Frobenius on the ℓ-torsion for sufficiently-many prime numbers ℓ and by using the Chinese Remainder Theorem. A bird's eye view of this algorithm is given in Algorithm 6. The main difficulty resides in the step where one computes an explicit description of $J[\ell]$. Since $J[\ell]$ is a 0-dimensional variety of degree ℓ^{2g}, what we will compute is a geometric resolution of the corresponding radical ideal, that is a univariate squarefree polynomial $F_\ell(T)$, together with $2g$ coordinate polynomials $\gamma_i(T)$, such that the coordinates of the ℓ-torsion elements are the evaluations of the vector $(\gamma_1(T), \ldots, \gamma_{2g}(T))$ at the roots of F_ℓ.

input : $q \in \mathbb{Z}_{>0}$ a prime power, and $f \in \mathbb{F}_q[X]$ a monic squarefree of degree $2g+1$.
output: The characteristic polynomial $\chi \in \mathbb{Z}[T]$ of the Frobenius endomorphism on the Jacobian J of the hyperelliptic curve defined over $\mathbb{F}_q$ with Weierstrass form $Y^2 = f(X)$.

$\ell \leftarrow 1$;
$R \leftarrow 1$;
while $R \leq 2\binom{2g}{g}q^g + 1$ **do**
  $\ell \leftarrow$NextPrime(ℓ);
  if ℓ *divides* q **then**
    $\ell \leftarrow$NextPrime(ℓ);
  end
  Compute a description of $J[\ell]$;
  Compute a $2g \times 2g$ matrix F with coefficients in $\mathbb{Z}/\ell\mathbb{Z}$ representing the action of the Frobenius on $J[\ell] \cong (\mathbb{Z}/\ell\mathbb{Z})^{2g}$;
  Compute the characteristic polynomial χ mod ℓ of the matrix F;
  $R \leftarrow R \cdot \ell$;
end
Reconstruct χ using the Chinese Remainder Theorem.

Algorithm 6: A bird's eye view of Pila's point counting algorithm for hyperelliptic curves.

To be more precise, the Mumford coordinates are in fact a set of g affine systems of coordinates, each corresponding to a different weight of the represented divisor (the definition is recalled in Section 5.3). The variety $J[\ell]$ will accordingly be represented by a set of g geometric resolutions, each encoding ℓ-torsion divisors of a given weight $w \in [1, g]$. Generically, we expect that all the elements in $J[\ell]$ have weight g, except for the neutral element which has weight 0. Most of the chapter is dedicated to computing efficiently this representation for $J[\ell]$. The cornerstone of the proof of Theorem 5.1 relies on the following statement.

Proposition 5.2. *Let $\mathcal{C}$ be a hyperelliptic curve of genus g over $\mathbb{F}_q$ with Weierstrass form $Y^2 = f(X)$ (f monic squarefree of degree $2g+1$) and J be its Jacobian variety. Let $\ell > g$ be a prime not dividing q. Assuming that the characteristic of $\mathbb{F}_q$ is sufficiently large as in Theorem 5.1, there is a Las Vegas probabilistic algorithm which takes as input q, ℓ, f and which computes geometric resolutions for the varieties $\{J_w[\ell]\}_{w\in[1,g]}$ of ℓ-torsion points of weight w in the Jacobian variety. This algorithm can be implemented by a Turing machine with space and*

expected time $O_g\left((\ell \log q)^{O(g)}\right)$.

Assuming this complexity bound, performing a complexity analysis as done in [114] leads to a complexity bound for Algorithm 6 that corresponds to Theorem 5.1. We recall it here for completeness, with some simplifications due to the fact that we consider a probabilistic algorithm, so we can factor polynomials using Cantor-Zassenhaus' algorithm.

Proof of Theorem 5.1 assuming Proposition 5.2. By Weil's bounds, the absolute values of the coefficients of the characteristic polynomial χ are bounded by $\binom{2g}{g}q^g$. Therefore at the end of the loop of Algorithm 6, these coefficients are completely determined by their values modulo all the primes ℓ that have been explored. It follows from [140, Cor. 10.1] that the largest ℓ in the loop is at most linear in $g \log q$. From this and Proposition 5.2, computing the description of $J[\ell]$ as a union of geometric resolutions for all the $J_w[\ell]$ can be achieved within expected complexity $O_g\left((\log q)^{O(g)}\right)$.

Factoring the univariate polynomials involved in the geometric resolutions can be done within the same time bound $O_g\left((\log q)^{O(g)}\right)$, since the sum of their degrees is ℓ^{2g} and factoring polynomials in finite fields can be done in time linear in $\log(q)$ and quasi-quadratic in the degree [54, Thm. 14.14]. Therefore, it is possible to construct a Mumford representation for each ℓ-torsion divisor within the same complexity, each of them possibly defined over a different extension of $\mathbb{F}_q$. In fact, due to the rationality of the group law that acts on $J[\ell]$, one of these extensions of $\mathbb{F}_q$ contains all the others.

Using elementary linear algebra for the Frobenius endomorphism φ acting on $J[\ell]$ (seen as an $\mathbb{F}_\ell$-vector space), we can deduce $\chi_\ell = \chi \bmod \ell$. We first compute a basis of $J[\ell]$ by brute force and a dictionary of how all elements decompose on it. Then, the action of φ on the basis elements can be computed and the result is a matrix whose characteristic polynomial is χ_ℓ. All of this fits in the $O_g((\log q)^{O(g)})$ complexity bound. The loop is repeated $O_g(\log q)$ times, and this additional factor does not affect the overall complexity. □

5.2 Computing geometric resolutions

5.2.1 Main complexity result

The following proposition is a cornerstone of our complexity result for computing the ℓ-torsion of the Jacobian of a hyperelliptic curve. The statement and its proof combine three main ingredients: (1) the geometric resolution algorithm [64] and its version for finite fields [25], which are methods for solving polynomial systems detailed in Section 2.4 whose complexity depends mainly on geometric degrees; (2) the multi-homogeneous Bézout bound presented in Section 2.4.1 which allows us to control the geometric degrees by separating the variables in our modelling in two blocks, where the block supporting most of the degrees has small cardinality; (3) a variant of Bertini's theorem to process our polynomial system into a reduced regular sequence which is a valid input for the geometric resolution algorithm.

As we shall see in the next sections, our polynomial system modelling the ℓ-torsion will have two blocks of variables. The first block occurs with large degree $\ell^{O(1)}$ but it has a very small cardinality in $O(g)$. The second block has a larger cardinality, but the degrees of the equations with respect to this block do not depend on ℓ, but only on g. Taking this bi-homogeneous structure into account is crucial to reach our claimed complexity bound. The following proposition provides a bound on the complexity of solving polynomial systems having this structure, and the remainder of this section is dedicated to its proof. This section is devoted to describing

tools that we will use to estimate the complexity of computing a convenient representation of the ℓ-torsion of the Jacobian of hyperelliptic curves.

Let us recall the notation of Section 2.4.1: if $f \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]$, then we let $\deg_x(f)$ (resp. $\deg_y(f)$) denote the degree of $f(X_1, \ldots, X_{n_x}, y_1, \ldots, y_{n_y}) \in \overline{\mathbb{F}_q}[X_1, \ldots, X_{n_x}]$ (resp. $f(x_1, \ldots, x_{n_x}, Y_1, \ldots, Y_{n_y}) \in \overline{\mathbb{F}_q}[Y_1, \ldots, Y_{n_y}]$), where $y_1, \ldots, y_{n_y}$ (resp. $x_1, \ldots, x_{n_x}$) are generic values in $\overline{\mathbb{F}_q}$.

Proposition 5.3. *There exists a probabilistic Turing machine* $\mathbf{T}$ *which takes as input polynomial systems with coefficients in a finite field* $\mathbb{F}_q$ *and which satisfies the following property. For any function* $h : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$*, for any positive number* $C > 0$ *and for any* $\varepsilon > 0$*, there exists a function* $\nu : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ *and a positive number* $D > 0$ *such that for all positive integers* $g, \ell, n_x, n_y, d_x, d_y, m > 0$ *such that* $n_x < C\, g$*,* $n_y < h(g)$*,* $d_x < h(g)\, \ell^C$*,* $d_y < h(g)$*,* $m < h(g)$*, for any prime power* q *such that the prime number* p *dividing* q *satisfies* $2^{n_x+n_y} d_x^{n_x}\, d_y^{n_y} < p$*, and for any polynomial system* $f_1, \ldots, f_m \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]$ *such that*

- *for all* $i \in [1, m]$*,* $\deg_x(f_i) \leq d_x$ *and* $\deg_y(f_i) \leq d_y$*,*
- *the ideal* $I = \langle f_1, \ldots, f_m \rangle$ *has dimension* 0 *and is radical,*

the Turing machine $\mathbf{T}$ *with input* $f_1, \ldots, f_m$ *returns an* $\mathbb{F}_{q^{\lceil \nu(g) \log \ell \rceil}}$*-geometric resolution of the variety* $\{\mathbf{x} \in \overline{\mathbb{F}_q} \mid f_1(\mathbf{x}) = \cdots = f_m(\mathbf{x}) = 0\}$ *with probability at least* $5/6$*, using space and time bounded above by* $\nu(g)\, \ell^{D\, g}\, (\log q)^{2+\varepsilon}$*.*

Proof. Postponed to Subsection 5.2.3. □

5.2.2 Input preparation

Since the geometric resolution requires its input to be a reduced regular sequence, we first need to ensure that we can construct such a sequence from our input system. A classical way to achieve this is to replace the input system by a generic linear combination of the polynomials. If the ideal generated by the input system is 0-dimensional and radical, then a variant of Bertini's theorem ensures that the obtained sequence is regular and reduced in the sense of Definitions 2.26 and 2.47.

Proposition 5.4. *[132, Thm. A.8.7] Let* $(f_1, \ldots, f_m) \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]^m$ *be polynomials such that the ideal* $I = \langle f_1, \ldots, f_m \rangle$ *has dimension* 0 *and is radical. Let* d_x, d_y *be two integers such that* $\deg_x(f_i) \leq d_x$*,* $\deg_y(f_i) \leq d_y$ *for all* $i \in [1, m]$*. Let* p *be the characteristic of* $\mathbb{F}_q$*, and assume that* $2^{n_x+n_y} d_x^{n_x}\, d_y^{n_y} < p$*. For* M *an* $(n_x + n_y) \times m$ *matrix with entries in* $\overline{\mathbb{F}_q}$*, let* $(f_1^{(M)}, \ldots, f_{n_x+n_y}^{(M)}) \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]^{n_x+n_y}$ *be defined as*

$$\begin{bmatrix} f_1^{(M)} \\ f_2^{(M)} \\ \vdots \\ f_{n_x+n_y}^{(M)} \end{bmatrix} = M \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}.$$

Then there exists a nonempty open subset $\mathcal{O} \subset \overline{\mathbb{F}_q}^{(n_x+n_y)\times m}$ *of the space of* $(n_x + n_y) \times m$ *matrices such that for any* $M \in \mathcal{O}$*, for any* $s \in [1, n_x + n_y]$*, and at any point* $(\mathbf{x}, \mathbf{y}) \in \overline{\mathbb{F}_q}^{n_x+n_y}$ *such that* $f_1^{(M)}(\mathbf{x}, \mathbf{y}) = \cdots = f_s^{(M)}(\mathbf{x}, \mathbf{y}) = 0$*, the derivatives* $Df_1^{(M)}(\mathbf{x}, \mathbf{y}), \ldots, Df_s^{(M)}(\mathbf{x}, \mathbf{y})$ *are linearly independent over* $\overline{\mathbb{F}_q}$*. In particular, for any* $M \in \mathcal{O}$*, the sequence* $(f_1^{(M)}, \ldots, f_{n_x+n_y}^{(M)})$ *is reduced and regular.*

Proof. This is a reformulation of [132, Thm. A.8.7] in the case of finite fields. In [132, Thm. A.8.7], this result is stated over the field $\mathbb{C}$, but this statement holds over any field k, provided that an extra separability assumption is satisfied. More precisely, set $n = n_x + n_y$ and let $V_s \subset \overline{k}^n \times \overline{k}^{n\,m}$ be the variety of pairs $((\mathbf{x}, \mathbf{y}), M)$ such that $f_1^{(M)}(\mathbf{x}, \mathbf{y}) = \cdots = f_s^{(M)}(\mathbf{x}, \mathbf{y}) = 0$. In this setting, the extra condition that is required for the proposition to hold is that the projection π of V_s to $\overline{k}^{n\,m}$ must be separable for all $s \in [1, n]$ (this is always true in characteristic 0). We refer to [84, Thm. 4.2] for more details on this separability argument. In our setting, the degree of a generic fiber of π is bounded by $2^n d_x^{n_x} d_y^{n_y} < p$ using the multi-homogeneous Bézout bound (see e.g. Proposition 2.48) and hence the separability condition is satisfied. □

Since we are looking at polynomial systems over finite fields, we must estimate the size of the extension of the base field that is required to find with sufficiently large probability a matrix M such that $f_1^{(M)}, \ldots, f_{n_x+n_y}^{(M)}$ is reduced and regular.

Lemma 5.5. *Let $(f_1, \ldots, f_m) \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]^m$ be polynomials satisfying the assumptions of Proposition 5.4 and such that their total degree is bounded above by $d \in \mathbb{Z}_{\geq 0}$. Set $n = n_x + n_y$ and*

$$e = \left\lceil (2n+1)\log_q(d+1) + \log_q(11) \right\rceil.$$

If M is an $n \times m$ matrix with entries in $\mathbb{F}_{q^e}$ picked uniformly at random, then the probability that $(f_1^{(M)}, \ldots, f_n^{(M)})$ is a reduced regular sequence is bounded below by $10/11$.

Proof. Let Λ denote an $n \times m$ matrix with indeterminate entries

$$\Lambda = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1m} \\ \vdots & \vdots & \vdots \\ \lambda_{n1} & \cdots & \lambda_{nm} \end{bmatrix}$$

and let $F_1(\Lambda, X, Y), \ldots, F_n(\Lambda, X, Y) \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}, \lambda_{11}, \ldots, \lambda_{nm}]$ be the polynomials defined as

$$\begin{bmatrix} F_1(\Lambda, X, Y) \\ \vdots \\ F_n(\Lambda, X, Y) \end{bmatrix} = \Lambda \cdot \begin{bmatrix} f_1(X, Y) \\ \vdots \\ f_m(X, Y) \end{bmatrix}.$$

For $s \in [1, n]$, we consider the $s \times m$ matrix $\Lambda^{(s)}$ obtained by truncating Λ to its s first rows, a new set of variables $\{\mu_1, \ldots, \mu_{s-1}\}$ and the following polynomial system:

$$F_1(\Lambda^{(s)}, X, Y) = \cdots = F_s(\Lambda^{(s)}, X, Y) = 0$$

$$\begin{bmatrix} \mu_1 & \cdots & \mu_{s-1} & 1 \end{bmatrix} \cdot \begin{bmatrix} \dfrac{\partial F_1}{\partial X_1} & \cdots & \dfrac{\partial F_1}{\partial X_{n_x}} & \dfrac{\partial F_1}{\partial Y_1} & \cdots & \dfrac{\partial F_1}{\partial Y_{n_y}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial F_s}{\partial X_1} & \cdots & \dfrac{\partial F_s}{\partial X_{n_x}} & \dfrac{\partial F_s}{\partial Y_1} & \cdots & \dfrac{\partial F_s}{\partial Y_{n_y}} \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}$$

This is a system of $n + s$ polynomials of degree bounded above by $d + 1$ in $n + s - 1 + ms$ variables. By Bézout inequality (see e.g. [72, Thm. 1]), this system defines a variety V_s which is either empty, or its degree is at most $(d+1)^{n+s}$. We remark that if V_s is not empty, then it has dimension at least $ms - 1$ since its vanishing ideal is generated by $n + s$ elements. The Zariski closure of its projection W_s to the space $\overline{\mathbb{F}_q}^{sm}$ of matrices $\Lambda^{(s)}$ is either empty, the whole

space or a proper sub-variety. By Proposition 5.4, it must be empty or a proper sub-variety. Next, we remark that the degree of the image of a variety by a linear projection cannot increase. Therefore, the sum of the degrees of the irreducible components of W_s is also bounded by $(d+1)^{n+s}$ if $W_s \neq \emptyset$. In what follows, we let $h_s(\lambda_{11}, \ldots, \lambda_{sm})$ denote a polynomial vanishing on W_s of degree bounded by $(d+1)^{n+s}$ (we set $h_s(\lambda_{11}, \ldots, \lambda_{sm}) = 1$ if $W_s = \emptyset$).

The Schwartz-Zippel Lemma implies that the cardinality of the set

$$E = \left\{ \begin{bmatrix} M_{11} & \cdots & M_{1m} \\ \vdots & \vdots & \vdots \\ M_{n1} & \cdots & M_{nm} \end{bmatrix} \in \mathbb{F}_{q^e}^{nm} \mid h_1(M_{11}, \ldots, M_{1m}) \cdots h_n(M_{11}, \ldots, M_{nm}) \neq 0 \right\}$$

is bounded above by $q^e/11$, for the value of e given in the statement.

The proof is concluded by noticing that for any $M \in E$, for any $s \in [1, n]$, and for any $(\mathbf{x}, \mathbf{y}) \in \overline{\mathbb{F}_q}^n$ such that $f_1^{(M)}(\mathbf{x}, \mathbf{y}) = \cdots = f_s^{(M)}(\mathbf{x}, \mathbf{y}) = 0$ the derivatives $Df_1^M(\mathbf{x}, \mathbf{y}), \ldots, Df_s^{(M)}(\mathbf{x}, \mathbf{y})$ span the normal space at $(\mathbf{x}, \mathbf{y})$ to the variety associated with $\langle f_1^{(M)}, \ldots, f_s^{(M)} \rangle$. Hence, $f_1^{(M)}, \ldots, f_n^{(M)}$ is a reduced regular sequence. □

Once we have a reduced regular sequence, we can use Theorem 2.53 to solve the system. We note that in [25] there is a general assumption that for all $s \in [1, n]$ the intermediate ideals $\langle f_1^{(M)}, \ldots, f_s^{(M)} \rangle$ define absolutely irreducible varieties. However, the proof of Theorem 2.53 does not require this assumption (this assumption is only required in algorithms for finding a rational point in [25, Section 6]). To apply the theorem, we need our input to be represented by division-free straight line programms (DFSLP) as in Definition 2.49 and we can bound the size of such SLP using Lemma 2.50.

The last ingredient to derive Proposition 5.3 from Theorem 2.53 is an upper bound on $\delta = \max_i \deg\langle f_1, \ldots, f_i \rangle$ which was given in Proposition 2.48. Let us now complete the proof of Proposition 5.3.

5.2.3 Proof of the main complexity result

Proof of Proposition 5.3. Set $n = n_x + n_y$. First, we note that if $f_1, \ldots, f_m \in \mathbb{F}_q[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]$ is represented by a straight-line program over $\mathbb{F}_q$ with space $\mathcal{S}$ and time $\mathcal{T}$, then for any $e \in \mathbb{Z}_{\geq 0}$ and any $m \times n$ matrix M with entries in $\mathbb{F}_{q^e}$, the sequence $f_1^{(M)}, \ldots, f_n^{(M)} \in \mathbb{F}_{q^e}[X_1, \ldots, X_{n_x}, Y_1, \ldots, Y_{n_y}]$ can be represented by a straight-line program over $\mathbb{F}_{q^e}$ with space $\mathcal{S}'$ and time $\mathcal{T}'$, where $S' = O(\mathcal{S})$ and $\mathcal{T}' = O(\mathcal{T} + m\,n)$. We consider the probabilistic Turing machine which performs the following steps:

1. It chooses an $m \times n$ matrix uniformly at random with entries in $\mathbb{F}_{q^e}$, with

$$e = \max\left(\left\lceil (2n+1)\log_q(d+1) + \log_q(11) \right\rceil, \left\lceil \log_q(60\,n^4\,d\,\delta) \right\rceil \right),$$

   where $d = d_x + d_y = (\ell^C + 1)\,h(g)$, $n = n_x + n_y = Cg + h(g)$, $\delta = 2^n d_x^{n_x} d_y^{n_y} = (2h(g))^{Cg+h(g)} \ell^{C^2 g}$. Using the inequalities $n_x < C\,g, n_y < h(g), d_x < h(g)\ell^C, d_y < h(g)$, we get that $e = O_g(\log_q \ell)$;

2. It constructs the straight-line program representing $f_1^{(M)}, \ldots, f_n^{(M)}$ with space $\mathcal{S}' = O(\mathcal{S})$ and time $\mathcal{T}' = O(\mathcal{T} + m\,n)$;

3. It applies the probabilistic Turing machine from Theorem 2.53 to compute a geometric resolution of the algebraic set defined by $f_1^{(M)}(X) = \cdots = f_n^{(M)}(X) = 0$; By Theorem 2.53, it returns a geometric resolution $((\ell_1, \ldots, \ell_n), q(T), (q_1(T), \ldots, q_n(T)))$ with probability 11/12, provided that $f_1^{(M)}(X), \ldots, f_n^{(M)}(X)$ is a reduced regular sequence;

4. It computes $\lambda(T) = \mathsf{GCD}(q(T), f_1(q_1(T), \ldots, q_n(T)), \ldots, f_m(q_1(T), \ldots, q_n(T)))$;

5. It computes $\nu_1(T) = q_1(T) \bmod \lambda(T), \ldots, \nu_n(T) = q_n(T) \bmod \lambda(T)$ and returns the geometric resolution $((\ell_1, \ldots, \ell_n), \lambda(T), (\nu_1(T), \ldots, \nu_n(T)))$.

We start by showing that the output of this algorithm is indeed a geometric resolution of the algebraic set $V = \{\mathbf{x} \in \overline{\mathbb{F}_q}^n \mid f_1(\mathbf{x}) = \cdots = f_m(\mathbf{x}) = 0\}$, assuming that the probabilistic algorithm in Step 3 returns the correct result and that $(f_1^{(M)}, \ldots, f_n^{(M)})$ is a reduced regular sequence. Let W be the algebraic set $\{\mathbf{x} \in \overline{\mathbb{F}_q}^n \mid f_1^{(M)}(\mathbf{x}) = \cdots = f_m^{(M)}(\mathbf{x}) = 0\}$. Since $\langle f_1^{(M)}, \ldots, f_n^{(M)}\rangle \subset \langle f_1, \ldots, f_m\rangle$, we have $V \subset W$. By construction, the algebraic set defined by the geometric resolution $((\ell_1, \ldots, \ell_n), \lambda(T), (\nu_1(T), \ldots, \nu_n(T)))$ is precisely the subset of W where all polynomials $f_1, \ldots, f_m$ simultaneously vanish.

It remains to prove that this Turing machine runs within the desired complexity. Steps 1 and 2 require negligible time. Step 3 is done within space $O((\mathcal{S}' + n + d)\delta^2 \log(q^e\delta))$ and time $\widetilde{O}((n\mathcal{T}' + n^5)\delta(d\delta + \log(q^e\delta))\log(q^e\delta))$ (Theorem 2.53), provided that δ is an upper bound on the degrees of the intermediate ideals. Step 4 is done within space and time bounded by $\widetilde{O}(\delta\, e \log q(\mathcal{T} + m))$ by evaluating the SLP modulo $q(T)$ (whose degree is bounded by δ) and then by computing m GCD using a quasi-linear algorithm. Finally, Step 5 can be done within time and space $\widetilde{O}(\delta\, e \log q)$.

Then, Proposition 2.48 shows that δ is an upper bound on the degrees of the intermediate ideals. Using the facts that $\binom{n_x+d_x}{d_x} \leq (n_x+d_x)^{n_x} = O_g(\ell^{C^2 g})$ and $\binom{n_y+d_y}{d_y} \leq (n_y+d_y)^{n_y} = O_g(1)$, Lemma 2.50 provides bounds on $\mathcal{S}$ and $\mathcal{T}$. Summing these complexities leads to the claimed complexity estimate. Finally, the probability of success is bounded below by the probability that the sequence $f_1^{(M)}, \ldots, f_n^{(M)}$ is reduced and regular (Lemma 5.5) multiplied by the probability of success of the probabilistic Turing machine in Theorem 2.53, namely $10/11 \cdot 11/12 = 5/6$. □

5.3 Computing generic ℓ-torsion points

Let $\mathcal{C}$ be a hyperelliptic curve of genus g over $\mathbb{F}_q$ with Weierstrass form $Y^2 = f(X)$ (f monic, squarefree, and $\deg(f) = 2g + 1$) and let J be its Jacobian. Let $\ell > g$ be a prime not dividing q. In this section, we define a notion of genericity for ℓ-torsion elements in J and we show that a geometric resolution for the variety they form can be computed efficiently using the tools described in Section 5.2. Our starting point is the modelling of the ℓ-torsion sketched by Cantor in the point (5) of Section 9 of [28]. This section and the next one that deals with the non-generic cases rely heavily on the Mumford representation detailed in Theorem 1.29.

In what follows, we often also call Mumford representation a pair of polynomials where u is not monic. In that case, unicity of the representation is no longer guaranteed, but there is no ambiguity on the element of J represented this way.

In genus 1, the ℓ-torsion points are the points whose abscissae are the roots of the ℓ-division polynomial, which has degree $O(\ell^2)$. For higher genera, Cantor [28] described analogous polynomials δ_ℓ and ε_ℓ. By Theorem 4.1, for (x, y) the generic point of the curve and $\ell > g$, we

have

$$\ell\left((x,y)-\infty\right)=\left\langle \delta_\ell\left(\frac{x-X}{4y^2}\right),\varepsilon_\ell\left(\frac{x-X}{4y^2}\right)\right\rangle.$$

Let us now restate Theorem 4.13, which is proven in Section 4.2 of Chapter 4 :

The polynomial $\delta_\ell(X)$ has degree g and its coefficients are polynomials in $\mathbb{F}_q[x]$ of degree bounded by $\frac{1}{3}g\ell^3+O_g(\ell^2)$. The polynomial $\varepsilon_\ell(X)/y$ has degree less than g and its coefficients are rational fractions in $\mathbb{F}_q(x)$. The degrees of the numerators and denominators of these coefficients are bounded by $\frac{2}{3}g\ell^3+O_g(\ell^2)$. Furthermore, any root of a denominator is also a root of the leading coefficient of $\delta_\ell(X)$.

Remark that this result is also proven for any non-prime $\ell>g$, it will be used in Section 5.4 where we handle non-generic situations. However, we will also need to define analogues of these polynomials to describe ℓP when P is not generic. This is done in Definition 5.9 and we also remark later on that the previous degree bounds still apply to non generic division polynomials.

Later on, we will need explicit names for these coefficients of δ_ℓ and ε_ℓ, so we define the univariate polynomials d_i and e_i (the notation does not show the dependence on ℓ for simplicity) such that, after clearing denominators we have:

$$\delta_\ell\left(\frac{x-X}{4y^2}\right)=\sum_{i=0}^{g}d_i(x)X^i,\quad\text{and}\quad\varepsilon_\ell\left(\frac{x-X}{4y^2}\right)=y\sum_{i=0}^{g-1}\frac{e_i(x)}{e_g(x)}X^i.$$

Definition 5.6. *In what follows, we shall say that an element of J is ℓ-generic if it has weight g and the corresponding reduced divisor $\sum_{i=1}^{g}(P_i-\infty)$ satisfies the following two properties:*

- *For any i, the u-coordinate of the divisor $\ell(P_i-\infty)$ in Mumford form has degree g;*
- *For any $i\neq j$, the u-coordinates of the divisors $\ell(P_i-\infty)$ and $\ell(P_j-\infty)$ are coprime.*

This implies that the P_i are distinct, and that if an affine point P occurs in the support of a $\ell(P_i-\infty)$ then neither P nor $-P$ appears in the support of another $\ell(P_j-\infty)$.

Proposition 5.7. *For any $\varepsilon>0$, there is a constant D such that for all prime $\ell>g$ coprime to the base field characteristic, there is a Monte Carlo algorithm which computes an $\mathbb{F}_{q^e}$-geometric resolution of the sub-variety of $J[\ell]$ consisting of ℓ-generic ℓ-torsion elements, where $e=O_g(\log\ell)$. The time and space complexities of this algorithm are bounded by $O_g(\ell^{Dg}(\log q)^{2+\varepsilon})$ and it returns the correct result with probability at least $5/6$.*

Proof. Let $D=\sum_{i=1}^{g}(P_i-\infty)$ be an ℓ-generic divisor in J. We shall consider a system equivalent to $\ell D=0$ but let us first introduce some notation. For each point $P_i=(x_i,y_i)$ in the support of D, we denote by $\langle u_i,v_i\rangle$ the Mumford form of $\ell(P_i-\infty)$ and by $(\alpha_{ij},\beta_{ij})_{1\leq j\leq g}$ the coordinates of the g points in its support counted with multiplicities, which means that for any i the g roots of u_i are exactly the α_{ij}, and that for any j, $\beta_{ij}=v_i(\alpha_{ij})$. Note that using the previous notation, $u_i(X)=\delta_\ell\left(\frac{x_i-X}{4y_i^2}\right)$ and $v_i(X)=\varepsilon_\ell\left(\frac{x_i-X}{4y_i^2}\right)$.

We have $\ell D=0$ if and only if the sum of the divisors $\sum_{i=1}^{g}\ell(P_i-\infty)$ is a principal divisor. The only pole is at infinity, so this is equivalent to the existence of a non-zero function $\varphi\in\mathbb{F}_q(\mathcal{C})$ of the form $P(X)+YQ(X)$ with P and Q two polynomials such that the g^2 points (α_{ij},β_{ij}) are the zeros of φ, with multiplicities. Since we want φ to have g^2 affine points of intersection with the curve $\mathcal{C}$ (once again, counted with multiplicities), the polynomial $\mathrm{Res}_Y(Y^2-f,P+YQ)=P^2-fQ^2$ must have degree g^2 which yields $2\deg(P)\leq g^2$ and $2\deg(Q)\leq g^2-2g-1$. Exactly

one of those two bounds is even (it depends on the parity of g), and for this particular bound, the inequality must be an equality, otherwise the degree of the resultant would not be g^2. Since the function φ is defined up to a multiplicative constant, we can normalize it so that the polynomial $P^2 + fQ^2$ is monic, which is equivalent to enforce that either P or Q is monic depending on the parity of g.

For a fixed $i \in [1, g]$, requiring the $(\alpha_{ij}, \beta_{ij})$ to be zeros of φ amounts to asking for the α_{ij} to be roots of $P(X) + Q(X)v_i(X)$, with multiplicities. Since the α_{ij} are by definition the roots of the u_i, $\ell D = 0$ is equivalent to g congruence relations $P + Qv_i \equiv 0 \bmod u_i$ which we can rephrase using Cantor's polynomials:

$$P(X) + \varepsilon_\ell \left(\frac{x_i - X}{4y_i^2} \right) Q(X) \equiv 0 \bmod \delta_\ell \left(\frac{x_i - X}{4y_i^2} \right). \tag{5.1}$$

Thus, for any ℓ-generic divisor, $\ell D = 0$ is equivalent to the existence of P and Q satisfying the above g congruence relations.

The variables are the coefficients of P and Q, as well as the x_i and y_i. With the degree conditions and the normalization, we have $g^2 - g$ variables coming from P and Q. Adding the $2g$ variables x_i and y_i, we get a total of $g^2 + g$ variables. Each one of the g congruence relations (5.1) amounts to g equations providing a total of g^2 conditions on the coefficients of P and Q. The fact that the (x_i, y_i) are points of the curve yields the g additional equations $y_i^2 = f(x_i)$. Finally, we have to enforce the ℓ-genericity of the solutions, which can be done by requiring that $\prod_i d_g(x_i) \prod_{i<j} \mathrm{Res}(u_i, u_j) \neq 0$. Therefore, we get a polynomial system with $g^2 + g$ equations in $g^2 + g$ variables, together with an inequality. We remark that in principle, the denominators $e_g(x_i)$ involved in ε_ℓ would generate additional conditions, but by Theorem 4.13 this is already covered by the condition $d_g(x_i) \neq 0$.

In order to apply Proposition 5.3, we now estimate the degrees to which the variables occur in the equations. We start with the equations coming from (5.1). Each congruence relation is obtained by reducing $P + Qv_i$, which is a polynomial of degree $O(g^2)$ in X, by u_i which is of degree g. We can do it by repeatedly replacing X^g by $-\sum_{j<g}(d_j(x_i)/d_g(x_i))X^j$, which we will have to do at most $O(g^2)$ times. Since by Theorem 4.13 the d_j have degree in $O_g(\ell^3)$ in x_i the fully reduced polynomial will have coefficients that are fractions for which the degrees of the numerators and of the denominators are at most $O_g(\ell^3)$ in the x_i variables. In these equations, the degree in the y_i variables and in the variables for the coefficients of P and Q is 1. The degrees in x_i and y_i in the curve equations are $2g + 1$ and 2 respectively.

It remains to study the degree of the inequality. Each resultant is the determinant of a $2g \times 2g$ Sylvester matrix whose coefficients are the d_i, which have degrees bounded by $O_g(\ell^3)$. Since for any i there are exactly g resultants involving x_i in the product, the degree of this inequality in any x_i is in $O_g(\ell^3)$, and it does not involve the other variables. In order to be able to use Proposition 5.3, we must model this inequality by an equation, which is done classically by introducing a new variable T and by using the equation $T \cdot \prod_i d_g(x_i) \prod_{i<j} \mathrm{Res}(u_i, u_j) = 1$.

To conclude, we have a polynomial system with two blocks of variables: the $2g$ variables x_i and y_i and the $g^2 - g$ variables coming from the coefficients of P and Q. The degree of the equations in the first block of variables grows cubically in ℓ, while the degree in the other block of variables depends only on g. The system therefore verifies the conditions of Proposition 5.3 and the complexity follows, provided that we can show that the system is 0-dimensional and radical.

Let us consider the sub-variety $S \subset J[\ell]$ consisting of ℓ-generic ℓ-torsion elements, and I the corresponding ideal. More precisely, we see I as the ideal of a sub-scheme of the ℓ-torsion

scheme, which is the kernel of a finite and étale map because ℓ is coprime to the characteristic. Therefore I is 0-dimensional and radical. Since all the elements in S have the same weight g we can use the Mumford coordinates $\langle u(X), v(X)\rangle$ with $\deg u = g$ and $\deg v < g-1$ as a local system of coordinates to represent them. But the polynomial system that we have built is with the (x_i, y_i) coordinates, that is, it generates the ideal I^{unsym} obtained by adjoining to the equations defining I the $2g$ equations coming from $u(X) = \prod(X - x_i)$ and $y_i = v(x_i)$. Then we have $\deg I^{\text{unsym}} = g! \deg I$. By the ℓ-genericity condition, all the fibers in the variety have exactly $g!$ distinct points corresponding to permuting the (x_i, y_i) which are all distinct. Therefore the radicality of I implies the radicality of I^{unsym} and we can apply Proposition 5.3 to our polynomial system. □

We emphasize that, although the algorithm in Proposition 5.7 is Monte Carlo, we expect that it returns a correct and verifiable result in most of the cases. Indeed, if all the $\ell^{2g} - 1$ nonzero ℓ-torsion elements are ℓ-generic (which is the situation that we expect to happen in most of the cases) and if the algorithm returns the correct result, then we can check that these elements are indeed ℓ-torsion elements, and that we have all of them. In that favorable case, the proof of Proposition 5.2 is completed.

5.4 Non-generic cases

For most of the curves, we expect that for all the primes ℓ considered in Algorithm 6 the set $J[\ell]$ contains only ℓ-generic elements (apart from 0), so that the result of the previous section is sufficient. If this is not the case, then it is very likely that the orbit under the Frobenius endomorphism of the ℓ-torsion elements computed contains an $\mathbb{F}_\ell$-basis of $J[\ell]$, so that we can easily recover the missing elements using the group law or the Frobenius. Still, unless we could prove otherwise, we can not exclude the case where the set of ℓ-generic ℓ-torsion elements generate a proper subgroup of $J[\ell]$ which is stable under the action of φ. In that unlikely case, we would maybe not be able to deduce χ_ℓ. An option is then to skip this unlucky ℓ and proceed with the algorithm; this would only marginally increase the largest considered ℓ. But then, we would be left to prove that the number of unlucky ℓ's is small enough, which seems as hard.

Our only remaining option is to perform a tedious, systematic study of all the non-generic cases and to show that they can all be modelled by polynomial systems that can be solved within the target complexity. The number of these systems must also be bounded independently of ℓ, so that with our setting where g is fixed and q grows to infinity the global complexity remains the same. All this is the purpose of Subsection 5.4.2. As a warm-up, we will first describe some simple degeneracy cases and, informally, how to deal with them. Since several causes of non-genericity may simultaneously appear, we then describe a data structure to encode all the possible non-generic cases. Then, we detail how to build a polynomial system modelling each of these cases. Note that all these systems will have more equations ($O(g^4)$, see Table 5.2) than variables ($O(g^2)$, see Table 5.1), which is no wonder since we expect them to have no solution in general.

Lastly, we point out that Subsections 5.4.1 and 5.4.2 can easily be skipped at first reading as it is only devoted to proving the main theorem of the chapter and will not be used in other chapters or sections of this thesis.

5.4.1 Simple degeneracies

Case 1: Low weight ℓ-torsion elements. In order to compute the ℓ-torsion elements that satisfy all the conditions of ℓ-genericity except that their weight is less than g, we can proceed as in the proof of Proposition 5.7 with the following modifications. This time, $D = \sum_{i=1}^{w}(P_i - \infty)$, and the only difference is that there are w points instead of g. Following the same method, we search φ of the form $P(X) + YQ(X)$ such that the points in the reduced divisor $\ell(P_i - \infty)$ are exactly the zeros of φ. We now want φ to have gw points of intersection with $\mathcal{C}$ instead of g^2, and we similarly deduce $2\deg(P) \leq gw$ and $2\deg(Q) \leq gw - 2g - 1$. By similar parity considerations we deduce that exactly one of these bounds is even, and the corresponding polynomial will be made monic to normalize the function. The number of variables from P and Q is thus $gw - g$, and after adding the $2w$ variables x_i and y_i, we have a total of $(g+1)w + w - g$ variables. As for the number of equations, the number of congruence relations is now w but the relations themselves remain unchanged, and we get a total of $(g+1)w$ equations after adding the w equations $y_i^2 = f(x_i)$. Since we keep the degrees unchanged but reduce the number of variables, the complexity bounds are still valid in this case.

Case 2: Multiple points in the ℓ-torsion divisor. It may happen that the reduced forms of ℓ-torsion divisors contain multiple points. In that case, the u-coordinate in the Mumford representation of such a point is not squarefree. Although the modelling by the polynomial system described in Section 5.3 is still faithful, such multiple points will induce multiplicities since what we actually compute is the variety describing the points in the reduced divisor. Therefore, the ideal generated by the polynomial system is not radical in this case. We use the following workaround: For $\lambda = (\lambda_1, \ldots, \lambda_k)$ a partition of w, we write a polynomial system generating a radical ideal whose solutions represent the reduced divisors of the form $D = \lambda_1 P_1 + \cdots + \lambda_k P_k - w\,\infty$. To build this polynomial system, we do as if we were looking for elements of weight k, but instead of multiplying P_i by ℓ, we multiply it by $\lambda_i \ell$, using Cantor's polynomials $\delta_{\lambda_i \ell}$ and $\varepsilon_{\lambda_i \ell}$. This system has the same number of variables and equations as if we were looking for elements of weight k. Since λ_i is bounded above by g, the degrees of the equations are multiplied by a quantity which depends only on g but not on ℓ. Consequently, the complexity bounds are still valid in this case. To avoid multiplicity problems that could arise from subpartitions of λ, we add the inequalities $x_i \neq x_j$ for $i \neq j$, where x_i is the x-coordinate of P_i. Again, this does not change our complexity estimate.

Case 3: Low weight after multiplication by ℓ. We study here the case where the ℓ-genericity property that is not verified is that the $\ell(P_i - \infty)$ are of weight g, all the others being satisfied. We denote by $w_i \leq g$ the weight of $\ell(P_i - \infty)$. Then each u_i will have degree w_i, so that each congruence relation (5.1) yields only w_i equations instead of g. In Cantor's article (on top of page 141 in [28]), it is stated that $\ell \cdot (P_i - \infty)$ is of weight w_i if and only if for any k such that $w_i < k \leq g$ we have $\psi_{\ell - k + w_i + 1}(x_i) = 0$ and $\psi_{\ell - g + w_i}(x_i) \neq 0$, where the polynomials ψ_i are efficiently computable and of degrees bounded by $O_g(\ell^2)$. Therefore the total number of equations is unchanged. Since the function φ will have to vanish at $\sum_i w_i$ points instead of g^2, we also reduce the degree of P and Q accordingly. The number of variables from P and Q thus becomes $\sum_i w_i - g$ which is smaller than in the generic case, while the number of equations remains the same, and their degrees are also smaller. Thus we can still describe this non-generic situation with systems that can be handled within the same complexity bounds.

Case 4: Non semi-reduced principal divisor. We now consider the case where the ℓ-genericity property fails due to the presence of a point of abscissa ξ which appears with positive multiplicity ν_i in an $\ell(P_i - \infty)$ and with a negative multiplicity $-\nu_j$ in another $\ell(P_j - \infty)$. Let $\nu = \min(\nu_i, \nu_j)$. This event implies that $(X - \xi)^\nu$ divides both P and Q so that we can write $\varphi(X, Y) = (X - \xi)^\nu(\widetilde{P}(X) + Y\widetilde{Q}(X))$, with $\widetilde{P}$ coprime to $\widetilde{Q}$. The number of variables coming from φ is reduced compared to the generic case: we add one (the variable ξ), but the number of coefficients in $\widetilde{P}$ is reduced by ν compared to P, and the same is true for $\widetilde{Q}$ and Q. To write the conditions on φ, we write the congruences exactly like in the generic case and we add conditions to ensure that the multiplicities are respected. Namely, u_i, u_j and $v_i + v_j$ must all be divisible by $(X - \xi)^\nu$, which adds $3\nu \leq 3g$ equations. The degree in ξ in these equations is bounded by g^2. Since this does not depend on ℓ, the complexity result is maintained. The general study will cover the case where there are several ξ's at which the semi-reduction genericity assumption fails. Also, there is no reason why such a root ξ should occur in only two of the $\ell(P_i - \infty)$'s. Such a situation will be also taken into account in Section 5.4.2.

Case 5: Multiplicity in ℓD. The last situation that could lead to not satisfying ℓ-genericity is when the same point is shared within different $\ell(P_i - \infty)$, which causes some trouble as the congruence relations of the generic case will not be able to handle the subsequent multiplicity. Note that if the multiplicity occurs only within a single $\ell(P_i - \infty)$ this is already dealt within the generic case. One can view our method as using the Chinese remainder theorem on the modular conditions (5.1) to see that multiplicities within a single congruence is handled whereas common factors within different u_i-polynomials are an obstacle that needs special strategies. There is some similarities with the previous case that also implies a common factor between two different u_i's.

We devise the following workaround: instead of considering the congruences modulo the u_i's separately, we group them into a single congruence of the form $P + QV \equiv 0 \bmod U$, with $U = \prod_i u_i$ and V a polynomial whose coefficients shall be new variables such that $V \equiv v_i \bmod u_i$ for all i. Note that if some non semi-reduced case occurs simultaneously, U must actually be divided by the aforementioned $X - \xi$; such situations will be dealt with later, in the general study (Section 5.4.2). In order for V to encode enough information and ensure that the condition $P + QV \equiv 0 \bmod U$ enforces a function with exactly the correct principal divisor, we have to follow Mumford's representation and add the condition $U | V^2 - f$, with $\deg V < \deg U$. Together with the other conditions on U and V, we then have existence and unicity (up to a constant factor): they are the result of Cantor's composition algorithm.

In order to write the polynomial system modelling this situation, some care must be taken so as to stay within the scope of Proposition 5.3. The polynomial U is of degree g^2 and its coefficients are polynomials in the x_i's of degrees bounded by $O_g(\ell^3)$. New variables are added for the coordinates of V. For each i, the condition $V \equiv v_i \bmod u_i$ is converted in $O(g)$ equations, with degrees $O_g(\ell^3)$ in x_i and 1 in the coordinates of V. The condition $U | V^2 - f$ contributes to $O(g^2)$ additional equations, each of them of degree 2 in the coordinates of V, and degree $O_g(\ell^3)$ in the coordinates x_i. And finally, the equation $P + QV \equiv 0 \bmod U$, contributes also to $O(g^2)$ equations, each of them of degree 1 in the coordinates of V, P and Q, and of degree $O_g(\ell^3)$ in the coordinates x_i. Skipping the details, we can again apply Proposition 5.3 and get the expected complexity.

5.4.2 Combining all possible degeneracies

A data structure to describe each type of non-genericity. We want to describe a family of polynomial systems that covers all the possible non-generic cases, possibly mixing all kind of problems that have been listed. We begin by grouping together non-genericity situations that can be covered by the same polynomial system.

We consider an ℓ-torsion divisor D of weight $w \leq g$ (like in case 1). Next, a partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ of w is picked to represent the multiplicity pattern in the u-coordinate of the ℓ-torsion divisor, as in case 2 so that $D = \sum_{i=1}^{k} \lambda_i (P_i - \infty)$. Then, a vector $t = (t_1, \ldots, t_k)$ is chosen, to represent the weights of the P_i after multiplication by $\lambda_i\,\ell$ as in case 3: For i in $[1, k]$, the reduced divisor $\lambda_i\,\ell(P_i - \infty)$ is of weight t_i. Then, we need to consider how many common or opposite points these divisors are in their support to take into account the cases 4 and 5. We denote by $Q_1, \ldots, Q_s$ the points in the union of the supports of all the reduced divisors $\lambda_i\,\ell(P_i - \infty)$, keeping only one point in each orbit under the hyperelliptic involution. We represent the non-genericity by a $k \times s$ matrix M such that its non-zero entries m_{ij} satisfy $m_{ij} = \mathrm{ord}_{Q_j}(\lambda_i\,\ell(P_i - \infty))$ when Q_j is in the support of $\lambda_i\,\ell(P_i - \infty)$ or $m_{ij} = -\mathrm{ord}_{Q'_j}(\lambda_i\,\ell(P_i - \infty))$ when the hyperelliptic conjugate Q'_j of Q_j is in the support. Note that this matrix, that we shall call the matrix of shared points, represents both multiplicities and non-semi-reduction. Since the row i represents what happens with points in the support of $\lambda_i\,\ell(P_i - \infty)$, which is of weight t_i, the sum of the absolute values of the entries of the row i of M is equal to t_i.

Also, by construction, in each column, there is at least one non-zero entry. An additional complication arises when one of the P_i is a ramification point, i.e. when its y-coordinate is zero, because this would cause multiplicities if care is not taken, leading to non-radicality of the polynomial system we build. Since this corresponds to $P_i - \infty$ being of order 2, the weight t_i is equal to $\lambda_i \ell \bmod 2$, namely 0 or 1. If $t_i = 0$, then the divisor $D - \lambda_i(P_i - \infty)$ is also an ℓ-torsion divisor of weight $w - \lambda_i$, so that we can reconstruct D from another polynomial system. There is however no obvious way to preclude the possibility $t_i = 1$. Therefore, we will encode the fact that P_i is a ramification point by a bit ϵ_i that can be set only in the cases where $t_i = 1$ and $\lambda_i = 1$.

A tuple $(w, \lambda = (\lambda_1, \ldots, \lambda_k), t = (t_1, \ldots, t_k), \epsilon = (\epsilon_1, \ldots, \epsilon_k), M)$ is from now on the piece of data with which we represent a non-generic situation, and a polynomial system will be associated to each tuple. Changing the order of the columns of M amounts to permuting the points Q_j. Also, changing the sign of all the entries of a column j corresponds to taking the opposite of the point Q_j. While it would not change the final complexity not to do so, it therefore makes sense to consider only normalized tuples, in the sense that the columns of M are sorted in lexicographical order, and the choice between a point Q_j and its opposite is done so that the sum of all elements in the corresponding column is nonnegative. We remark that this is not enough to guarantee that two normalized tuples do not describe similar situations. For instance, if $\lambda = (1, \ldots, 1)$ and two t_i values are equal, then permuting the two corresponding rows could lead to another normalized matrix that would describe the same situation. This is not a problem for the general algorithm: we might get the same ℓ-torsion elements from two different systems, but what is important to us is non-multiplicity (i.e. radicality of the ideal) in each individual system.

Definition 5.8. *A* normalized non-genericity tuple *is a tuple $(w, \lambda, t, \epsilon, M)$, where $1 \leq w \leq g$ is an integer, $\lambda = (\lambda_1, \ldots, \lambda_k)$ is a partition of w, t and ϵ are vectors $t = (t_1, \ldots, t_k)$ and $\epsilon = (\epsilon_1, \ldots, \epsilon_k)$ of the same length as λ with $1 \leq t_i \leq g$ and $\epsilon_i \in \{0, 1\}$, where ϵ_i can be 1 only if $t_i = 1$ and $\lambda_i = 1$, and finally M is a matrix with k rows and s columns, where $0 \leq s \leq g\,k$, and its entries are integers such that:*

- *For all $1 \leq i \leq k$, the sum of the absolute values of the entries on the row i is equal to t_i;*
- *The columns are sorted in lexicographical order;*
- *The sum of the rows of the matrix is a vector whose coordinates are nonnegative.*

From the discussion above, any ℓ-torsion element is described by (at least) one normalized non-genericity tuple. In the following we will give a polynomial system for each normalized non-genericity tuple, so that all ℓ-torsion elements described by it are modelled by this system. Furthermore, the system will have the properties required to apply Proposition 5.3, so that the complexity result will follow.

Before starting this, we discuss briefly a bound on the number of normalized non-genericity tuples. Assuming everything is always of maximal size, and not sorted, we have g choices for w, then at most g^g choices for λ and t, at most 2^g choices for ϵ, and finally at most $(g^{2g+1})^{g^2}$ choices for M, which gives $g^{O(g^3)}$. As bad as it is, such a factor that depends only on g will not hinder the final complexity estimate in $O_g\left((\log q)^{O(g)}\right)$, as explained in Section 5.1.

Non-generic division polynomials. The expression of $\lambda_i \ell \cdot (P_i - \infty)$ in Mumford representation will be the same as in the generic case when its weight t_i is equal to g and Theorem 4.13 can be applied. But when t_i is strictly less than g, the weight-g coordinate system is no longer available; this is explicitly visible by the fact that the denominator $e_g(x_i)$ of the coefficients of the v-polynomial vanishes.

Therefore we need to use a weight-t coordinate system for describing a non-generic divisor $\lambda_i \ell \cdot (P_i - \infty)$ in Mumford representation. In this paragraph, in order to keep simple notation, we will work with $\ell(P_i - \infty)$, keeping in mind that we do not impose any condition on ℓ, so that we can later replace ℓ by $\lambda_i \ell$.

We consider, for $1 \leq t < g$, the set $V_{\ell,t}$ of points of the curve which are mapped to a weight-t divisor after multiplication by ℓ:

$$V_{\ell,t} = \{(x, y) \in \mathcal{C} \mid \ell \cdot ((x, y) - \infty) \text{ is of weight } t\}.$$

This is a (possibly empty) variety of dimension 0 that can be described with the classical (generic) division polynomials of Cantor: we define

$$\Delta_{\ell,t} = \mathsf{GCD}(\psi_\ell(x), \psi_{\ell-1}(x), \ldots, +\psi_{\ell-g+t+1}(x)),$$

so that $V_{\ell,t}$ is precisely the set of points (x, y) for which $\Delta_{\ell,t}(x) = 0$ and $\psi_{\ell-g+t}(x) \neq 0$, as stated by Cantor in [28] on page 141. The polynomial ψ_ℓ is essentially the square root of the leading coefficient of δ_ℓ. It can be computed efficiently and has degree in $O_g(\ell^2)$ by Theorem 8.17 of [28]. To avoid multiplicities, we define $\tilde{\Delta}_{\ell,t}(x)$ the square-free polynomial whose roots are exactly the roots of $\Delta_{\ell,t}(x)$ that are not roots of $\psi_{\ell-g+t}(x)$. The degree of $\tilde{\Delta}_{\ell,t}(x)$ is again bounded by $O_g(\ell^2)$. Furthermore since the points of $V_{\ell,t}$ come in pairs of conjugate points sharing the same x-value, the degree of $V_{\ell,t}$ is $2 \deg \tilde{\Delta}_{\ell,t}(x)$.

Definition 5.9. *The non-generic division polynomials $\mathfrak{u}_{\ell,t}$ and $\mathfrak{v}_{\ell,t}$ are the polynomials in X with coefficients in $\mathbb{F}_p[x, y]/(\tilde{\Delta}_{\ell,t}(x), y^2 - f(x))$ such that*

$$\ell \cdot ((x, y) - \infty) = \left\langle \mathfrak{u}_{\ell,t}(X), \mathfrak{v}_{\ell,t}(X) \right\rangle,$$

in weight-t Mumford representation: $\mathfrak{u}_{\ell,t}(X)$ is monic of degree t, $\mathfrak{v}_{\ell,t}(X)$ is of degree at most $t - 1$ and they satisfy $\mathfrak{u}_{\ell,t} \mid \mathfrak{v}_{\ell,t}^2 - f$.

Just like for the classical division polynomials, the coefficients of $\mathfrak{u}_{\ell,t}(X)$ and of $\frac{1}{y}\,\mathfrak{v}_{\ell,t}(X)$ are in $\mathbb{F}_p[x]/\tilde{\Delta}_{\ell,t}(x)$ (they do not depend on y) and we can choose representatives of them that are polynomials of degree less than $\deg \tilde{\Delta}_{\ell,t}(x)$. Hence, the bounds given in Theorem 4.13 are also valid for the non-generic division polynomials; and since there are no denominators in the coefficients of $\mathfrak{v}_{\ell,t}(X)$, the other part of Theorem 4.13 also holds trivially.

The non-generic division polynomials can be computed efficiently, once the classical division polynomials are known: the polynomial $\tilde{\Delta}_{\ell,t}(x)$ can be easily deduced, and then working in the quotient algebra yields the result in a time $\widetilde{O}_g(\ell^2)$, which is negligible compared to the other parts of the algorithm.

5.4.3 Polynomial system derived from a normalized non-genericity tuple.

We now want to write a polynomial system whose solutions are the ℓ-torsion elements following a given normalized non-genericity tuple $(w, \lambda, t, \epsilon, M)$.

First, we need variables for the coordinates of the P_i such that the ℓ-torsion element is $D = \sum_{i=1}^{k} \lambda_i (P_i - \infty)$, with $P_i \neq \pm P_j$ for all $i \neq j$. As a consequence, we introduce $2k$ variables for the coordinates (x_i, y_i) of all the points P_i. Since these points are on the curve, they satisfy $y_i^2 = f(x_i)$, however if P_i is a ramification point this can be simplified into $y_i = 0 = f(x_i)$, which avoids the multiplicities. We get a first set of equations

$$\begin{cases} y_i^2 = f(x_i) \neq 0, & \text{for all } i \text{ in } [1,k] \text{ such that } \epsilon_i = 0, \\ y_i = f(x_i) = 0, & \text{for all } i \text{ in } [1,k] \text{ such that } \epsilon_i = 1. \end{cases} \tag{Sys.1}$$

As we just discussed, we must model the fact that $P_i \neq \pm P_j$ for $i \neq j$. This is done via the following set of inequalities:

$$x_i \neq x_j, \quad \text{for all } i, j \text{ in } [1,k] \text{ such that } i \neq j. \tag{Sys.2}$$

The next step is to enforce the fact that the element $\lambda_i\, \ell(P_i - \infty)$ is of weight t_i. For the indices for which $t_i < g$, this is encoded by the equation defining $V_{\lambda_i \ell, t_i}$:

$$\begin{cases} \tilde{\Delta}_{\lambda_i \ell, t_i}(x_i) = 0, \\ d_{t_i}(x_i) \neq 0, \end{cases} \quad \text{for all } i \text{ in } [1,k] \text{ such that } t_i < g, \tag{Sys.3}$$

while for the indices for which $t_i = g$, this is encoded by the non-vanishing of the leading coefficient of the Cantor polynomial in degree $\lambda_i \ell$:

$$d_g(x_i) \neq 0, \quad \text{for all } i \text{ in } [1,k] \text{ such that } t_i = g. \tag{Sys.4}$$

We now need to model the fact that the $\lambda_i\, \ell(P_i - \infty)$ satisfy the conditions given by the matrix M. We write $\lambda_i\, \ell(P_i - \infty) = \langle u_i(X), v_i(X) \rangle$ in Mumford representation, where $u_i(X)$ and $v_i(X)$ are Cantor's classical division polynomials in degree $\lambda_i \ell$ if $t_i = g$ or the non-generic division polynomials $\mathfrak{u}_{\lambda_i \ell, t_i}$ and $\mathfrak{v}_{\lambda_i \ell, t_i}$, if $t_i < g$. In both cases, these are polynomials in X whose coefficients are polynomials in x_i and y_i. Recall that the entries of M, denoted by $(m_{ij})_{i \in [1,k], j \in [1,s]}$, are such that m_{ij} is the order of Q_j in $\lambda_i\, \ell(P_i - \infty)$ if it is positive, or the opposite of the order of Q'_j if it is negative. To this effect, we introduce s new variables ξ_j for

the abscissae of the Q_j, and the following equations enforce the multiplicities:

$$u_i^{(n)}(\xi_j) = 0, \quad \text{for all } i,j \text{ in } [1,k]\times[1,s] \text{ and for all } n \le |m_{ij}| - 1 \tag{Sys.5}$$
$$u_i^{(|m_{ij}|)}(\xi_j) \neq 0, \quad \text{for all } i,j \text{ in } [1,k]\times[1,s] \tag{Sys.6}$$
$$v_i(\xi_j) - v_{i'}(\xi_j) = 0, \quad \text{for all } i,i',j \text{ such that } m_{ij}m_{i'j} > 0 \tag{Sys.7}$$
$$v_i(\xi_j) + v_{i'}(\xi_j) = 0, \quad \text{for all } i,i',j \text{ such that } m_{ij}m_{i'j} < 0 \tag{Sys.8}$$
$$\xi_j \neq \xi_{j'}, \quad \text{for all } j \neq j'. \tag{Sys.9}$$

In Equations Sys.5 and Sys.6, the notation $u_i^{(n)}$ is for the n-th derivative of u_i. This simple way of describing multiple roots is valid because the characteristic is large enough.

The next step of the construction is to consider a semi-reduced version of the divisor $\ell D = \sum_{i=1}^{k} \lambda_i \, \ell(P_i - \infty)$. This semi-reduction process can be described directly on the matrix M: if two entries in a same column have opposite signs, a semi-reduction can occur (corresponding to subtracting the principal divisor of the function $(x - \xi_j)$), thus reducing the difference between these entries. This semi-reduction can continue until one of these two entries reaches zero. This whole process can be repeated as long as there are still columns containing entries with opposite signs. This is formalized in Algorithm 7, which takes as input a matrix M and returns a matrix $\widetilde{M}$ with the same dimensions such that if M describes all the multiplicities in a divisor, then $\widetilde{M}$ describes all the multiplicities of a semi-reduced divisor equivalent to the input divisor. More precisely, the matrix $\widetilde{M}$ satisfies the following properties: (1) In each column, all elements are nonnegative; (2) The sum of the rows of M equals the sum of the rows of $\widetilde{M}$; (3) For all i,j such that $M_{i,j}$ is nonnegative, $\widetilde{M}_{ij} \le M_{ij}$.

Data: M the $k \times s$ matrix of shared points of the system
Result: $\widetilde{M}$, the matrix after semi-reduction
$\widetilde{M} \leftarrow k \times s$ zero matrix
for j *from* 1 *to* s **do**
  $\mu_j \leftarrow \sum_{i=1}^{k} M_{ij}$
  for i *from* 1 *to* k **do**
    if $M_{ij} > 0$ **then**
      $\widetilde{M}_{ij} \leftarrow \min(M_{ij}, \mu_j)$
      $\mu_j \leftarrow \mu_j - \widetilde{M}_{ij}$
    else
      $\widetilde{M}_{ij} \leftarrow 0$
    end
  end
end
return $\widetilde{M}$

Algorithm 7: Reducing the matrix of shared points

The function φ that we will use to model the principality of the divisor ℓD will have two parts: a product of "vertical lines" corresponding to semi-reductions, and a part of the form $P(X)+YQ(X)$, where P and Q are coprime. Modelling the existence of this second part requires to introduce new entities $\widetilde{u}_i$ that are the u_i polynomials from which we remove the linear factors coming from semi-reduction as described by $\widetilde{M}$. Formally, we have the following equations,

defining $\widetilde{u}_i$:

$$u_i(X) = \widetilde{u}_i(X) \prod_{j=1}^{s} (X - \xi_j)^{|m_{ij}| - \widetilde{m}_{ij}}, \quad \text{for all } i \in [1, k]. \tag{Sys.10}$$

Indeed, by definition of the matrix M, the factor $(X - \xi_j)^{|m_{ij}|}$ divides exactly $u_i(X)$, and the factor $(X - \xi_j)^{\widetilde{m}_{ij}}$ divides exactly $\widetilde{u}_i(X)$. In order to express these conditions efficiently in the polynomial system, we introduce new variables for the coefficients of the $\widetilde{u}_i$ polynomials.

Since we are now dealing with a semi-reduced divisor, we can consider its Mumford representation, *i.e.* two polynomials U and V with the following properties:

$$U = \prod_{i=1}^{k} \widetilde{u}_i, \quad U | V^2 - f, \tag{Sys.11}$$

$$V \equiv v_i \bmod \widetilde{u}_i, \quad \text{for all } i \in [1, k]. \tag{Sys.12}$$

The expression of U is simple enough, so we do not have to introduce new variables for its coefficients. However, this will be necessary for the coefficients of the V polynomial. Finally, in order to impose that the semi-reduced part of φ has exactly the zeros described by this divisor, we have the equation

$$P + QV \equiv 0 \bmod U, \tag{Sys.13}$$

which is expressed with new variables for the coefficients of P and Q.

In Table 5.1, we summarize all the variables used in the polynomial system and count them. A key quantity for this count is the degree of U which is the sum of the degrees of the $\widetilde{u}_i$'s. It can be computed directly from the tuple $(w, \lambda, t, \epsilon, M)$. Then, to ensure existence and unicity of the V polynomial to represent the semi-reduced divisor, we have to impose that $\deg V < \deg U$, so that we have exactly $\deg U$ variables for the coefficients of V. For the polynomials P and Q, we need the degree of $P^2 - Q^2 f$ to be exactly $\deg U$. After a normalization like in Section 5.3 depending on the parity of $\deg U$, we get $\deg U - g$ variables for their coefficients.

Variables	Number of variables	Bound
Coordinates (x_i, y_i) of P_i	$2k$	$2g$
Abscissae ξ_j of shared points	s, column-size of the matrix M	g^2
Coefficients of the $\widetilde{u}_i$ polynomials	$\deg U = \sum_i (t_i - \sum_j (\lvert m_{ij}\rvert - \widetilde{m}_{ij}))$	g^2
Coefficients of the V polynomial	$\deg U$	g^2
Coefficients of the P and Q polynomials	$\deg U - g$	$g^2 - g$
Total	$s + 2k + 3 \deg U - g$	$4g^2 + g$

Table 5.1: Summary of the variables in the polynomial system corresponding to a normalized non-genericity tuple $(w, \lambda, t, \epsilon, M)$.

In order to apply Proposition 5.3, we need to evaluate the degrees of all the equations and inequalities that we have listed, with respect to two groups of variables: The first group contains just the variables x_i and y_i, and we will denote $\deg_1(f)$ the degree of a polynomial f with respect to those variables (said otherwise, $\deg_1(f)$ is the degree of f if we consider only the symbols x_i, y_i as variables, and all the other indeterminates are considered as parameters). The second group of variables contains all the other indeterminates and the degree with respect to this group is denoted by $\deg_2$.

The crucial point is to ensure that each polynomial equation has a $\deg_1$ bounded by $O_g(\ell^3)$, while $\deg_2$ is bounded by $O_g(1)$. For the inequalities, we require the same degree conditions:

Indeed, an inequality $f \neq 0$ can be modeled by the equality $T \cdot f - 1 = 0$, where T is a fresh variable that belongs to our second group of variables. This trick requires only one more variable for each inequality and the degree of the equation $T \cdot f - 1 = 0$ is only one more than the degree of the inequality. Since the number of inequalities is bounded by $O_g(1)$, the number of extra variables required in the second group will not impact the asymptotic complexity (the second group already contains $O_g(1)$ variables). We remark that the input of the geometric resolution algorithm over fields of characteristic 0 in [64] allows inequalities. However, we use the aforementioned trick to model inequalities by equalities since the solving method that we use is the variant for positive characteristic whose complexity analysis is given in [25, Thm. 4.8].

The number of equations and inequalities and their degrees with respect to the two groups of variables can be easily checked and are summarized in Table 5.2.

Equations reference	Number of equations (and bound)	$\deg_1$	$\deg_2$
Eq. and Ineq. Sys.1	$2k \leq 2g$	$2g+1$	0
InEq. Sys.2	$k(k-1)/2 \leq g(g-1)/2$	1	0
Eq. and Ineq. Sys.3	$\leq 2g$	$O_g(\ell^3)$	0
InEq. Sys.4	$\leq g$	$O_g(\ell^3)$	0
Eq. Sys.5	$\sum_{i=1}^{k}\sum_{j=1}^{s} \lvert m_{ij}\rvert \leq g^4$	$O_g(\ell^3)$	$\leq g$
InEq. Sys.6	$ks \leq g^3$	$O_g(\ell^3)$	$\leq g$
Eq. Sys.7 and Sys.8	$\leq k^2 s \leq g^4$	$O_g(\ell^3)$	$\leq g$
InEq. Sys.9	$\leq s^2 \leq g^4$	0	1
Eq. Sys.10	$\sum_{i=1}^{k} t_i \leq g^2$	$O_g(\ell^3)$	$\leq g$
Eq. Sys.11	$\deg U \leq g^2$	0	$O(g^3)$
Eq. Sys.12	$\sum_{i=1}^{k} \deg \widetilde{u}_i \leq g^2$	$O_g(\ell^3)$	$O(g^2)$
Eq. Sys.13	$\deg U \leq g^2$	0	$O(g^3)$

Table 5.2: Summary of the degrees of the equations in the polynomial system corresponding to a normalized non-genericity tuple $(w, \lambda, t, \epsilon, M)$.

Finally, since we have been very careful in describing elements that are ℓ-torsion points on J, without room for parasite solutions or multiplicities, we can again appeal to the finite and étale property of multiplication by ℓ in J to deduce that the system is 0-dimensional and radical. Therefore, by Proposition 5.3, each system can be solved in the claimed complexity bound. To conclude the proof of Proposition 5.2, and hence of our main result, we need a few more observations.

First, notice that the solutions of our polynomial systems can be grouped by weight of the ℓ-torsion divisor: once geometric resolutions of two 0-dimensional sets V_1 and V_2 are known, a geometric resolution of $V_1 \cup V_2$ can be computed very efficiently. The strategy to do so is to change the primitive element of the geometric resolutions for a random element, so that both resolution share the same primitive element. This can done within complexity linear in the number of variables and polynomial in $\deg(V_1 \cup V_2)$ using Algorithm 6 in [64]. Then, computing the LCM of the univariate polynomials of the geometric resolutions and interpolating the parametrization provides a geometric resolution of $V_1 \cup V_2$. Using this procedure for regrouping the solutions of all the systems derived from the non-degeneracy tuples with the same weight w provides geometric solutions of $J_w[\ell]$ within the claimed complexity.

Finally, we need to transform the Monte Carlo algorithm from Proposition 5.3 in a Las Vegas algorithm. This can be easily achieved since the probability that the Monte Carlo algorithm succeeds is bounded below by a quantity which does not depend on the input size, and the output

can be verified since we know that the sum of the degrees of the varieties $J_w[\ell]$ for $w \in [1, g]$ must equal $\ell^{2g} - 1$. Consequently, once all polynomial systems corresponding to non-generic situations have been solved, it is easy to count the number of ℓ-torsion elements found and to check that none of them is missing by comparing their number with the theoretical value $\ell^{2g} - 1$. The Las Vegas algorithm consists in repeating the Monte Carlo algorithm until the result is verified and is correct (i.e. all elements found are ℓ-torsion elements and none of them is missing). The expected complexity of the Las Vegas variant equals the complexity of the Monte Carlo variant up to multiplication by a constant. This concludes the proof of Proposition 5.2.

Chapter 6

The case of genus-3 hyperelliptic curves with RM

Contrary to p-adic methods that have been adapted to any genus, implementations of ℓ-adic point-counting algorithms were limited to genus 1 and 2, probably because of the lack of cryptographic applications of genus-3 curves but also because such an algorithm would very likely have a prohibitive complexity that would impede any practical attempt. In fact, the complexity of a genus-3 analogue of Schoof's algorithm is subject to speculations as mentioned in [70] with an estimation in $\widetilde{O}(\log^{12} q)$ that is prohibitive indeed. However, as in genus 2, we may try to find easier instances and in particular consider the RM case.

The aim of this chapter is thus to show — both with theoretical proofs and practical experiments — that the complexity of ℓ-adic methods for genus-3 hyperelliptic curves can be dramatically decreased as soon as an explicitly computable non-integer endomorphism $\eta \in \mathrm{End}(\mathrm{Jac}(\mathcal{C}))$ is known. More precisely, we consider $\mathcal{C}$ a genus-3 hyperelliptic curve with explicit RM by $\mathbb{Z}[\eta]$ in the sense of Definition 3.3. This means that we have explicit formulas describing $\eta(P - P_\infty)$ for P a generic point of $\mathcal{C}$. By explicit formulas, we mean polynomials $(\eta_i^{(u)}(x,y))_{i\in\{0,1,2,3\}}$ and $(\eta_i^{(v)}(x,y))_{i\in\{0,1,2,3\}}$ in $\mathbb{F}_q[x,y]$, such that, when $\mathcal{C}$ is given in odd-degree Weierstrass form, the Mumford coordinates of $\eta(x,y)$ are $\left\langle \sum_{i=0}^{3} \eta_i^{(u)}(x,y)X^i, \sum_{i=0}^{2} (\eta_i^{(v)}(x,y)/\eta_3^{(v)}(x,y))X^i \right\rangle$, where (x,y) is the generic point of the curve. In cases where $\mathcal{C}$ does not have an odd-degree Weierstrass model, we can work in an extension of degree at most 8 of the base field in order to ensure the existence of a rational Weierstrass point.

Examples of curves with RM are given by modular curves. For instance, the genus-3 curve $y^2 = x^7 + 3x^6 + 2x^5 - x^4 - 2x^3 - 2x^2 - x - 1$ is a quotient of $X_0(284)$ and therefore has real multiplication by an element of $\mathbb{Q}[x]/(x^3 - 3x - 1)$. This follows from the properties of the Hecke operators as explained in [130, Chapter 7]. Based on this theory, algorithms for constructing such curves are explained in [50]; however the explicit expression for the real endomorphism is not given. We expect that tracking the Hecke correspondences along their construction, and using techniques like in [144] to reconstruct the rational fractions describing the real endomorphism could solve this question. In any case, these are only isolated points in the moduli space. Larger families are obtained from cyclotomic covering. This line of research has produced several families of hyperelliptic genus-3 curves having explicit RM by $\mathbb{Z}[2\cos(2\pi/7)]$. In particular, such explicit families are given in [101] and [138], and explicit formulas for their RM endomorphism are obtained in [87]. We use the 1-dimensional family of curves from [138, Theorem 1 with $p = 7$] for our experiments. Other families of genus-3 curves (but not necessarily hyperelliptic)

with RM have been made explicit in [23, Chapter 2], following [43]. We would like to point out that within the moduli space of complex polarized abelian varieties of dimension 3, those with RM by a fixed order in a cubic field form a moduli space of codimension 3 [19, Sec. 9.2]. Since Jacobians of hyperelliptic curves form a codimension 1 space, we would expect the moduli space of hyperelliptic curves of genus 3 with RM by a given cubic order to have dimension 2.

We insist on the fact that all the $O()$ and the $\widetilde{O}()$ notation used throughout the chapter should be understood up to a multiplicative constant which may depend on the ring $\mathbb{Z}[\eta]$ and on the degrees of the polynomials $\eta_i^{(u)}$ and $\eta_i^{(v)}$. There are natural families of curves for which these degrees are bounded by an absolute constant and for which $\mathbb{Z}[\eta]$ is fixed: reductions at primes (of good reduction) of a hyperelliptic curve with explicit RM defined over a number field. Most of this chapter is joint work with Pierrick Gaudry and Pierre-Jean Spaenlehauer and is to appear as [2].

Organization of the chapter. In Section 6.1 we give an overview of both our algorithm and its complexity. The main task is the computation of kernels of some endomorphisms detailed in Section 6.3. This is achieved by solving a polynomial system using resultants. Section 6.4 is devoted to implementation of the algorithm using Gröbner bases instead of resultants and ending with an exponential collision search which can be run massively in parallel. Indeed, although using Gröbner bases seems to be more efficient in practice, we do not see any hope of proving with rigorous arguments that it is asymptotically competitive.

6.1 Overview of the algorithm

Let $\mathcal{C}$ be a genus-3 hyperelliptic curve over a finite field $\mathbb{F}_q$ with explicit RM, and let η be the given explicit endomorphism. We denote by μ_0, μ_1, μ_2 the coefficients of the minimal polynomial $T^3 + \mu_2 T^2 + \mu_1 T + \mu_0$ of η over $\mathbb{Q}$.

6.1.1 The characteristic equation of the Frobenius

The characteristic polynomial of the Frobenius endomorphism π is of the form $\chi_\pi(T) = T^6 - \sigma_1 T^5 + \sigma_2 T^4 - \sigma_3 T^3 + q\sigma_2 T^2 - q^2\sigma_1 T + q^3$, and Weil's bounds give

$$|\sigma_1| \leq 6\sqrt{q}, \quad |\sigma_2| \leq 15q, \quad |\sigma_3| \leq 20q^{3/2}.$$

In order to take advantage of the explicit RM, we consider the endomorphism $\psi = \pi + \pi^\vee$, for which we can derive the real Weil polynomial $\chi_\psi(T) = T^3 - \sigma_1 T^2 + (\sigma_2 - 3q)T - (\sigma_3 - 2q\sigma_1)$, which corresponds to the characteristic polynomial of ψ viewed as an element of the real subfield of $\mathrm{End}(\mathrm{Jac}(C)) \otimes \mathbb{Q}$. The endomorphism ψ belongs to the ring of integers of $\mathbb{Q}(\eta)$. The ring $\mathbb{Z}[\eta]$ might be a proper sub-order of the ring of integers, so let us call Δ its index, so that ψ can be written $\psi = a + b\eta + c\eta^2$, where a, b, c are rationals with a denominator that divides Δ. By computing formally the characteristic polynomial of $a + b\eta + c\eta^2$ in $\mathbb{Q}(\eta)$ and by equating it with the expression for the characteristic polynomial of $\chi_\psi(T)$, we obtain a direct way to compute σ_1, σ_2 and σ_3 in terms of a, b, c:

$$\begin{aligned}
\sigma_1 &= 3\,a - b\,\mu_2 - 2\,c\,\mu_1 + c\,\mu_2^2\ , \\
\sigma_2 - 3q &= 3\,a^2 - 2\,a\,b\,\mu_2 + 2\,a\,c\,(\mu_2^2 - 2\mu_1) + b^2\,\mu_1 + 3\,b\,c\,\mu_0 - b\,c\,\mu_1\,\mu_2 - \\
&\quad c^2\,(2\,\mu_0\,\mu_2 + \mu_1^2)\ , \\
\sigma_3 - 2q\sigma_1 &= a^3 - a^2\,b\,\mu_2 + a^2\,c\,(\mu_2^2 - 2\mu_1) + a\,b^2\,\mu_1 + a\,b\,c\,(3\,\mu_0 - \mu_1\,\mu_2) + \\
&\quad a\,c^2\,(\mu_1^2 - 2\,\mu_0\,\mu_2) - b^3\,\mu_0 + b^2\,c\,\mu_0\,\mu_2 - b\,c^2\,\mu_0\,\mu_1 + c^3\,\mu_0^2\ .
\end{aligned} \tag{6.1}$$

In Section 6.2.1, it is shown that the coefficients a, b and c can be bounded in $O(\sqrt{q})$. More precisely, we denote by C_{abc} a constant that depends only on η such that their absolute values are bounded by $C_{abc}\sqrt{q}$. Since these bounds are much smaller than the bounds for σ_1, σ_2, σ_3, it makes sense to design an algorithm that reconstruct these coefficients of ψ instead of the coefficients of χ_π as in the classical Schoof algorithm, and this is what we are going to do later on.

Another important bound that we need concerns the size of small elements that can be found in ideals of $\mathbb{Z}[\eta]$. Let ℓ be a prime that splits completely in $\mathbb{Z}[\eta]$, so that we can write $\ell = \mathfrak{p}_1\,\mathfrak{p}_2\,\mathfrak{p}_3$, where the $\mathfrak{p}_i$'s are distinct prime ideals of norm ℓ. In Section 6.2.2, it is shown that each $\mathfrak{p}_i$ contains a non-zero element $\alpha_i = a_i + b_i\eta + c_i\eta^2$, where a_i, b_i and c_i are integers and are bounded in absolute value by $O(\ell^{1/3})$.

6.1.2 A point-counting algorithm

Our genus-3 RM point counting algorithm is Algorithm 8. We give a description of it, allowing some black-box primitives that will be detailed in dedicated sections. As mentioned above, we will work with the a, b, c coefficients of the ψ endomorphism. More precisely, we compute their values modulo sufficiently many completely split primes ℓ until we can deduce their values from the bounds of Lemma 6.1 from Section 6.2.1 by the Chinese Remainder Theorem, taking into account their potential denominator Δ. Then the coefficients of χ_π are deduced by Equations (6.1).

We now explain how the algorithm works for a given split ℓ. First its decomposition as a product of prime ideals $\ell\,\mathbb{Z}[\eta] = \mathfrak{p}_1\,\mathfrak{p}_2\,\mathfrak{p}_3$ is computed, and for each prime ideal $\mathfrak{p}_i$, a non-zero element α_i of $\mathfrak{p}_i$ is found with a small representation $\alpha_i = a_i + b_i\eta + c_i\eta^2$ as in Lemma 6.2 of Section 6.2.2. The kernel of α_i is denoted by $J[\alpha_i]$ and it contains a subgroup G_i isomorphic to $\mathbb{Z}/\ell\,\mathbb{Z} \times \mathbb{Z}/\ell\,\mathbb{Z}$, since the norm of α_i is a small multiple of ℓ. We further denote by λ_i the eigenvalue of η in $J[\ell]$ such that $\mathfrak{p}_i$ is the ideal $(\ell, \eta - \lambda_i)$.

On $G_i \subset J[\alpha_i]$, the endomorphism η acts as the multiplication by λ_i. Therefore, $\psi = a + b\eta + c\eta^2$ also acts as a scalar multiplication on this 2-dimensional space, and we write $k_i \in \mathbb{Z}/\ell\,\mathbb{Z}$ the corresponding eigenvalue: for any D_i in G_i, we have $\psi(D_i) = k_i D_i$. On the other hand, from the definition of ψ, it follows that $\psi\pi = \pi^2 + q$. Therefore, if such a D_i is known, we can test which value of $k_i \in \mathbb{Z}/\ell\,\mathbb{Z}$ satisfies

$$k_i\pi(D_i) = \pi^2(D_i) + qD_i. \tag{6.2}$$

Since ℓ is a prime and D_i is of order exactly ℓ, this is also the case for $\pi(D_i)$. Finding k_i can then be seen as a discrete logarithm problem in the subgroup of order ℓ generated by $\pi(D_i)$; hence the solution is unique. Equating the two expressions for ψ, we get explicit relations between a, b, c modulo ℓ:

$$a + b\lambda_i + c\lambda_i^2 \equiv k_i \bmod \ell.$$

Therefore we have a linear system of three equations in three unknowns, the determinant of which is the Vandermonde determinant of the λ_i, which are distinct by hypothesis. Hence the system can be solved and it has a unique solution modulo ℓ.

It remains to show how to construct a divisor D_i in G_i, i.e. an element of order ℓ in the kernel $J[\alpha_i]$. Since an explicit expression of η as an endomorphism of the Jacobian of $\mathcal{C}$ is known, an explicit expression can be deduced for α_i, using the explicit group law. The coordinates of the elements of this kernel are solutions of a polynomial system that can be directly derived from this expression of α_i. Using standard techniques, it is possible to find the solutions of this

input : q an odd prime power, and $f \in \mathbb{F}_q[X]$ a monic squarefree polynomial of degree 7 such that the curve $Y^2 = f(X)$ has explicit RM by $\mathbb{Z}[\eta]$.
output: The characteristic polynomial $\chi_\pi \in \mathbb{Z}[T]$ of the Frobenius endomorphism on the Jacobian J of the curve.

$R \leftarrow 1$;
while $R \leq 2\,\Delta\,C_{abc}\,\sqrt{q} + 1$ **do**
  Pick the next prime ℓ that satisfies conditions (C1) to (C4);
  Compute the ideal decomposition $\ell\,\mathbb{Z}[\eta] = \mathfrak{p}_1\,\mathfrak{p}_2\,\mathfrak{p}_3$, corresponding to the eigenvalues λ_1, λ_2, λ_3 of η in $J[\ell]$;
  for $i \leftarrow 1$ **to** 3 **do**
    Compute a small element α_i of $\mathfrak{p}_i$ as in Lemma 6.2;
    Compute a non-zero element D_i of order ℓ in $J[\alpha_i]$;
    Find the unique $k_i \in \mathbb{Z}/\ell\,\mathbb{Z}$ such that $k_i\pi(D_i) = \pi^2(D_i) + qD_i$;
  end
  Find the unique triple (a, b, c) in $(\mathbb{Z}/\ell\,\mathbb{Z})^3$ such that $a + b\lambda_i + c\lambda_i^2 = k_i$, for i in $\{1, 2, 3\}$;
  $R \leftarrow R \cdot \ell$;
end
Reconstruct (a, b, c) using the Chinese Remainder Theorem ;
Deduce χ_π from Equations (6.1).

Algorithm 8: Overview of our genus-3 RM point-counting algorithm

system, perhaps in a finite extension of the base field (of degree bounded by the degree of the ideal generated by the system, i.e. in $O(\ell^2)$), from which divisors in $J[\alpha_i]$ can be constructed. Multiplying by the appropriate cofactor, we can reach all the elements of G_i; but we stop as soon as we get a non-trivial one.

We summarize the conditions that must be satisfied by the primes ℓ that we work with:

(C1) ℓ must be different from the characteristic of the base field;

(C2) ℓ must be coprime to the discriminant of the minimal polynomial of η;

(C3) there must exist $\alpha_i \in \mathfrak{p}_i$ as in Lemma 6.2 with norm non-divisible by ℓ^3 for $i \in \{1, 2, 3\}$;

(C4) the ideal $\ell\,\mathbb{Z}[\eta]$ must split completely.

The first 3 conditions eliminate only a finite number of ℓ's that depends only on η, while the last one eliminates a constant proportion. The condition (C3) implies that there is a unique subgroup G_i of order ℓ^2 in $J[\alpha_i]$ (our description of the algorithm could actually be adapted to handle the cases where this is not true).

6.1.3 Complexity overview

The field $\mathbb{Q}(\eta)$ is of degree 3, so its Galois group has order at most 6 and by Chebotarev's density theorem the density of primes that split completely is at least 1/6. Therefore the main loop is done $O(\log q/\log\log q)$ times, with primes ℓ that are in $O(\log q)$. All the steps that take place in the number field take a negligible time. For instance, a small generator like in Lemma 6.2 can be found by exhaustive search: only $O(\ell)$ trials are needed since we are searching over all elements of the form $a + b\eta + c\eta^2$, with $|a|, |b|, |c|$ in $O(\ell^{1/3})$.

The bottleneck of the algorithm is the computation of a non-zero element of order ℓ in the kernel $J[\alpha_i]$ of α_i. This part will be treated in detail in Section 6.3, where it is shown to be feasible in $\widetilde{O}(\ell^4)$ operations in $\mathbb{F}_q$. The output is a divisor D_i of order ℓ in $J[\alpha_i]$ that is defined over an extension field $\mathbb{F}_{q^\delta}$, where δ is in $O(\ell^2)$.

In order to check Equation (6.2), we first need to compute $\pi(D_i)$ and $\pi^2(D_i)$ which amounts to raising the coordinates to the q-th power. The cost is in $\widetilde{O}(\ell^2 \log q)$ operations in $\mathbb{F}_q$. Then, each Jacobian operation in the group generated by $\pi(D_i)$ costs $\widetilde{O}(\ell^2)$ operations in the base field, and we need $O(\sqrt{\ell})$ of them to solve the discrete logarithm problem given by Equation (6.2). The overall cost of finding k_i, once D_i is known is therefore $\widetilde{O}(\ell^2(\sqrt{\ell} + \log q))$ operations in $\mathbb{F}_q$.

Finally, the amount of work performed for each ℓ is $\widetilde{O}(\ell^2(\ell^2 + \log q))$ operations in the base field $\mathbb{F}_q$. Summing up for all the primes, and taking into account the cost of the operations in $\mathbb{F}_q$, we obtain a global bit-complexity of $\widetilde{O}((\log q)^6)$.

6.2 Bounds for Algorithm 8

6.2.1 Bounds on the coefficients of ψ

The system of equations (6.1) giving σ_1, σ_2 and σ_3 in terms of a, b, c is homogeneous if we put weight $1/2$ to a, b, c and σ_1, weight 1 to q and σ_2, weight $3/2$ to σ_3, and weight 0 to μ_0, μ_1, and μ_2 so any polynomial in a reduced Gröbner basis of the corresponding ideal will have the same property. Computing such a Gröbner basis with the lexicographical ordering $a > b > c > \sigma_1 > \sigma_2 > \sigma_3 > \mu_0 > \mu_1 > \mu_2 > q$ (we did this computation with the Magma V2.23-4 software), we get a polynomial Ψ_c of degree 6 in c that does not involve a or b, and which has the following form:

$$\Psi_c(q, c, \sigma_1, \sigma_2, \sigma_3, \mu_0, \mu_1, \mu_2) = D(\mu_0, \mu_1, \mu_2)^3\, c^6 + \textstyle\sum_{i=0}^{5} \psi_c^{(i)}(q, \sigma_1, \sigma_2, \sigma_3, \mu_0, \mu_1, \mu_2)\, c^i,$$

where $D(\mu_0, \mu_1, \mu_2) = -27\,\mu_0^2 + 18\,\mu_0\mu_1\mu_2 - 4\,\mu_0\mu_2^3 - 4\,\mu_1^3 + \mu_1^2\mu_2^2$ is the discriminant of the polynomial $T^3 + \mu_2 T^2 + \mu_1 T + \mu_0$.

By computing Gröbner bases for other lexicographical orderings (with $a > c > b > \sigma_1 > \sigma_2 > \sigma_3 > \mu_0 > \mu_1 > \mu_2 > q$ and $b > c > a > \sigma_1 > \sigma_2 > \sigma_3 > \mu_0 > \mu_1 > \mu_2 > q$ respectively), we obtain that polynomials of the following form also belong to the ideal generated by the polynomials in the system of equations (6.1):

$$\Psi_b(q, b, \sigma_1, \sigma_2, \sigma_3, \mu_0, \mu_1, \mu_2) = D(\mu_0, \mu_1, \mu_2)^3\, b^6 + \textstyle\sum_{i=0}^{5} \psi_b^{(i)}(q, \sigma_1, \sigma_2, \sigma_3, \mu_0, \mu_1, \mu_2)\, b^i,$$
$$\Psi_a(q, a, \sigma_1, \sigma_2, \sigma_3, \mu_0, \mu_1, \mu_2) = D(\mu_0, \mu_1, \mu_2)^3\, a^6 + \textstyle\sum_{i=0}^{5} \psi_a^{(i)}(q, \sigma_1, \sigma_2, \sigma_3, \mu_0, \mu_1, \mu_2)\, a^i.$$

The polynomials $\psi_a^{(i)}$, $\psi_b^{(i)}$ and $\psi_c^{(i)}$ are homogeneous of weighted degree $3 - i/2$ with respect to the grading given above.

Lemma 6.1. *The absolute values of the coefficients a, b, c of $\psi = a + b\eta + c\eta^2$ are bounded above by $O(q^{1/2})$.*

Proof. First, we consider the equation $\Psi_c = 0$. We write $c = \widetilde{c}\, q^{1/2}$, $\sigma_1 = \widetilde{\sigma_1}\, q^{1/2}$, $\sigma_2 = \widetilde{\sigma_2}\, q$, $\sigma_3 = \widetilde{\sigma_3}\, q^{3/2}$. Since $\psi_c^{(i)}$ is homogeneous and has weighted degree $3 - i/2$, there is a polynomial $\theta_c^{(i)}(\widetilde{\sigma_1}, \widetilde{\sigma_2}, \widetilde{\sigma_3}, \mu_0, \mu_1, \mu_2)$ such that

$$\psi_c^{(i)}\,(q, \sigma_1, \sigma_2, \sigma_3, \mu_0, \mu_1, \mu_2) \cdot c^i = q^3 \widetilde{c}^{\,i}\, \theta_c^{(i)}(\widetilde{\sigma_1}, \widetilde{\sigma_2}, \widetilde{\sigma_3}, \mu_0, \mu_1, \mu_2). \tag{6.3}$$

Weil's bounds imply that $|\widetilde{\sigma_i}| = O(1)$ for $i \in \{1, 2, 3\}$. Therefore, for all $i \in \{0, \ldots, 5\}$, we obtain that $|\theta_c^{(i)}(\widetilde{\sigma_1}, \widetilde{\sigma_2}, \widetilde{\sigma_3}, \mu_0, \mu_1, \mu_2)| = O(1)$. For fixed $\mu_0, \mu_1, \mu_2 \in \mathbb{Q}$ such that $\mu_0 + \mu_1 T + \mu_2 T^2 + T^3$

is the minimal polynomial of a totally real algebraic number, the discriminant $D(\mu_0, \mu_1, \mu_2)$ must be nonzero. Equations $\Psi_c = 0$ and (6.3) imply the following inequality:

$$|\widetilde{c}|^6 - \sum_{i=0}^{5} \frac{|\theta_c^{(i)}(\widetilde{\sigma_1}, \widetilde{\sigma_2}, \widetilde{\sigma_3}, \mu_0, \mu_1, \mu_2)|}{|D(\mu_0, \mu_1, \mu_2)|^3} |\widetilde{c}|^i \leq 0.$$

Then $|\widetilde{c}|$ must be smaller or equal to the largest root of this polynomial inequality, which can itself be bounded, for instance, with Cauchy's bound

$$|\widetilde{c}| \leq 1 + \max_{0 \leq i \leq 5} \left\{ \frac{|\theta_c^{(i)}(\widetilde{\sigma_1}, \widetilde{\sigma_2}, \widetilde{\sigma_3}, \mu_0, \mu_1, \mu_2)|}{|D(\mu_0, \mu_1, \mu_2)|^3} \right\},$$

which shows that $|\widetilde{c}| = O(1)$, and hence $|c| = O(q^{1/2})$. The proof for the bounds on $|a|$ and $|b|$ are similar, using the equations $\Psi_a = 0$ and $\Psi_b = 0$. □

6.2.2 Small elements in ideals of $\mathbb{Z}[\eta]$

We first recall that we consider only primes ℓ that do not divide the discriminant of the minimal polynomial of η (Condition (C2)). Hence, if $\mathbb{Z}[\eta]$ is not the maximal order of $\mathbb{Q}(\eta)$, this has no consequence on the factorization properties of ℓ.

Lemma 6.2. *For any prime ℓ that splits completely in $\mathbb{Z}[\eta]$, each prime ideal $\mathfrak{p}_i$ above ℓ contains a non-zero element α_i of the form $\alpha_i = a_i + b_i\eta + c_i\eta^2$, where $|a_i|$, $|b_i|$ and $|c_i|$ are integers in $O(\ell^{1/3})$, and the norm of α_i is in $O(\ell)$.*

Proof. The coefficients of the elements of the ideal $\mathfrak{p}_i$ represented by polynomials in η form a lattice. Applying Minkowski's bound to this lattice, we obtain the existence of a non-zero element $\alpha_i = a_i + b_i\eta + c_i\eta^2$, in $\mathfrak{p}_i$ for which the L_2-norm of (a_i, b_i, c_i) is in $O(\ell^{1/3})$. From this bound on the L_2-norm, we derive a bound on the L_∞-norm, and finally on the norm of α_i as an algebraic number. At each step, the constant hidden in the $O()$ gets worse but still depends only on $\mathbb{Z}[\eta]$. □

For any given η, it is not difficult to make the constants in the $O()$ fully explicit. We do it in the particular case of $\mathbb{Z}[\eta_7]$, with $\eta_7 = 2\cos(2\pi/7)$, which is the RM used in our practical experiments. Since $\mathbb{Z}[\eta_7]$ is a principal ring, a more direct approach leads to bounds for a generator that are tighter than what would be obtained by a naive application of the previous lemma.

Lemma 6.3. *Every ideal $\mathfrak{p}_i$ of norm ℓ in $\mathbb{Z}[\eta_7]$ has a generator α_i of the form $a_i + b_i\eta_7 + c_i\eta_7^2$, where $a_i, b_i, c_i \in \mathbb{Z}$ satisfy*

$$|a_i| < 2.415 \cdot \ell^{1/3}; \quad |b_i| < 1.850 \cdot \ell^{1/3}; \quad |c_i| < 1.764 \cdot \ell^{1/3}.$$

Proof. By abuse of notation, we identify $\mathbb{Q}(\eta_7)$ with the algebraic number field $\mathbb{Q}[X]/(X^3+X^2-2X-1)$ and we let σ_1, σ_2, σ_3 be the three real embeddings of $\mathbb{Q}(\eta_7)$ in $\mathbb{R}$ and let $\epsilon_1 = 1 - \eta_7^2$ and $\epsilon_2 = 1 + \eta_7$ be a pair of fundamental units. Let μ_i be a generator of $\mathfrak{p}_i$. The logarithmic embedding $\varphi : x \mapsto (\log|\sigma_1(x)|, \log|\sigma_2(x)|, \log|\sigma_3(x)|)$ sends the set of generators of $\mathfrak{p}_i$ to the lattice generated by $\varphi(\epsilon_1)$ and $\varphi(\epsilon_2)$ translated by $\varphi(\mu_i)$. Solving a CVP for the projection of $\varphi(\mu_i)$ on the plane where the 3 coordinates sum-up to zero, we deduce a unit ξ_i such that $\alpha_i = \xi_i\mu_i$ is a generator whose real embeddings are bounded by

$$|\sigma_1(\alpha_i)| \leq 2.247 \cdot \ell^{1/3}, \quad |\sigma_2(\alpha_i)| \leq 1.803 \cdot \ell^{1/3}, \quad |\sigma_3(\alpha_i)| \leq 2.247 \cdot \ell^{1/3}.$$

Writing $\alpha_i = a_i + b_i\eta_7 + c_i\eta_7^2$, the real embeddings can also be expressed as $(\sigma_1(\alpha_i), \sigma_2(\alpha_i), \sigma_3(\alpha_i))^T = V \cdot (a_i, b_i, c_i)^T$, where V is the Vandermonde matrix of $(\sigma_1(\eta_7), \sigma_2(\eta_7), \sigma_3(\eta_7))$. A numerical evaluation of its inverse allows to translate the bounds on $\sigma_1(\alpha_i)$, $\sigma_2(\alpha_i)$, $\sigma_3(\alpha_i)$ into the claimed bounds on a_i, b_i, c_i. □

6.3 Computing kernels of endomorphisms

6.3.1 Modelling the kernel computation by a polynomial system

Let α be an explicit endomorphism of degree $O(\ell^2)$ on the Jacobian of $\mathcal{C}$, which satisfies the properties of Lemma 6.2. We want to compute a triangular polynomial system that describes the kernel $J[\alpha]$ of α. This will provide us with a nice description of a subgroup of the ℓ-torsion on which we will be able to test the action of $\psi = \pi + \pi^\vee$ and deduce a, b, c such that $\psi = a + b\eta + c\eta^2 \bmod \ell$.

We first model $J[\alpha]$ by a system of polynomial equations that we will then put in triangular form. To do so, we consider a generic divisor $D = P_1 + P_2 + P_3 - 3\infty$, where P_i is an affine point of $\mathcal{C}$ of coordinates (x_i, y_i). We then write $\alpha(D) = 0$, i.e $\alpha(P_1 - \infty) + \alpha(P_2 - \infty) = -\alpha(P_3 - \infty)$. Generically, we expect each $\alpha(P_i - \infty)$ to be of weight 3, and we write $\langle u_i, v_i\rangle$ for its Mumford form. We derive our equations by computing the Mumford form $\langle u_{12}, v_{12}\rangle$ of $\alpha(P_1 - \infty) + \alpha(P_2 - \infty)$ and then writing coefficient-wise the conditions $u_{12} = u_3$ and $v_{12} = -v_3$. The case where the genericity conditions are not satisfied is discussed at the end of the section.

Similarly to the Schoof-Pila algorithm, we define polynomials — which are equivalent to Cantor's division polynomials — by the formulas

$$u_{12}(X) = X^3 + \sum_{i=0}^{2} \frac{\tilde{d}_i(x_1, x_2, y_1, y_2)}{\tilde{d}_3(x_1, x_2)} X^i, \quad v_{12}(X) = \sum_{i=0}^{2} \frac{\tilde{e}_i(x_1, x_2, y_1, y_2)}{\tilde{e}_3(x_1, x_2)} X^i,$$
$$u_3(X) = X^3 + \sum_{i=0}^{2} \frac{d_i(x_3)}{d_3(x_3)} X^i, \quad v_3(X) = y_3 \sum_{i=0}^{2} \frac{e_i(x_3)}{e_3(x_3)} X^i.$$

Lemma 6.4. *For any $i \in \{1, 2, 3\}$, the degrees of $\tilde{d}_i$, $\tilde{e}_i$, d_i and e_i are in $O(\ell^{2/3})$.*

Proof. Let us first remark that the $\tilde{d}_i$'s and $\tilde{e}_i$'s are obtained after adding two divisors $\langle u_1, v_1\rangle$ and $\langle u_2, v_2\rangle$ such that the coefficients of the u_i and v_i are respectively the d_j/d_3 and $y_i e_j/e_3$ evaluated at x_i. Thus, since this application of the group law involves a number of operations that is bounded independently of ℓ and q, the degree stays within a constant multiplicative factor, which is captured by the $O()$. Therefore it is enough to prove the result for the d_i's and e_i's.

Since the endomorphism α satisfies the properties of Lemma 6.2, it is a linear combination of 1, η and η^2 with coefficients of size $O(\ell^{1/3})$. Using the same argument about the group law, we can further reduce our proof to the case where $\alpha = n\eta^k$, with $k \in \{0, 1, 2\}$ and n an integer in $O(\ell^{1/3})$. But once again, η^k does not depend on ℓ so that, provided we can prove that Cantor's n-division polynomials have degrees in $O(n^2)$, we have proven that $n\eta^k(P - \infty) = \eta^k(n(P - \infty))$ have coefficients whose degrees are in $O(n^2)$, and then so does $\alpha(P - \infty)$. This quadratic bound on the degrees of Cantor's division polynomials in genus 3 is precisely Theorem 4.14, whose proof is done in Section 4.3. □

6.3.2 Solving the system with resultants

Typical tools for solving a polynomial system are the F4 algorithm, methods based on geometric resolution, or homotopy techniques. To obtain reasonable complexity bounds, they all require some knowledge of the properties of the system, and this might be hard to prove. Since we have a system in essentially 3 variables (in fact, there are six variables $x_1, x_2, x_3, y_1, y_2, y_3$, but the y_i variables can be directly eliminated by using the equation defining the curve), we prefer to stick to an approach based on resultants. It ends up having a complexity that is quasi-quadratic in the degree of the ideal, which is the best that can be hoped for anyway for all of the advanced techniques, and the complexity analysis requires only elementary tools. A complication that can occur with resultants is that $\mathrm{Res}_x(f, g)$ is identically zero when f and g have a nonconstant GCD. This is not a problem in our case since we can divide polynomials f and g by their GCD, by factoring them at the cost of $O(\max(\deg(f), \deg(g))^\omega)$ field operations — where $\omega < 2.38$ is the exponent of linear algebra — using the bivariate recombination methods in [22] (the trivariate case can be reduced to the bivariate case by using the techniques in [147, Sec. 21.2]). In what follows, the complexities of computing the resultants are larger than $O(\max(\deg(f), \deg(g))^\omega)$, so we can forget about this complication.

Following our modelling, the equality of the u-coordinates gives three equations

$$\forall i \in \{0, 1, 2\}, \quad \tilde{d}_i(x_1, x_2, y_1, y_2) d_3(x_3) = \tilde{d}_3(x_1, x_2) d_i(x_3), \tag{6.4}$$

of degree $O(\ell^{2/3})$ in the x_i's. By computing resultants with the equations $y_i^2 = f(x_i)$, we derive three equations $E_i(x_1, x_2, x_3) = 0$ whose degrees are still in $O(\ell^{2/3})$.

We then eliminate x_1 by computing 3 trivariate resultants R_i (between the two equations E_j with $j \neq i$). We get three equations $R_i(x_2, x_3) = 0$ of degrees $O(\ell^{4/3})$ within a complexity in $\widetilde{O}(\ell^{10/3})$ field operations, as proven in Proposition 2.40.

Then, we compute bivariate resultants S_i (between the two equations R_j with $j \neq i$) to eliminate x_2. From Proposition 2.39, we get three univariate equations $S_i(x_3) = 0$ of degree bounded by $O(\ell^{8/3})$ for a complexity in $\widetilde{O}(\ell^4)$ field operations. And we compute the polynomial $S(x_3)$ as the GCD of the $S_i(x_3)$, which belongs to the ideal defined by our original system.

The bound on the degree of S is much larger than $\ell^2 - 1$, the expected degree of the kernel. Although we can expect the actual degree to be in $O(\ell^2)$, we need to add the constraints coming from the v-coordinates to be able to prove it.

The polynomial system coming from $v_{12} = -v_3$ has the same characteristics as the one coming from the u-coordinates. Therefore, we can proceed in a similar way and deduce, at a cost of $\widetilde{O}(\ell^4)$ operations another univariate polynomial $\tilde{S}(x_3)$ belonging to the ideal. Now, since all the original equations have been taken into account all common roots of S and $\tilde{S}$ will correspond to a solution of the original system for which we know that there are $O(\ell^2)$ solutions. Therefore taking the squarefree part of the GCD of S and $\tilde{S}$ yields a polynomial of degree $O(\ell^2)$.

This univariate polynomial can be factored at a cost of $\widetilde{O}(\ell^4)$ operations in $\mathbb{F}_q$ with standard algorithms [54] (there exist asymptotically faster algorithms, but we already fit in our target complexity). We then deal with each irreducible factor in turn, until one is found that leads to a genuine solution of the original system. Let δ be the degree of such an irreducible factor $\phi(x_3)$. In the field extension $\mathbb{F}_{q^\delta} = \mathbb{F}_q[x_3]/(\phi(x_3))$, we have by construction a root x_3 of ϕ. We then solve again the original polynomial system where x_3 is instantiated with this root. This system is bivariate in x_1 and x_2 and there are $O(1)$ solutions, that possibly live in another finite extension $\mathbb{F}_{q^{\delta'}}$ of $\mathbb{F}_{q^\delta}$. Since the degrees of the bivariate polynomials are in $O(\ell^{2/3})$, by Proposition 2.39, this system solving costs $\widetilde{O}(\ell^2)$ operations in $\mathbb{F}_{q^\delta}$.

A solution obtained in this way must be checked, because it could come from a vanishing denominator that has been cleared when constructing the system or from non-generic situations. But given a set of candidate coordinates for a D_i element of $J[\alpha_i]$, it is cheap to check that this is indeed an element of the Jacobian and that it is killed by α_i. Also, if α_i is not a generator of $\mathfrak{p}_i$, it is necessary to check the order of D_i: if this is a multiple of ℓ, then multiplying D_i by the cofactor gives an order-ℓ element. But it is also possible to get an unlucky element that is of a small order coprime to ℓ, and then we have to take another solution of the system.

Since an operation in $\mathbb{F}_{q^\delta}$ requires a number of operations in $\mathbb{F}_q$ that is quasi-linear in δ, and since the sum of all the degrees δ of the irreducible factors of $\mathrm{GCD}(S, \tilde{S})$ is in $O(\ell^2)$, the amortized cost is $\widetilde{O}(\ell^4)$ operations in $\mathbb{F}_q$ to deduce a divisor D_i in $J[\alpha_i]$.

Note that using the algorithm of Villard mentioned at the end of Section 2.3 for bivariate resultants, the complexity is lowered as follows. First, to compute trivariate resultants of polynomials whose degrees are bounded by d, we perform a Kronecker substitution and compute bivariate resultants of equations of degrees $d_x \leq d^3$ and $d_y \leq d$. Since $d = \ell^{2/3}$, we end up with a complexity in $O(\ell^{2/3(2-1/\omega)+2+o(1)})$, which is dominated by $\ell^{3+1/9}$. Once this is done, it remains to compute bivariate resultants of equations with degrees in x and y both smaller than $2d^2$. This yields a complexity in $O(\ell^{4/3(2-1/\omega)}\ell^{4/3+o(1)})$ field operations. This is dominated by $O(\ell^{3+5/9})$. It follows from our complexity analysis that the overall complexity of Algorithm 8 is thus decreased by a factor $(\log q)^{4/9}$ at least.

6.3.3 Remarks

In Section 6.3, the algorithms work by evaluation / interpolation, which requires to have enough elements in the base field. Were it not the case, we simply take a field extension $\mathbb{F}_{q^\delta}$ of $\mathbb{F}_q$, that will add a factor $\widetilde{O}(\delta)$ to the complexity. The complexity of the algorithms will be polynomial in the number of evaluation points, therefore, δ will be logarithmic in the final complexity, so that the cost of taking a field extension will be hidden in the $\widetilde{O}()$ notation.

Another difficulty is that an evaluation / interpolation strategy assumes that the points of evaluation are generic enough, so that all the degrees after evaluation are generic. This is again guaranteed by taking a large enough base field. Still, the algorithm remains a Monte-Carlo one. However, the ultimate goal is to construct kernel elements, which is an easily verified property. Turning this into a Las Vegas algorithm can therefore be done with standard techniques.

Last but not least, our analysis assumes in the first place that the ℓ-torsion elements are generic in a rather strong sense, as in Definition 5.6. This is expected to be the case with overwhelming probability, when the base field is large enough and the curve is taken at random in a large family. However, to obtain a proven complexity we must also consider the cases where there exist ℓ-torsion elements that are non-generic. We follow the strategy of Section 5.4 where another polynomial system is designed and solved for each non-generic situation, for instance the fact that an ℓ-torsion divisor is of weight less than 3, or that some points involved in the modelling are not distinct while they generically are. We do not give all the details, but the number of polynomial systems to consider is bounded by a constant, and each of these polynomial systems describes a situation that is smaller than the generic one in the sense that it has either less variables or a lower degree, so that the complexity bound is maintained.

6.4 Practical results

In this section, we compute the zeta function of a genus 3 hyperelliptic curve with explicit RM defined over $\mathbb{F}_p$ with $p = 2^{64} - 59$. To our knowledge the largest genus-3 computation that had been achieved previously was the computation of the zeta function of a hyperelliptic curve defined over $\mathbb{F}_p$ with $p = 2^{61} - 1$, done by Sutherland [133] using generic group algorithms.

In order to evaluate the practicality of our algorithm, we have tested it on one of the families of genus-3 hyperelliptic curves having explicit RM given in [138, Theorem 1]. Formulas for their RM endomorphisms are described in [87]: for $t \neq \pm 2$, the curve $\mathcal{C}_t$ with equation

$$y^2 = x^7 - 7x^5 + 14x^3 - 7x + t,$$

admits an endomorphism given in Mumford representation by

$$\eta_7(x, y) = \langle X^2 + 11\, xX/2 + x^2 - 16/9, y \rangle.$$

The fact that this expression has degree 2 while one would generically expect a degree 3 is no accident: it comes from the construction in [138] of the endomorphism as a sum of two automorphisms on a double cover of the curve. We have $\eta_7^3 + \eta_7^2 - 2\eta_7 - 1 = 0$, so that the ring $\mathbb{Z}[\eta_7]$ is isomorphic to the ring of integers $\mathbb{Z}[2\cos(2\pi/7)]$ of the real subfield of the cyclotomic field $\mathbb{Q}(e^{2i\pi/7})$. All the numerical data in this section have been obtained for the parameter $t = 42$, on the prime field $\mathbb{F}_p$ with $p = 2^{64} - 59$.

In our practical computations, the main differences with the theoretical description are the following: we use Gröbner basis algorithms instead of resultants, we consider also small non-split primes ℓ and small powers, and we finish the computation with a parallel collision search. The source code for our experiments is available at `https://members.loria.fr/SAbelard/RMg3.tgz`.

6.4.1 Retrieving modular information

Although the polynomial system resolution using resultants has a complexity in $\widetilde{O}(\ell^4)$, the real cost for small values of ℓ is already pretty large. In the resolution method described in Section 2.3, each bivariate resultant is computed by evaluation / interpolation and hence requires the computation of many univariate resultants. We illustrate this by counting the number of univariate resultants to perform and their degrees for the main step of the resolution (the part that reaches the peak complexity). We also measure the cost of such resultant computations using the NTL 10.5.0 and FLINT 2.5.2 libraries, both linked against GMP 6, when the base field is $\mathbb{F}_{2^{64}-59}$. These costs do not include the evaluation / interpolation steps which might also be problematic for large instances, because they are hard to parallelize.

ℓ	#res	Deg	Cost (NTL)	Cost (FLINT)
13	525M	16,000	1,850 days	735 days
29	12.8G	80,000	310,000 days	190,000 days

We were more successful with the direct approach using Gröbner bases that we now describe. For computing the kernel of a given endomorphism, we computed a Gröbner basis of the system (6.4) with some small modifications. First, we observe that the only occurrences of y_1 and y_2 are within the monomial $y_1 y_2$. Consequently, we can remove one variable by replacing each occurrence of $y_1 y_2$ by a fresh variable y. Next, we need to make the system 0-dimensional by encoding the fact that $d_3(x_3)$ and $\widetilde{d_3}(x_1, x_2)$ are nonzero. This is done by introducing another

fresh variable t and by adding the polynomial $S(x_1, x_2, x_3)t-1$ to the system, where $S(x_1, x_2, x_3)$ is the squarefree part of $d_3(x_3)\widetilde{d_3}(x_1, x_2)$. Finally, it appears that each polynomial is symmetric with respect to the transposition of the variables x_1 and x_2. Consequently, we can rewrite the equations using the symmetric polynomials $s_1 = x_1 + x_2$ and $s_2 = x_1\, x_2$. This divides by two the degree in x_1 and x_2 of the equations. We end-up with a system in 5 variables.

The whole construction can be slightly modified to compute the pre-image of a given divisor by the endomorphism: to model $\alpha(D) = Q - P_\infty$, we write $D = P_1 + P_2 + P_3 - 3P_\infty$ and solve for $\alpha(P_1 - P_\infty) + \alpha(P_2 - P_\infty) = Q - P_\infty - \alpha(P_3 - P_\infty)$. In that case, the variable y_3 gets involved in all the equations, so that we get a system in 6 variables.

For $\ell = 2$, the 2-torsion elements are easily deduced from the factorization of f, and by computing a pre-image of a 2-torsion divisor, we got a point in $J[4]$ from which we could deduce $a, b, c \bmod 4$. Dividing again by 2 was too costly, due to the fact that the 4-torsion point was in an extension of degree 4. For $\ell = 3$, which is an inert prime, we ran the kernel computation for the multiplication-by-3 endomorphism, without using the RM property. The norm being 27, this is the largest modular computation that we performed (and the most costly in terms of time and memory). The prime $\ell = 7$ ramifies in $\mathbb{Z}[\eta_7]$ as the cube of the ideal generated by $\alpha_7 = -2 - \eta_7 + \eta_7^2$. The kernel of α_7 can be computed but it yields only one linear relation in $a, b, c \bmod 7$. Dividing the kernel elements by α_7 would give more information, but again, this computation did not finish due to the field extension in which the divisors are defined. The first split prime is $\ell = 13$. We use the following small generators: $(13) = (2 - \eta_7 - 2\eta_7^2)(-2 + 2\eta_7 + \eta_7^2)(3 + \eta_7 - \eta_7^2)$, which seem to produce the polynomial systems with the smallest degrees. For instance, the apparently smaller element $1 + \eta_7^2$ of norm 13 yields equations of much higher degrees $7, 71, 72, 73, 72$. The next split prime is 29, which would maybe have been feasible, but was not necessary for our setting. In the following table, we summarize the data for these systems, that were obtained with Magma V2.23-4 on a Xeon E7-4850v3 at 2.20GHz, with 1.5 TB RAM.

mod ℓ^k	#var	degree of each eq.	time	memory	$a, b, c \bmod \ell^k$
2	—	—	—	—	$0, 0, 0$
4 (inert^2)	6	$7, 7, 14, 15, 15, 10$	1 min	negl.	$2, 2, 2$
3 (inert)	5	$7, 53, 54, 55, 26$	14 days	140 GB	$1, 2, 1$
$7 = \mathfrak{p}_1^3$	5	$7, 35, 36, 37, 36$	3.5h	6.6 GB	$a + 2b + 4c \equiv 2$
$13 = \mathfrak{p}_1\, \mathfrak{p}_2\, \mathfrak{p}_3$	5	$7, 44, 45, 46, 52$	3×3 days	41 GB	$12, 10, 9$
$29 = \mathfrak{p}_1\, \mathfrak{p}_2\, \mathfrak{p}_3$	5	$7, 92, 93, 94, 100$	$>3\times2$ weeks	>0.8 TB	—

6.4.2 Final collision search

The classical square-root-complexity search in genus 3 requires $O(q)$ group operations [42]. For RM curves, this can be improved by searching for the coefficients a, b, c of $\psi = \pi + \pi^\vee$ in $\mathbb{Z}[\eta]$. This readily yields a complexity in $O(q^{3/4})$, using the equation $aD + b\eta(D) + c\eta^2(D) = (q+1)D$, that must be satisfied for any rational divisor D. While a baby-step giant-step approach is immediate to design, it needs $O(q^{3/4})$ space and this is the bottleneck. A low-memory, parallel version of this search can be obtained with the algorithm of [61], where the details are given only for a 2-dimensional problem, while here this is a 3-dimensional problem. We explain below how we modified this algorithm to fit our needs. Just like in [61], including some anterior modular knowledge is straightforward: if a, b, c are known modulo m, the expected time is in $O(q^{3/4}/m^{3/2})$.

This time, the search was performed in a cuboid instead of a rectangle. Contrary to the

general genus-2 case, this time the cuboid is not flat since a, b and c have the same order of magnitude. Let us start by picking a random divisor D in J and set

$$K_D = [a \bmod m]D + [b \bmod m]\eta(D) + [c \bmod m]\eta^2(D).$$

As in Section 3.2.3, we look for a collision between two sets

$$T = \{s_1 mD + s_2 m\eta(D) + s_3 m\eta^2(D) \mid (s_1, s_2, s_3) \in [-B/m, B/m]^3\},$$

and

$$W = \{K_D + s_1 mD + s_2 m\eta(D) + s_3 m\eta^2(D) \mid (s_1, s_2, s_3) \in [-B/m, B/m]^3\}.$$

From the relations (6.1) between the coefficients of ψ and the coefficients of χ_π, one could translate the Weil bounds into precise bounds in the coefficients a, b and c. Instead, we set an *ad hoc* bound $B = 5\sqrt{q}$ for their respective absolute values. Our choice was satisfactory and we did not encounter any problem so we did not modify it, although fine tuning this parameter would certainly reduce the average running time.

Each chain consists in a pseudo-random deterministic walk in either W or T that stops whenever it encounters a distinguished point, which is the only information stored from each chain. Indeed, the deterministic nature of the process guarantees that any collision between two chains will propagate to their last point. This increases the running time compared to the baby-step giant-step approach but allows for negligible memory requirements, as explained in 3.2.3. While the probability p_D of being distinguished is an important parameter, the distinguishing feature itself is not. For instance, we say that an element is distinguished if the $\lfloor -\log_2 p_D \rfloor$ bits of low weight of its Mumford representation are equal to 0. By the birthday paradox, we expect a collision to be found after browsing through $(2B/m)^{3/2}$ points in the searchspace. Denoting by C the number of chains, we therefore expect each chain to be of length $(2B/m)^{3/2}/C$, and since each chain stops whenever it hits a distinguished element, p_D is precisely the inverse of this quantity. In our experiments, we set $p_D = 50000(B/m)^{-3/2}$, thus expecting the number of chains to be about 140000 before a collision occurs. Recall that the number of chains must be small enough to keep the memory requirements reasonable, but large enough to avoid taking too much time.

To design the deterministic walks, we start each chain by an element of either T or W defined by a triple (s_1, s_2, s_3) taken uniformly at random in $[-B/m, B/m]^3$. Then, given a divisor $\tilde{D}$ in a chain, the next one is computed as $\tilde{D} + O_{h(\tilde{D})}$, where the O's are a set of 120 precomputed offsets and h a hash function mapping $\tilde{D}$ into a triple $(b_1, b_2, i) \in \{0,1\}^2 \times \{1, 2, \ldots, 30\}$. The offset corresponding to that triple is

$$\alpha_i mD + (-1)^{b_1}\beta_i m\eta(D) + (-1)^{b_2}\gamma_i m\eta^2(D),$$

where α_i, β_i and γ_i are integers respectively taken uniformly at random in $\{1, 2, \ldots, 2L_i\}$ and then fixed during the whole search. The L_i's are chosen to reduce the risk of a chain exiting the cuboid and considering points on which collisions are impossible. This could actually even lead to a neverending chain which is why some bound can be set to discard any chain whose length is much longer than expected, but a convenient choice for the L_i's make this extremely unlikely. Our practical choice followed the genus-2 case and set the L_i's such that, on average, each chain terminates on a point whose coordinates are ten times smaller than the size of the cuboid. In the first direction, our offset is always positive and the expected length of a chain is $1/p_D$, so that the expected distance in the first direction is L_1/p_D and we choose $L_1 = 2Bp_D/10$. For the two other directions, the offsets have changing signs so that we bound the distance

reached using the central limit theorem. This yields an expected distance in $2\sqrt{2/3\pi}L_2/\sqrt{p_D}$ as in the genus-2 case. This is not surprising because since we study the problem dimension by dimension, we always consider one-dimensional random walks no matter the dimension of the searchspace. Approximating $2\sqrt{2/3\pi}$ by $9/10$ and dividing by 10, we choose L_2 and L_3 both equal to $2B\sqrt{p_D}/10$.

We wrote a dedicated C implementation with a few lines of assembly to speed-up the additions and multiplications in $\mathbb{F}_p$, taking advantage of the special form of p. This implementation performs $10.7M$ operations in the Jacobian per second using 32 (hyperthreaded) threads of a 16-core bi-Xeon E5-2650 at 2 GHz. We used the knowledge of ψ modulo 156 but not of the known relation modulo 7 for simplicity (there is no obstruction to using it and saving an additional $7^{1/2}$ factor).

After computing about 190,000 chains of average length 32,000,000, we got a collision, from which we deduced

$$\psi = 2551309006 + 2431319810\ \eta_7 - 847267802\ \eta_7^2,$$

and the coefficients of the characteristic polynomial χ_π of the Frobenius are then

$$\sigma_1 = 986268198,\quad \sigma_2 = 35389772484832465583,\quad \sigma_3 = 109560528621042368187702122 44.$$

The number of group operations that were done is slightly less than $43\,(p^{3/4}/156^{3/2})$. This factor 43 is close to the average that we observed in our numerous experiments with smaller sizes. Scaled on a single (physical) core, we can estimate the cost of this collision search to be 105 core-days.

Chapter 7

Counting points on hyperelliptic curves with explicit RM

In this chapter, we study the benefits of real multiplication in arbitrary genus. We extend the process of Chapter 6 and Section 3.1.2 for (families of) hyperelliptic curves with RM by an order $\mathbb{Z}[\eta]$. For primes ℓ that split into $\prod_{i=1}^{g} \mathfrak{p}_i$ in $\mathbb{Z}[\eta]$, we split $J[\ell]$ into a direct sum of g subspaces $J[\mathfrak{p}_i]$ isomorphic to $(\mathbb{Z}/\ell\mathbb{Z})^2$. One can therefore expect that for RM curves, Algorithm 6 detailed in Chapter 5 can be adapted to find non-zero elements of $J[\mathfrak{p}_i]$ instead of $J[\ell]$ with a complexity bound in $O_\eta\left((\log q)^{O(1)}\right)$ instead of $O_g\left((\log q)^{O(g)}\right)$. Note that we do not use the $O_g()$-notation because, as in Chapter 6, there is an additional dependency in η. Since g is nothing more than the degree of the algebraic number η, we replace the $O_g()$-notation by the $O_\eta()$-notation which takes into account both dependencies on g and η.

Using a theoretical machinery similar to that of Chapter 5, we will prove that it is indeed the case. However, we warn the reader that this complexity is still exponential in g. Even though each of the ideals $J[\alpha_i]$ has degree independent of g, we model them by polynomial systems whose multihomogeneous Bézout bounds involve a combinatorial factor of the form $\binom{g^2+g}{g}$. Since the complexity of the geometric resolution algorithm is quadratic in the multihomogeneous Bézout bound, this exponential factor also appears in the overall complexity of our algorithm.

Organization. In Section 7.1, we give an overview of our point-counting algorithm, along with an example of families of hyperelliptic curves of arbitrary high genus with RM by a real subfield of a cyclotomic field. In particular, we prove a bound on the size and number of primes ℓ to consider in our algorithm. Section 7.2 focuses on the main primitive of our algorithm: the computation of a non-zero element in the kernel of an endomorphism α whose degree is a small multiple of ℓ^2. This section adapts methods and results of Chapter 5 to design structured polynomial systems whose solution sets are subsets of $J[\alpha]$. Section 7.3 concludes on the complexity of solving these systems, and on the overall complexity of our point-counting algorithm. We also present an analysis on the exponent of g in the final complexity, investigating the various places where exponential factors may occur and how to avoid them when it is possible.

7.1 Overview

The main result of this chapter can be summarized by the following theorem, in which we give more precision on the notation $O_\eta(\log^c q)$ for our complexity result, and make the dependency

in η explicit. In Section 7.3, we also bound c by 8 and conjecture that it should be 6. Note that whenever we give a bound with an explicit constant, we can no longer hide the polylogarithmic factor in the exponent, so we use the notation $\widetilde{O}_\eta()$ to hide both factors depending only on η and factors that are polylogarithmic in q.

Theorem 7.1. *For any g and any $\eta \in \overline{\mathbb{Q}}$ such that $\mathbb{Q}(\eta)$ is a totally-real number field of degree g, there exists an explicitly computable $c(\eta) > 0$ such that there is an integer $q_0(g,\eta)$ such that for all prime power $q = p^n$ larger than $q_0(g,\eta)$ with $p \geq (\log q)^{c(\eta)}$ and for all genus-g hyperelliptic curves $\mathcal{C}$ with explicit RM by $\mathbb{Z}[\eta]$ defined over $\mathbb{F}_q$, the local zeta function of $\mathcal{C}$ can be computed with a probabilistic algorithm in expected time bounded by $(\log q)^{c(\eta)}$.*

7.1.1 Families of RM curves

We present one-dimensional families of hyperelliptic curves from [138], constructed via cyclotomic covers. They have an affine model $\mathcal{C}_{n,t} : Y^2 = D_n(X)+t$, where t is a parameter and D_n is the n-th Dickson polynomial with parameter 1 defined inductively by $D_0(X) = 2$, $D_1(X) = X$, and

$$D_n(X) = XD_{n-1}(X) - D_{n-2}(X).$$

Since $D_n(X)$ has degree n, setting $n = 2g + 1$ for odd n yields a one-dimensional family $\mathcal{C}_{n,t}$ of genus g hyperelliptic curves given by an odd-degree Weierstrass model. Their Jacobians all have an explicit endomorphism η, and when n is prime, Proposition 2 of [87] shows that $\mathbb{Z}[\eta] \cong \mathbb{Z}[\zeta_n + \zeta_n^{-1}]$, where ζ_n is a primitive n-th root of unity over $\mathbb{Q}$. Another family based on Artin-Schreier covering is detailed in the same paper but these curves have genus $(p-1)/2$ where p is the characteristic of the base field, so that our complexity study using the $O_\eta()$ notation would be pointless in that case. Since g becomes much larger than $\log p$ in that case, it would be more efficient to use p-adic algorithms anyway.

Let $\mathcal{C}$ be a hyperelliptic curve of genus in the family $\mathcal{C}_{2g+1,t}$. In [87], Kohel and Smith compute formulas for the Mumford form of $\eta((x,y) - P_\infty)$, where (x,y) is the generic point on $\mathcal{C}$. These formulas are given explicitly for some examples in genus 2 and 3, and an algorithm [87, Algorithm 5] is presented to compute them for any $\mathcal{C}$. This algorithm has a time complexity in $O(g^2)$ and requires to store $O(g^3)$ field elements. Thus, given a curve from that family as input, an explicit endomorphism of its Jacobian can be computed once and for all in $O(g^3 \log q)$ time and space complexity, which is negligible compared to the cost of counting points on the curve.

7.1.2 The characteristic equation

As in genus 3, let us consider $\psi = \pi + \pi^\vee$ and recall that $\psi \in \mathbb{Q}[\eta]$. We still have $\psi\pi = \pi^2 + q$ and once again, we test this equation to determine ψ instead of the characteristic equation of π. The link between ψ and π needs to be made explicit, which is the aim of the present section.

Since χ_π is a Weil polynomial, we can write $\chi_\pi(X) = \sum_{i=0}^{g}(-1)^i\sigma_i(X^{2g-i} + q^{g-i}X^i)$, with $\sigma_0 = 1$ and the convention that σ_g is actually twice smaller than the g-th coefficient of χ_π. By the Cayley-Hamilton theorem, we have $q^{-g}(\pi^\vee)^g\chi_\pi(\pi) = 0$. Using the fact that $\pi\pi^\vee = q$, we rewrite that as

$$\sum_{i=0}^{g}(-1)^{g-i}\sigma_{g-i}(\pi^i + (\pi^\vee)^i) = 0.$$

Our plan is to compute $\chi_\pi \bmod \ell$ by determining ψ . Let us write $\psi = \sum_{i=0}^{g-1} a_i\eta^i$, the goal of the section is to prove bounds on the coefficients a_i, so that we can estimate the number and maximal size of primes ℓ required to compute ψ without ambiguity. Note that ψ is in the

maximal order of $\mathbb{Q}(\eta)$, but not necessarily in $\mathbb{Z}[\eta]$. However, as in the genus-3 case, $\mathbb{Z}[\eta]$ has finite index Δ in the maximal order and the possible common denominator of the a_i's has to divide Δ. This denominator entails that additional primes may be required to fully determine ψ, however Δ depends only on η so that it will disappear in the O_η-notation of our complexity estimates. Therefore, we do not detail further this subtlety and assume for simplicity that the a_i's are integers, which we wish to bound by $O_\eta(\sqrt{q})$.

Let us first express the quantities $\pi^i + (\pi^\vee)^i$ in terms of powers of ψ as a first step towards expressing the σ_i's as functions of the a_i's.

Lemma 7.2. *For any $i \in \{1, \dots, g\}$, there exist integers $(\alpha_{i,j})_{0 \le j < i}$ such that $\alpha_{i,j} = O(q^{(i-j)/2})$ and*

$$\pi^i + (\pi^\vee)^i = \psi^i + \sum_{j=0}^{i-1} \alpha_{i,j} \psi^j.$$

Proof. The statement holds for $i = 1$ with $\alpha_{1,0}$ by the definition of ψ. For $i = 2$, we have $\psi^2 = \pi^2 + (\pi^\vee)^2 + 2\pi\pi^\vee$, so that we have the result with $\alpha_{2,0} = -2q$ and $\alpha_{2,1} = 0$.

In this proof, we set the convention $\alpha_{i,i} = 1$ to simplify our recurrence relations.

Let us now assume the lemma holds for any positive integer no greater than a certain i. We therefore have

$$\psi^{i+1} = (\pi + \pi^\vee)\psi^i = (\pi + \pi^\vee) \left[(\pi^i + (\pi^\vee)^i) - \sum_{j=0}^{i-1} \alpha_{i,j} \psi^j \right].$$

The first term is equal to $\pi^{i+1} + (\pi^\vee)^{i+1} + q(\pi^{i-1} + (\pi^\vee)^{i-1})$ so that we can use the lemma once again for $i - 1$ and get

$$\psi^{i+1} = \pi^{i+1} + (\pi^\vee)^{i+1} - \alpha_{i,i-1}\psi^i + q\alpha_{i-1,0} + \sum_{j=1}^{i-1} (q\alpha_{i-1,j} - \alpha_{i,j-1})\psi^j.$$

Thus, we have computed the $\alpha_{i+1,j}$ given by

$$\alpha_{i+1,j} = \begin{cases} \alpha_{i,i-1} & \text{if } j = i, \\ -q\alpha_{i-1,0} & \text{if } j = 0, \\ \alpha_{i,j-1} - q\alpha_{i-1,j} & \text{else.} \end{cases}$$

Let us now study the order of magnitude of the $\alpha_{i+1,j}$: from the recurrence hypothesis on both i and $i-1$, $\alpha_{i,i-1} = \alpha_{i+1,i}$ is in $O(\sqrt{q})$, $\alpha_{i-1,0}$ is in $O(q^{(i-1)/2})$ so that $\alpha_{i+1,0}$ is in $O(q^{(i+1)/2})$, and both $q\alpha_{i-1,j}$ and $\alpha_{i,j-1}$ are in $O(q^{(i+1-j)/2})$, which proves the result for any other $\alpha_{i+1,j}$. By induction, the lemma is proven. □

Note that our O-notation in the previous statement and proof can be a bit misleading as there may not be an absolute constant bounding all the $\alpha_{i,j}/q^{(i-j)/2}$. However, from the recurrence relation between the $a_{i,j}$'s, one sees that each $\alpha_{i,j}$ is equal to $q^{(i-j)/2}$ plus an error term that is in $O_\eta(q^{(i-j-1)/2})$ and at worst quadratic in g, hence the error term is negligible compared to $q^{(i-j)/2}$.

Proposition 7.3. *Let the a_i's be the coefficients of ψ in the basis $(1, \eta, \dots, \eta^{g-1})$ and σ_i be the i-th coefficient of χ_π, or half this coefficient if $i = g$. Then χ_π is uniquely determined by the a_i's and there exists $C_\eta > 0$ depending only on g and η such that for any $i \in \{0, \dots, g-1\}$, we have $|a_i| \le C_\eta \sqrt{q}$.*

Proof. Using Lemma 7.2 for any $i \in \{1, \ldots, g\}$ and setting $\alpha_{i,i} = 1$, we have

$$\sum_{i=0}^{g}(-1)^{g-i}\sigma_{g-i}\sum_{j=0}^{i}\alpha_{i,j}\psi^j = \sum_{j=0}^{g}\psi^j\sum_{i=j}^{g}(-1)^{g-i}\alpha_{i,j}\sigma_{g-i} = 0.$$

Let us define $\chi_\psi(X) = X^g + s_{g-1}X^{g-1} + \cdots + s_0$ with $s_i = \sum_{i=j}^{g}(-1)^{g-i}\alpha_{i,j}\sigma_{g-i}$. Invoking the Weil conjectures for the σ_{g-i}'s and Lemma 7.2 for the $\alpha_{i,j}$, one concludes that each s_i is in $O(q^{(g-i)/2})$. Furthermore, the expressions of the s_i's in terms of the σ_i's form a linear triangular system whose determinant equals 1, so that there is an efficiently computable one-to-one correspondence between χ_ψ and χ_π.

Let us now make explicit the link between the coordinates a_i of $\psi = \sum_{i=0}^{g-1} a_i\eta^i$ and the coefficients s_i of χ_ψ. For instance, $s_{g-1} = -\operatorname{Tr}(\psi) = -\sum_{i=0}^{g-1} a_i \operatorname{Tr}(\eta^i)$. To get the other relations, let us now order the g conjugates of η (possibly in the Galois-closure of $\mathbb{Q}(\eta)$), numbering them from η_1 to η_g, and proceed to the linear change of variables $\psi_k = \sum_{i=0}^{g-1} a_i\eta_k^i$ for any k in $\{1, \ldots, g\}$. The matrix associated to this linear transformation is the Vandermonde matrix of the conjugates η_k's. This matrix is invertible because η is separable so that the η_i are all distinct reals.

Note that χ_ψ is a degree-g monic polynomial vanishing on ψ, and it is therefore its characteristic polynomial. Since the ψ_k are exactly the real roots (possibly in the Galois-closure of $\mathbb{Q}(\eta)$) of χ_ψ, by Vieta's formula they satisfy the g equations

$$s_{g-i} = (-1)^i S_i(\psi_1, \ldots, \psi_g) \text{ for } 1 \le i \le g,$$

where the S_i's are the elementary symmetric polynomials in g variables. Thus, once the a_i's are known, the values for ψ and its conjugates are known and a unique value for each s_i is deduced. Furthermore, the Fujiwara bounds from [51] imply that for any $k \in \{1, \ldots, g\}$ we have

$$|\psi_k| \le 2 \max_{0 \le k \le g} \left(|s_{g-k}|^{1/k} \right).$$

We already know that $|s_{g-k}| = O(\sqrt{q}^k)$, so we deduce that the $|\psi_k|$ are in $O(\sqrt{q})$. Then, inverting the linear change of variable, we prove that the a_i are also in $O_\eta(\sqrt{q})$ since the matrix norm of the inverse of the Vandermonde matrix only depends on η. □

Our algorithm is based on determining the a_i's modulo ℓ for sufficiently many ℓ until they are known without ambiguity and we can deduce χ_π. While the Weil bounds on the σ_i's are enough for our purpose, we have proven that the a_i's are in $O_g(\sqrt{q})$ as in the genus-3 case. The next section details the process of recovering such modular information on the a_i's.

7.1.3 Overview of our algorithm

The general RM point counting algorithm is Algorithm 9. As mentioned above, we want to compute the coefficients $a_0, \ldots, a_{g-1}$ of the endomorphism ψ. More precisely, we compute their values modulo sufficiently many totally-split primes ℓ until we can deduce their values from the bounds of Prop 7.3 and the Chinese Remainder Theorem. Then, the coefficients of χ_π are deduced from the a_i's.

We now explain how the algorithm works for a given split ℓ. First its decomposition as a product of prime ideals $\ell\,\mathbb{Z}[\eta] = \mathfrak{p}_1 \cdots \mathfrak{p}_g$ is computed, and for each prime ideal $\mathfrak{p}_i$, a non-zero element α_i in $\mathfrak{p}_i$ is found with a small representation as in Lemma 7.4 below. In fact, $\mathfrak{p}_i$ is not necessarily principal and α_i need not generate $\mathfrak{p}_i$. The kernel of α_i is denoted by $J[\alpha_i]$ and it

input : q an odd prime power, and $f \in \mathbb{F}_q[X]$ a monic squarefree polynomial of degree $2g+1$ such that the curve $Y^2 = f(X)$ has explicit RM by $\mathbb{Z}[\eta]$.
output: The characteristic polynomial $\chi_\pi \in \mathbb{Z}[T]$ of the Frobenius endomorphism on the Jacobian J of the curve.

$w \leftarrow 1$;
Define C_g as in Prop. 7.3;
while $w \leq 2\,\Delta C_g\,\sqrt{q} + 1$ **do**
  Pick the next prime ℓ that satisfies conditions (C1) to (C4);
  Compute the ideal decomposition $\ell\,\mathbb{Z}[\eta] = \mathfrak{p}_1 \cdots \mathfrak{p}_g$, corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_g$ of η in $J[\ell]$;
  for $i \leftarrow 1$ **to** g **do**
    Compute a small element α_i of $\mathfrak{p}_i$ as in Lemma 7.4;
    Compute a non-zero element D_i of order ℓ in $J[\alpha_i]$;
    Find the unique $k_i \in \mathbb{Z}/\ell\mathbb{Z}$ such that $k_i\pi(D_i) = \pi^2(D_i) + qD_i$;
  end
  Find the unique tuple $(a_0, \ldots, a_{g-1})$ in $(\mathbb{Z}/\ell\mathbb{Z})^g$ such that $\sum_{j=0}^{g-1} a_j\lambda_i^j = k_i$, for i in $\{1, \ldots, g\}$;
  $w \leftarrow w \cdot \ell$;
end
Reconstruct $(a_0, \ldots, a_{g-1})$ using the Chinese Remainder Theorem ;
Deduce χ_π from ψ.

Algorithm 9: Overview of our RM point-counting algorithm

contains a subgroup G_i isomorphic to $\mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z}$, since the norm of α_i is a multiple of ℓ. The two-element representation $(\ell, \eta - \lambda_i)$ of the ideal $\mathfrak{p}_i$ implies that λ_i is an eigenvalue of η viewed as an endomorphism of $J[\ell] \cong (\mathbb{Z}/\ell\mathbb{Z})^{2g}$.

On $G_i \subset J[\alpha_i]$, the endomorphism η acts as the multiplication by λ_i. Therefore, the endomorphism $\psi = \sum_{i=0}^{g-1} a_i\eta^i$ also acts as a scalar multiplication on this 2-dimensional space, and we write $k_i \in \mathbb{Z}/\ell\mathbb{Z}$ the corresponding eigenvalue: for any D_i in G_i, we have $\psi(D_i) = k_iD_i$. On the other hand, from the definition of ψ, it follows that $\psi\pi = \pi^2 + q$. Therefore, if such a D_i is known, we can test which value of $k_i \in \mathbb{Z}/\ell\mathbb{Z}$ satisfies

$$k_i\pi(D_i) = \pi^2(D_i) + qD_i. \tag{7.1}$$

Since ℓ is a prime and D_i is of order exactly ℓ, this is also the case for $\pi(D_i)$. Finding k_i can then be seen as a discrete logarithm problem in the subgroup of order ℓ generated by $\pi(D_i)$; hence the solution is unique. Equating the two expressions for ψ, we get explicit relations between the a_j's modulo ℓ:

$$\sum_{j=0}^{g-1} a_j\lambda_i^j \equiv k_i \bmod \ell.$$

Therefore we have a linear system of g equations in g unknowns, the determinant of which is the Vandermonde determinant of the λ_i, which are distinct by hypothesis. Hence the system can be solved and it has a unique solution modulo ℓ.

It remains to show how to construct a divisor D_i in G_i, i.e. an element of order ℓ in the kernel $J[\alpha_i]$. Since an explicit expression of η as an endomorphism of the Jacobian of $\mathcal{C}$ is known, an explicit expression can be deduced for α_i, using the explicit group law. The coordinates of

the elements of this kernel are solutions of a polynomial system that can be directly derived from this expression of α_i, using a modelling similar to that of Chapter 5. Likewise, we use the geometric resolution algorithm to find the solutions of this system, perhaps in a finite extension of the base field, from which divisors in $J[\alpha_i]$ can be constructed. Multiplying by the appropriate cofactor, we can reach all the elements of G_i; but we stop as soon as we get a non-trivial one.

We summarize the conditions that must be satisfied by the primes ℓ that we work with:

(C1) ℓ must be different from the characteristic of the base field;

(C2) ℓ must be coprime to the discriminant of the minimal polynomial of η;

(C3) there must exist $\alpha_i \in \mathfrak{p}_i$ as in Lemma 7.4 below with norm non-divisible by ℓ^3 for $i \in \{1, \dots g\}$;

(C4) the ideal $\ell\,\mathbb{Z}[\eta]$ must split completely.

The first 3 conditions eliminate only a finite number of ℓ's that depends only on η. The condition (C3) implies that there is a unique subgroup G_i of order ℓ^2 in $J[\alpha_i]$.

Given a genus-g curve $\mathcal{C}$ with RM by $\mathbb{Z}[\eta]$, by Chebotarev's density theorem, the proportion of primes ℓ satisfying the last condition is at least $1/\#\operatorname{Gal}(\mathbb{Q}(\eta)/\mathbb{Q})$, which is bounded below by $1/(g!)$. To count points on $\mathcal{C}$, we need to find L a set of primes satisfying all the above conditions and such that $\prod_{\ell\in L}\ell > 2\Delta C_\eta\sqrt{q}$. By the prime number theorem, both the number and size of the primes contained in L are in $O((g!)\log(C_g q))$. In some particular cases, the proportion of "nice" primes may be much larger: for instance when the RM field is the totally real subfield of a cyclotomic field. In the field $Q(\zeta_n + \zeta_n^{-1})$, a prime ℓ totally splits if and only if $\ell \equiv \pm 1 \bmod n$, and therefore condition (C4) is satisfied by a proportion of primes equal to $2/(n-1) = 1/g$.

Lemma 7.4. *For any prime ℓ that splits completely in $\mathbb{Z}[\eta]$, each prime ideal $\mathfrak{p}$ above ℓ contains a non-zero element α of the form $\alpha = \sum_{i=0}^{g-1}\alpha_i\eta^i$, where the $|\alpha_i|$ are integers smaller than $\Delta^{1/g}\ell^{1/g}$, where Δ is the index $\left[\mathcal{O}_{\mathbb{Q}(\eta)} : \mathbb{Z}[\eta]\right]$.*

Proof. The coefficients of the elements of the ideal $\mathfrak{p}$ represented by polynomials in η form a lattice L of dimension g. In $\mathbb{Z}[\eta]$, its volume is the norm of $\mathfrak{p}$, i.e. ℓ. Thus, its actual volume in $\mathbb{R}^g$ is $\ell\Delta$. Let us consider $C = \{x \in \mathbb{R}^g \mid ||x||_\infty \leq \Delta^{1/g}\ell^{1/g}\}$. The volume of the convex C is $2^g\Delta\ell$. Since g is the dimension of L and $\Delta\ell$ its volume, Minkowski's theorem guarantees the existence of a non-zero element v of L belonging to C. By definition, $v = \sum_{i=0}^{g-1} v_i\eta^i$ is an element of $\mathfrak{p}$ whose coordinates v_i's are integers of absolute values bounded by $\Delta^{1/g}\ell^{1/g}$, which concludes the proof. □

Since we know it exists, given one of the ideals $\mathfrak{p}_i$, we can find α_i a small element of $\mathfrak{p}_i$ as in Lemma 7.4 by exhaustive search in at most $2^g\Delta\ell$ operations in $\mathbb{Z}[\eta]$. Note that there is an extensive litterature on finding short vectors in a lattice of dimension d, motivated for instance by cryptographic applications. An example is the quantum algorithm of [39] which computes a $2^{\widetilde{O}(\sqrt{d})}$-approximation of the shortest non-zero vector in time polynomial in d. Restricting to classical algorithms, the best option in general is the BKZ algorithm [126] that computes a $2^{\widetilde{O}(d^\alpha)}$-approximation in time $2^{\widetilde{O}(d^{1-\alpha})}$, for any $\alpha \in [0,1]$. In our case however, the existence of a very short vector is already known and, more importantly, the factor 2^g due to the dimension is acceptable since it vanishes in the O_η-notation.

7.2 Modelling kernels of endomorphisms

Let α be an explicit endomorphism of degree $O(\ell^2)$ on the Jacobian of $\mathcal{C}$, which satisfies the properties of Lemma 7.4. We want to compute a polynomial system that describes the kernel $J[\alpha]$ of α, and then solve it. The resultant-based approach of Chapter 6 cannot be used as the degrees are squared each time we eliminate a variable, causing an exponential dependency in g in the exponent of ℓ. Instead, we use the modelling techniques from Chapter 5, where the endomorphism α replaces the multiplication by ℓ. This time, the g variables of large degrees have degrees in $O_\eta(\ell^{3/g})$ instead of $O_\eta(\ell^3)$ so that the final complexity bound for computing the kernel α is in $O_\eta(\ell^c)$, with c an absolute constant.

The main change between this section and Sections 5.3 and 5.4 is that the d_i and e_i no longer denote ℓ-division but α-division polynomials, and the polynomials u_j and v_j intervening in the Mumford representation of the candidate kernel element are modified accordingly. The structure of our modelling is very similar but require some adaptations at various places, which is the reason why we repeat the analysis in the generic case. In the non-generic case, we restate the main results of Section 5.4 but only detail the parts requiring adjustments.

7.2.1 The generic case

Let us first recall the definition of Cantor's ℓ-division polynomials, the coefficients of the polynomials $\delta_\ell(X)$ and $\varepsilon_\ell(X)$ such that, for (x, y) a generic point of the curve and $\ell > g$, we have

$$\ell\left((x,y) - P_\infty\right) = \left\langle \delta_\ell\left(\frac{x-X}{4y^2}\right), \varepsilon_\ell\left(\frac{x-X}{4y^2}\right)\right\rangle.$$

We recall Theorem 4.13 proven in Chapter 4:

For any integer $\ell > g$, the polynomial $\delta_\ell(X)$ of degree g in X has coefficients in $\mathbb{F}_q[x]$ whose degrees in x are bounded by $g\ell^3/3 + O_g(\ell^2)$; the polynomial $\varepsilon_\ell(X)/y$ has coefficients in $\mathbb{F}_q(x)$ such that the degrees of the numerators and the denominators have degrees bounded by $2g\ell^3/3 + O_g(\ell^2)$. Furthermore, the roots of the denominators are roots of the leading coefficient of $\delta_\ell(X)$.

These polynomials describe the multiplication by ℓ, but for our purpose we need to define the α-division polynomials d_i and e_i such that, denoting by $P = (x, y)$ the generic point of $\mathcal{C}$, the non-normalized Mumford form of $\alpha(P - P_\infty)$ is equal to

$$\left\langle \sum_{i=0}^{g} d_i(x)X^i, y\sum_{i=0}^{g-1}\frac{e_i(x)}{e_g(x)}X^i\right\rangle.$$

By Lemma 7.4, we know that $\alpha = \sum_{i=0}^{g-1}\alpha_i\eta^i$ with $|\alpha_i| = O_\eta(\ell^{1/g})$. Since the degrees of the $\eta^i(P - P_\infty)$ do not depend on ℓ, by Theorem 4.13 applied to Cantor's α_i-division polynomials we prove that the degrees of the d_i's and e_i's are in $O_\eta(\ell^{3/g})$.

Definition 7.5. *In what follows, we will say that an element of J is α-generic if it has weight g and the corresponding reduced divisor $\sum_{i=1}^{g}(P_i - P_\infty)$ satisfies the following two properties:*

- *For any i, the u-coordinate of the divisor $\alpha(P_i - P_\infty)$ in Mumford form has degree g;*
- *For any $i \neq j$, the u-coordinates of the divisors $\alpha(P_i - P_\infty)$ and $\alpha(P_j - P_\infty)$ are coprime.*

This implies that if an affine point P occurs in the support of a $\alpha(P_i - P_\infty)$ then neither P nor $-P$ appears in the support of another $\alpha(P_j - P_\infty)$.

Let $D = \sum_{i=1}^{g}(P_i - P_\infty)$ be an α-generic divisor in J. We shall consider a system equivalent to $\alpha(D) = 0$ but let us first introduce some notation. For each point $P_i = (x_i, y_i)$ in the support of D, we denote $\langle u_i, v_i\rangle$ the Mumford form of $\alpha(P_i - P_\infty)$ and $(a_{ij}, b_{ij})_{1\le j\le g}$ the coordinates of the g points in its support counted with multiplicities, which means that for any i the g roots of u_i are exactly the a_{ij}, and that for any j, $b_{ij} = v_i(a_{ij})$.

Proposition 7.6. *We can model the set of generic α-division elements as the solution set of a bihomogeneous polynomial system consisting of $O(g^2)$ equations in $\mathbb{F}_q[X_1, \ldots, X_g, Y_1, \ldots, Y_{n_y}]$ such that $n_y = O(g^2)$ and the degrees in the X_i's and Y_j's are respectively in $O_\eta(\ell^{3/g})$ and $O_\eta(1)$.*

Proof. Following the modelling of Section 5.3, we have $\alpha(D) = 0$ if and only if the sum of the divisors $\sum_{i=1}^{g}\alpha(P_i - P_\infty)$ is a principal divisor. The only pole is at infinity, so this is equivalent to the existence of a non-zero function $\varphi \in \mathbb{F}_q(\mathcal{C})$ of the form $P(X) + YQ(X)$ with P and Q two polynomials such that the g^2 points (a_{ij}, b_{ij}) are the zeros of φ, with multiplicities. Since we want φ to have g^2 affine points of intersection with the curve $\mathcal{C}$ (once again, counted with multiplicities), the polynomial $\mathrm{Res}_Y(Y^2 - f, P + YQ) = P^2 - fQ^2$ must have degree g^2 which yields $2\deg(P) \le g^2$ and $2\deg(Q) \le g^2 - 2g - 1$. Exactly one of those two bounds is even (it depends on the parity of g), and for this particular bound, the inequality must be an equality, otherwise the degree of the resultant would not be g^2. Since the function φ is defined up to a multiplicative constant, we can normalize it so that the polynomial $P^2 + fQ^2$ is monic, which is equivalent to enforce that either P or Q is monic depending on the parity of g.

For a fixed $i \in [1, g]$, requiring the (a_{ij}, b_{ij}) to be zeros of φ amounts to asking for the a_{ij} to be roots of $P(X) + Q(X)v_i(X)$, with multiplicities. Since the a_{ij} are by definition the roots of the u_i, $\alpha(D) = 0$ is equivalent to g congruence relations $P + Qv_i \equiv 0 \bmod u_i$. Thus, for any α-generic divisor, $\alpha(D) = 0$ is equivalent to the existence of P and Q satisfying the above g congruence relations.

The variables are the coefficients of P and Q, as well as the x_i and y_i. With the degree conditions and the normalization, we have $g^2 - g$ variables coming from P and Q. Adding the $2g$ variables x_i and y_i, we get a total of $g^2 + g$ variables. Each one of the g congruence relations amounts to g equations providing a total of g^2 conditions on the coefficients of P and Q. The fact that the (x_i, y_i) are points of the curve yields the g additional equations $y_i^2 = f(x_i)$. Finally, we have to enforce the α-genericity of the solutions, which can be done by requiring that $\prod_i d_g(x_i)e_g(x_i)\prod_{i<j}\mathrm{Res}(u_i, u_j) \neq 0$. Note that we do not extend Theorem 4.13 but instead add the non-vanishing condition for the denominator of the v-coordinate of $\alpha(D)$. Still, we get a polynomial system with $g^2 + g$ equations in $g^2 + g$ variables, together with an inequality.

We now estimate the degrees to which the variables occur in the equations. Each congruence relation is obtained by reducing $P+Qv_i$, which is a polynomial of degree $O(g^2)$ in X, by u_i which is of degree g. We can do it by repeatedly replacing X^g by $-\sum_{j<g}(d_j(x_i)/d_g(x_i))X^j$, which we will have to do at most $O(g^2)$ times. Since the d_j have degree in $O_\eta(\ell^{3/g})$ in x_i, the fully reduced polynomial will have coefficients that are fractions for which the degrees of the numerators and of the denominators are at most $O_\eta(\ell^{3/g})$ in the x_i variables. In these equations, the degree in the y_i variables and in the variables for the coefficients of P and Q is 1. The degrees in x_i and y_i in the curve equations are $2g + 1$ and 2 respectively.

It remains to study the degree of the inequality. Each resultant is the determinant of a $2g\times 2g$ Sylvester matrix whose coefficients are the d_i, which have degrees bounded by $O_\eta(\ell^{3/g})$. Since for any i there are exactly g resultants involving x_i in the product, the degree of this inequality in any x_i is in $O_\eta(\ell^{3/g})$, and it does not involve the other variables. In order to be able to use Proposition 5.3, we must model this inequality by an equation, which is done classically by introducing a new variable T and by using the equation $T \cdot \prod_i d_g(x_i)e_g(x_i)\prod_{i<j}\mathrm{Res}(u_i, u_j) = 1$.

To conclude, we have a polynomial system with two blocks of variables: the g variables x_i on the one hand and the $g^2 - g$ variables coming from the coefficients of P and Q, along with the g variables y_i on the other hand. The degree of the equations in the first block of variables grows cubically in $\ell^{1/g}$, while the degree in the other block of variables depends only on g (and η). □

7.2.2 Non-generic kernel elements

As in Section 5.3, apart from the neutral element, we expect to capture the whole kernel of the endomorphism α by using the modelling of Section 7.2.1. Contrary to Chapter 5, Algorithm 9 does not require us to find a basis of $J[\alpha]$ because the determination of the k_i's does only require a single non-zero element in each $J[\alpha_i]$. Thus, a study of non-generic elements in $J[\alpha]$ is necessary only if there is no α-generic element in $J[\alpha]$. Such a case happens if and only if the polynomial $\prod_{i=1}^{g} d_g(x_i)e_g(x_i)\prod_{i\neq j}\mathrm{Res}(u_i, u_j)$ in the variables $x_1, \ldots, x_g$ vanishes on $J[\alpha]$. It seems very unlikely that the whole set $J[\alpha]$ lives in such a hypersurface, and if it happens, one can discard the ℓ for which we fail to find an α-generic element. Although it seems even more unlikely that this situation could happen for sufficiently many ℓ so as to threaten the validity of our complexity bound, we are far from a proven statement and do not exclude it might be possible to design a highly non-generic curve providing a counterexample.

Therefore, we follow the non-genericity analysis of Section 5.4 except that we consider u_i and v_i defined as the Mumford form of $\alpha(P_i - P_\infty)$ instead of $\ell(P_i - P_\infty)$. Let us briefly review the non-generic situations that one can encounter, following Section 5.4.1 and keeping the same numbering.

Case 1: Modelling a kernel element of weight $w < g$. We write $D = \sum_{i=1}^{w}(P_i - P_\infty)$ and look for a $\varphi = P(X) + YQ(X)$ vanishing at each point of each reduced divisor $\alpha(P_i - P_\infty)$. This is similar to the Case 1 of Section 5.4.1.

Case 2: Modelling a kernel element with multiple points. It may happen that the element we are looking for is $D = \sum_{i=1}^{w}(P_i - P_\infty)$ but not all the P_i's are distinct. In that case, we rewrite it as $D = \sum_{j=1}^{s}\lambda_j(P_j - P_\infty)$ such that the P_j's are distinct and look for a $\varphi = P(X) + YQ(X)$ vanishing at each point of each reduced divisor $\lambda_j\alpha(P_j - P_\infty)$. Apart from the modification of u_i and v_i, the modelling is identical to that of Chapter 5.

Case 4: Modelling a kernel element after reduction. Even if all the $\alpha(P_i - P_\infty)$ had full weight, there still may be less than g^2 points in the union of their supports due to possible cancellations of points appearing in the supports of several $\alpha(P_i - P_\infty)$ with different signs. Exactly as in Section 5.4.1, if P appears within $\alpha(P_i - P_\infty)$ and $\alpha(P_j - P_\infty)$ with respective multiplicities ν_i and ν_j of opposite signs, this is modelled by ensuring that the corresponding u_i, u_j, and $v_i + v_j$ share a common factor $(X - \xi)^\nu$ where $\nu = \max(|\nu_i|, |\nu_j|)$. In that case, we look for $\varphi(X, Y) = (X - \xi)^\nu(\widetilde{P}(X) + Y\widetilde{Q}(X))$, with $\widetilde{P}$ coprime to $\widetilde{Q}$. Once modified the values of the u_i and v_i, nothing changes from Chapter 5.

Case 5: Modelling a kernel element with multiplicity. Conversely, $\alpha(P_i - P_\infty)$ and $\alpha(P_j - P_\infty)$ can also share the same point with multiplicities of identical sign, leading to multiplicities in the reduced divisor $\alpha(D)$. Similarly to what was done in the Case 5 of Section 5.4.1, we can group the corresponding u_i, u_j, v_i and v_j in polynomials U and V such that $U|V^2 - f$

and $\deg V < \deg U$, and then look for $\varphi = P(X) + YQ(X)$ such that $P + QV \equiv 0 \bmod U$. Once again, nothing changes apart from the definition of the u_i's and v_i's.

Case 3: Low weight after applying α. We kept this case for the end because it is not a straightforward extension of the Case 3 appearing in Section 5.4.1. Until now, we assumed that all the P_i's in the support of D were such that $\alpha(P_i - P_\infty)$ had weight g, i.e. $d_g(x_i) \neq 0$. We now want to model the case where $D = \sum_{i=1}^{w}(P_i - P_\infty)$ such that each $\alpha(P_i - P_\infty)$ has weight w_i. In Chapter 5, this was done using a result from [28] giving a necessary and sufficient condition for $\ell(P_i - P_\infty)$ to be of weight w_i. When α is an endomorphism other than scalar multiplication, no such result holds a priori. In what follows, we solve this issue by designing non-generic α-division polynomials $\Gamma_{\alpha,t}$ and $\Delta_{\alpha,t}$ such that $\alpha\left((x,y) - P_\infty\right)$ has weight w if and only if $\Delta_{\alpha,w}(x) = 0$ and $\Gamma_{\alpha,w-1}(x) \neq 0$.

Combining all degeneracies. As in Section 5.4.2, we have to consider situations in which several of the previous cases occur simultaneously. Note that while we wanted to compute the whole ℓ-torsion in Chapter 5, we now only need one kernel element per endomorphism α_i to determine $\chi_\pi \bmod \ell$. Therefore, after finding a non-zero solution to any of the subsequent systems, one need not consider the others. Once again, we will not perform a complete analysis as in Section 5.4.2 but rather detail when modifying the values of u_i and v_i is not sufficient. We also update the analysis on the numbers and degrees of equations and variables. The aim of the Section is to prove the following proposition.

Proposition 7.7. *We can model the set of non-generic elements of $J[\alpha]$ as the solution set of $O_\eta(1)$ bihomogeneous polynomial systems each consisting of $O(g^2)$ equations in $\mathbb{F}_q[X_1, \ldots, X_g, Y_1, \ldots, Y_{n_y}]$ such that $n_y = O(g^2)$ and the degrees in the X_i's and Y_j's are respectively in $O_\eta(\ell^{3/g})$ and $O_\eta(1)$.*

Proof. We similarly encode each situation by a non-genericity tuple $(w, \lambda, \tau, \varepsilon, M)$ in the sense of Definition 5.8, and derive an associated polynomial system whose solution set corresponds to elements $D \in J[\alpha]$ such that:

- the reduced divisor D of weight w has the form $\sum_{i=1}^{k} \lambda_i P_i$ with distinct P_i's,
- each $\lambda_i \alpha(P_i - P_\infty)$ has weight τ_i,
- each ε_i is in $\{0, 1\}$ and such that $\varepsilon_i = 1$ if and only if $\tau_i = \lambda_i = 1$.
- the $k \times s$ matrix M represents the points shared by the $\lambda_i \alpha(P_i - P_\infty)$ as in Section 5.4.2, with $s \leq gk$.

We can follow the analysis of Section 5.4.2 to describe more explicitly the equations and their degrees / number of variables, and remark that the only part that does not generalize readily is the definition of non-generic α-division polynomials, as in the Case 3 above. Let us first fix this issue.

When the weight t_i of $\lambda_i \alpha(P_i - P_\infty)$ is strictly smaller than g, the usual coordinate system given by the Mumford form is no longer available, due to the vanishing of the denominator $e_g(x_i)$. We define an adequate coordinate system to describe non-generic elements of weight t. Let us consider the variety

$$V_{\alpha,t} = \{(x,y) \in \mathcal{C} \mid \alpha\left((x,y) - P_\infty\right) \text{ has weight } t\}.$$

We want to define polynomials $\Delta_{\alpha,t}$ and $\Gamma_{\alpha,t}$ such that a point is in $V_{\alpha,w}$ if and only if $\Delta_{\alpha,w}(x) = 0$ and $\Gamma_{\alpha,w-1}(x) \neq 0$ iteratively. First, $\Delta_{\alpha,g-1} = \mathrm{GCD}(d_g, e_g)$, so that the points (x, y) of $V_{\alpha,g-1}$ satisfy $\Delta_{\alpha,g-1}(x, y) = 0$. Assuming that for $k < g$ we have already constructed a squarefree polynomial $\Delta_{\alpha,k}$ vanishing on the abscissae of points in $V_{\alpha,k}$, then one can compute $\alpha\left((x, y) - P_\infty\right)$ over $\mathbb{F}_p[x, y]/(\Delta_{\alpha,k}(x), y^2 - f(x))$. By our recurrence hypothesis, the Mumford form of the result is $\langle u, v\rangle$, with u of degree k and v of degree $k-1$. Let $\Gamma_{\alpha,k-1}$ be the product of $\mathrm{LC}(u)$ with the denominator of $LC(v)$, then $V_{\alpha,k}$ is the set of points (x, y) such that $\Delta_{\alpha,k}(x) = 0$ and $\Gamma_{\alpha,k-1}(x) \neq 0$. Furthermore, $\Delta_{\alpha,k-1} = \mathrm{GCD}(\Delta_{\alpha,k}, \Gamma_{\alpha,k-1})$ vanishes on the points of $V_{\alpha,k-1}$.

To avoid multiplicities, we replace $\Delta_{\alpha,t}(x)$ by the square-free polynomial whose roots are exactly the roots of $\Delta_{\alpha,t}(x)$ that are not roots of $\Gamma_{\alpha,t-1}(x)$ when it is necessary. Note that the degrees of the Δ and Γ are by construction bounded by $\deg \Delta_{\alpha,g-1} \leq \deg d_g$ with $\deg d_g$ itself bounded by $O_\eta(\ell^{1/g})$. This way, we state an analogue of Definition 5.9 for non-generic α-division polynomials:

Definition 7.8. *The non-generic α-division polynomials $\mathfrak{u}_{\alpha,t}$ and $\mathfrak{v}_{\alpha,t}$ are the polynomials in X with coefficients in $\mathbb{F}_p[x, y]/(\Delta_{\alpha,t}(x), y^2 - f(x))$ such that*

$$\alpha((x, y) - \infty) = \Big\langle \mathfrak{u}_{\alpha,t}(X), \mathfrak{v}_{\alpha,t}(X) \Big\rangle,$$

in weight-t Mumford representation: $\mathfrak{u}_{\alpha,t}(X)$ is monic of degree t, $\mathfrak{v}_{\ell,t}(X)$ is of degree at most $t-1$ and they satisfy $\mathfrak{u}_{\alpha,t} \mid \mathfrak{v}_{\alpha,t}^2 - f$.

All the equations associated to a non-genericity tuple $(w, \lambda, t, \epsilon, M)$ are merely identical to those of Section 5.4.2 except that the d_i, e_i and have different definitions and that $\Delta_{\alpha,t}$ replaces $\tilde{\Delta}_{\ell,t}$ so that Equation (Sys.3) now reads

$$\begin{cases} \Delta_{\lambda_i\alpha,t_i}(x_i) = 0, \\ \Gamma_{\lambda_i\alpha,t_i-1}(x_i) \neq 0, \end{cases} \quad \text{for all } i \text{ in } [1, k] \text{ such that } t_i < g. \tag{Sys.3b}$$

While turning the systems describing $J[\ell]$ into systems describing $J[\alpha]$, we did not add any variable, so that the study of Section 5.4.2 presented in Table 5.1 is still valid and we just recall that the total number of variables is bounded by $4g^2 + g$.

As for the number of equations and their respective degrees, the only change comes from the fact that the coefficients of the u_i and v_i have degrees in the x_i bounded by $O_\eta(\ell^{3/g})$ instead of $O_\eta(\ell^3)$, and Table 5.2 becomes Table 7.1.

Table 7.1 shows that any system corresponding to a non-genericity tuple satisfies the degree conditions of Proposition 7.7. As in the non-RM case, the number of such tuples is bounded by $g^{O(g^3)}$ and Proposition 7.7 is proved. □

7.3 Complexity analysis

Now that we have modelled subsets of $J[\alpha]$ by polynomial systems whose size in terms of equations, variables and degrees have been carefully bounded, we apply the geometric resolution algorithm and bound its complexity using analogues of Proposition 5.3.

7.3.1 Solving the polynomial systems modelling $J[\alpha]$

Proposition 7.9. *For any $\varepsilon > 0$, there is a constant D such that for any endomorphism $\alpha \in \mathbb{Z}[\eta]$ of norm a multiple of $\ell > g$ coprime to the base field characteristic, there is a Monte*

Equations reference	Number of equations (and bound)	$\deg_1$	$\deg_2$
Eq. and Ineq. Sys.1	$2k \leq 2g$	$2g+1$	0
InEq. Sys.2	$k(k-1)/2 \leq g(g-1)/2$	1	0
Eq. and Ineq. Sys.3b	$\leq 2g$	$O_\eta(\ell^{3/g})$	0
InEq. Sys.4	$\leq g$	$O_\eta(\ell^{3/g})$	0
Eq. Sys.5	$\sum_{i=1}^{k}\sum_{j=1}^{s} \lvert m_{ij}\rvert \leq g^4$	$O_\eta(\ell^{3/g})$	$\leq g$
InEq. Sys.6	$ks \leq g^3$	$O_\eta(\ell^{3/g})$	$\leq g$
Eq. Sys.7 and Sys.8	$\leq k^2 s \leq g^4$	$O_\eta(\ell^{3/g})$	$\leq g$
InEq. Sys.9	$\leq s^2 \leq g^4$	0	1
Eq. Sys.10	$\sum_{i=1}^{k} t_i \leq g^2$	$O_\eta(\ell^{3/g})$	$\leq g$
Eq. Sys.11	$\deg U \leq g^2$	0	$O(g^3)$
Eq. Sys.12	$\sum_{i=1}^{k} \deg \widetilde{u}_i \leq g^2$	$O_\eta(\ell^{3/g})$	$O(g^2)$
Eq. Sys.13	$\deg U \leq g^2$	0	$O(g^3)$

Table 7.1: Summary of the degrees of the equations in the polynomial system corresponding to a normalized non-genericity tuple $(w, \lambda, t, \epsilon, M)$.

Carlo algorithm which computes an $\mathbb{F}_{q^e}$-geometric resolution of the sub-variety of $J[\alpha]$ consisting of α-generic α-torsion elements, where $e = O_\eta(\log \ell)$. The time and space complexities of this algorithm are bounded by $O_\eta(\ell^D (\log q)^{2+\varepsilon})$ and it returns the correct result with probability at least $5/6$.

Proof. Let us consider the sub-variety $S \subset J[\alpha]$ consisting of α-generic elements, and I the corresponding ideal. More precisely, we see I as the ideal of a sub-scheme of the scheme $J[\alpha]$, itself subscheme of $J[\deg \alpha]$, which is the kernel of a finite and étale map because $\deg \alpha$ is a small multiple of ℓ and is hence coprime to the characteristic p thanks to our assumptions on the size of p in the statement of Theorem 7.1.

Therefore, I is 0-dimensional and radical. Since all the elements in S have the same weight g we can use the Mumford coordinates $\langle u(X), v(X)\rangle$ with $\deg u = g$ and $\deg v < g-1$ as a local system of coordinates to represent them. But the polynomial system that we have built is with the (x_i, y_i) coordinates, that is, it generates the ideal I^{unsym} obtained by adjoining to the equations defining I the $2g$ equations coming from $u(X) = \prod(X - x_i)$ and $y_i = v(x_i)$. Then we have $\deg I^{\text{unsym}} = g! \deg I$. By the α-genericity condition, all the fibers in the variety have exactly $g!$ distinct points corresponding to permuting the (x_i, y_i) which are all distinct. Therefore the radicality of I implies the radicality of I^{unsym} and we can apply a modified version of Proposition 5.3 to our polynomial system.

Indeed, by Proposition 7.6 we now have a function h and a constant C such that $d_x \leq h(g)\ell^{C/g}$ instead of $h(g)\ell^C$. This propagates in the proof of Proposition 5.3, and since the power of ℓ only comes from the bound on d_x, we can also replace ℓ by $\ell^{1/g}$ in the final result, so that we can compute an $\mathbb{F}_{q^e}$-geometric resolution of S in time and space bounded by $O_\eta(\ell^D (\log q)^{2+\varepsilon})$, with $e = O_\eta(\log \ell)$. □

Following the same proof but invoking Proposition 7.7 instead of Proposition 7.6, the same complexity bound holds for solving the polynomial system associated to any non-genericity tuple. Even if a non-zero α-torsion element is only found after solving all the systems associated to non-genericity tuples, the cost for computing $\psi \bmod \ell$ is only multiplied by a factor in $O_\eta(1)$.

We have proven that there exists a constant c such that for any prime ℓ satisfying conditions (C1) to (C4), computing $\chi_\pi \bmod \ell$ is achieved within $O_\eta(\ell^c)$ field operations. Taking into account

the size of the largest ℓ to consider and the cost of field operations, the overall complexity of our point-counting algorithm is in $O_\eta((\log q)^{c+2})$. The bottleneck is computing geometric resolutions of polynomial systems which is quadratic in their respective multihomogeneous Bézout bounds, up to a factor in $O_\eta(1)$. Still neglecting factors in $O_\eta(1)$, the multihomogeneous Bézout bound itself boils down to $O_\eta(\deg_1^g)$ by Definition 2.44. As shown in Table 7.1, $\deg_1 = O_\eta(\ell^{3/g})$ so we deduce that $c = 6$ and get an overall complexity bound in $\widetilde{O}_\eta(\log^8 q)$.

Note that our bound on $\deg_1$ is pessimistic because we used the proven cubic bound for the degrees of Cantor's division polynomials while we expect them to be actually quadratic. Under this assumption, $\deg_1$ is reduced to $O_\eta(\ell^{2/g})$ and the overall complexity would therefore be in $\widetilde{O}_\eta(\log^6 q)$ for any g. Apart from the part depending on g, this conjectural result is identical to what we proved for genus 3. In the next section, we push the analysis forward by investigating the dependency on g.

7.3.2 Dependency on g of the complexity

The goal of this section is to assess the potential of our algorithm to achieve a polynomial-time complexity both in g and $\log q$ on some family of curves. To this end, we review our complexity analysis with additional attention given to the factors that previously vanished in the O_η.

Dependency on g of the largest ℓ Let us first come back to the constant C_g of Section 7.1.2. We have seen that the only non-polynomial dependency on g came from the matrix norm when inverting the linear change of variables $\psi_k = \sum_{i=0}^{g-1} a_i \eta_k^i$, which is described by the Vandermonde matrix of the g conjugates of η, denoted by η_k for $k \in \{1, \dots, g\}$. Let B the inverse of this matrix, then we have

$$B_{ij} = \frac{\sum_{\substack{1 \le k_1 < \dots < k_{g-j} < g \\ k_1, \dots, k_{g-j} \neq i}} (-1)^{j-1} \eta_{k_1} \cdots \eta_{k_{g-j}}}{\eta_i \prod_{k \neq i} (\eta_k - \eta_i)}.$$

Let $E = \max_k(|\eta_1|, \dots, |\eta_k|)$, $e = 1/\min_k(|\eta_1|, \dots, |\eta_k|)$, and $D = \max_{i \neq j}\left(|\eta_i - \eta_j|^{-1}\right)$, then we can bound the absolute value of any entry of B very roughly either by $ge(2ED)^g$ or by ge if $2ED \le 1$, and the matrix-norm of B is bounded by g times this previous bound. Note that the factor Δ is also a nuisance but it is bounded by the discriminant of $\mathbb{Z}[\eta]$. This discriminant is in turn bounded by $\max_{i \neq j}\left(|\eta_i - \eta_j|\right)^{2g}$. Thus, the constant C_g can be bounded by $g^2 c^g$, where c has a polynomial dependency on η and its conjugates.

By the prime number theorem, the set L of primes such that $\prod_{\ell \in L} \ell > 2C_g\sqrt{q}$ is such that the number and size of primes in L is in $\widetilde{O}(g) \log q / \log\log q$. As we already mentioned, the primes to consider must satisfy the conditions $(C1)$ to $(C4)$ and that may cause them to be larger by a factor depending exponentially on g *a priori*. Since the complexity of computing $\chi_\pi \bmod \ell$ is polynomial in ℓ, this implies that the overall complexity depends exponentially on g in general.

However, a curve in the family $\mathcal{C}_{n,t}$ introduced in Section 7.1.1 has RM by the real subfield of $\mathbb{Q}(\zeta_n)$, for which we know that the proportion of split primes is $2/(n-1) = 1/g$. Therefore, this first obstacle due to the size of primes to consider can be overcome provided that we further strengthen the assumptions on the RM-curves we consider.

Finding small elements in lattices This time, the exhaustive search is no longer sufficient for our need because of the factor 2^g in the size of the ball $\left\{v \mid ||v||_\infty \le \Delta^{1/g} \ell^{1/g}\right\}$. Unfortunately, the current best known algorithms for finding short vectors in time subexponential in

the dimension of the lattice have a drawback that makes them unusable in our point-counting algorithm. Indeed, although they run faster than the naive approach, they do not necessarily output the shortest non-zero vector in the lattice, but an approximation that may be greater by a factor which is also subexponential in the dimension. The size of the short vector plays a prominent role in the complexity analysis of our point-counting algorithm as it gives a bound on the degrees of the equations modelling $J[\alpha]$. Even if we find an α whose coordinates are in $g\ell^{1/g}$ instead of $c\ell^{1/g}$, the factor g will cause a factor g^g in the multihomogeneous Bézout bound, and hence in the final complexity of solving the polynomial systems.

Although finding short generators of ideals in number fields is believed to be hard in general, we may still expect to further restrict the RM curves we consider so as to fall in a case for which the complexity of such task becomes affordable. Examples are given in [13], where a classical algorithm is shown to compute short generators of principal ideals in particular number fields called multiquadratics, i.e. fields of the form $\mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$. While we acknowledge that it is quite speculative to hope for families of curves of arbitrary high genus with RM by a $\mathbb{Z}[\eta]$ satisfying all the previous hypotheses, we do not linger on this because the next point is much more of a concern anyway.

Solving polynomial systems Using the strategy of Section 7.2, the complexity is polynomial in the multihomogeneous Bézout bound, itself including a combinatorial factor in g^{g^2}. Indeed, although the ideals of α-torsion have degree ℓ^2 independent of g, this is not true for the number of variables involved in our modelling, which is at least g^2 in the generic case. Worse than that, the size of the polynomial systems modelling the set of generic α-torsion elements is already exponential in g. Indeed, following the proof of Lemma 2.50, one sees that the number of monomials has a factor $\binom{n_x+d_x}{n_x}$ and our modelling is such that $n_x = g$ and $d_x \geq g\ell^{2/g}$ so that $\binom{n_x+d_x}{n_x} \geq (n_x + d_x)^{n_x}/n_x^{n_x}$ is bounded below by g^g.

Thus, there is no hope of turning our algorithm into something subexponential in g in its current state. Possible workarounds could be looking for easier instances in which we could model the α-torsion by even smaller polynomial systems, or cases for which there are simpler ways of obtaining a generic α-torsion divisor than the one we used.

Conclusion

In this thesis, we focus on point-counting on hyperelliptic curves over finite fields using methods derived from Schoof and Pila's algorithms. We have studied the asymptotic complexity of this task for curves of arbitrary genus defined over a sufficiently large field. In particular, the power of $\log q$ in the complexity has been reduced from $O(g^2 \log g)$ to $O(g)$ in Chapter 5. For families of curves equipped with an explict RM, we have further reduced this power to a constant in Chapter 7, and proved that our algorithm computes the zeta function of genus-g hyperelliptic curves with explicit RM in time bounded by $\widetilde{O}_\eta(\log^8 q)$. Conjecturally, we actually expect this complexity to be in $\widetilde{O}_\eta(\log^6 q)$.

Instanciating our general methods in small genus

It is natural to wonder whether the algorithm we described in Chapter 5 to establish those complexity bounds, or rather their instantiations for curves of small genus, are competitive with the previously existing extensions of Schoof's algorithm in terms of complexity. In genus 2, our general method cannot improve on the complexity in $\widetilde{O}(\log^8 q)$ of the Gaudry-Harley-Schost algorithm based on resultants, which can actually be further reduced to $O\left((\log q)^{8-2/\omega+\varepsilon}\right)$ using the algorithm of Villard [143] for bivariate resultants. Indeed, the ℓ-torsion ideals involved have degrees in $O(\ell^4)$, so that our algorithm based on geometric resolution requires a number of field operations at least quadratic in that degree, i.e. an overall cost of at least $\widetilde{O}(\log^{10} q)$ because of the number and sizes of primes ℓ to consider.

In genus 3, no previous instantiation of the Schoof-Pila algorithm had been presented and the complexity of a potential extension was subject to speculation. Since the ℓ-torsion ideals have degrees in $O(\ell^6)$, we expect our general algorithm to have a complexity at least in $\widetilde{O}(\log^{14} q)$. Extending the resultant-based elimination scheme of Gaudry-Harley-Schost, we obtained a proven complexity in $\widetilde{O}(\log^{14} q)$ in Chapter 6. In fact, using the algorithm of Villard for computing bivariate resultants, the complexity of the resultant-based approach can be decreased to $O\left((\log q)^{14-4/\omega+\varepsilon}\right)$, which is also less than quadratic in the degree of the ℓ-torsion.

Still in genus 3 but for hyperelliptic curves with explicit RM, we have to solve systems of much smaller degrees, and we turned this into a point-counting algorithm of complexity $\widetilde{O}(\log^6 q)$. Setting $g = 3$ in the general counterpart to this algorithm designed in Chapter 7, we achieve a similar complexity because for $g \leq 3$ the conjectural result leading to the $\widetilde{O}_\eta(\log^6 q)$ is actually proven. However, using Villard's algorithm for bivariate resultants, the resultant-based algorithm of Chapter 6 reaches a complexity in $O((\log q)^{6-4/(3\omega)+\varepsilon})$ which is once again better than the general approach. We sum up all these results in Table 2.

Table 2: Asymptotic complexities for counting points on hyperelliptic curves of genus ≤ 3

Approach	$g = 2$	$g = 3$	$g = 2$ with RM	$g = 3$ with RM
Chapters 5 and 7	$\widetilde{O}(\log^{10} q)$	$\widetilde{O}(\log^{14} q)$	$\widetilde{O}(\log^{6} q)$	$\widetilde{O}(\log^{6} q)$
Chapters 3 and 6	$\widetilde{O}(\log^{8} q)$	$\widetilde{O}(\log^{14} q)$	$\widetilde{O}(\log^{5} q)$	$\widetilde{O}(\log^{6} q)$
Using [143]	$O((\log q)^{8-2/\omega+\varepsilon})$	$O((\log q)^{14-4/\omega+\varepsilon})$	$O((\log q)^{5-1/\omega+\varepsilon})$	$O((\log q)^{6-4/(3\omega)+\varepsilon})$

Practical experiments

In practice, we expect our general methods to be no match for the tailor-made algorithms in genus ≤ 3, not only because their complexities are lower, but also because the general approaches hide constants that we expect to be much larger. However, a comparison based on practical experiments in full generality is unrealistic in genus ≥ 3 because of the prohibitive complexities of both the algorithms of Chapters 5 and 7.

For practical experiments in genus 3, we considered the easier case of curves with explicit real multiplication, an approach that had previously been studied with benefit in genus 2. We were able to successfully count points on a genus-3 curve defined over $\mathbb{F}_{2^{64}-59}$. This is comparable in size with previous record computations due to Sutherland using generic group methods which also take advantage of particularities of the input curves, although such peculiar curves are more frequent than curves with explicit RM. In our practical experiments, we used a trivariate elimination scheme except that we computed Gröbner bases instead of trivariate resultants.

The complexity estimate in $\widetilde{O}_\eta(\log^6 q)$ conjectured in Chapter 7 could give hope of pushing practical experiments to higher genus, since the exponent of $\log q$ is independent of g. However, considering the RM families we presented and the conditions on primes ℓ, the smallest example available in genus larger than 3 is the computation of the 23-torsion of a hyperelliptic curve of genus 5. Even over a relatively small finite field, this is unrealistic because the systems to solve would have 5 variables with "large" degrees (estimated to be at least 10) and at least 25 variables with degree 1.

Prospective

A natural question that applies to all of our contributions is the possibility of extending our complexity bounds to non-hyperelliptic curves. Even if the Mumford representation allows for a much more straightforward representation of elements and simpler conditions to express the nullity of an element, this is not an absolute necessity. The most important result is that the degree of the ℓ-torsion ideal is still ℓ^{2g} in any Jacobian of a genus-g curve. Provided that we can model this ideal by a polynomial system with a number of variables that depends only on its dimension g and such that "only" $O(g^c)$ of them have degrees actually depending on ℓ, then the geometric resolution algorithm yields a point-counting algorithm running in time $O_g\left((\log q)^{O(g^c)}\right)$. Controlling the constant c and giving an explicit bound would already improve the result of Adleman and Huang [3], but we expect that it should be possible to prove that $c = 1$ as we did in the hyperelliptic case, at least for Jacobians of plane curves.

In Chapter 5, we perform a tedious analysis of how to handle non-generic elements in the

torsion subgroups. It is quite unsatisfactory that such amount of work is performed for cases which are supposed not to happen, or with an incredibly low probability. Actually, while we consider many cases, we do not even prove that they happen. Therefore, one could wonder whether all those non-genericities are possible. In Chapter 7, non-genericities are even less likely to become a nuisance since it is sufficient to have only one generic element in the kernel of our endomorphisms. Even better, one could try to completely remove the non-genericity analysis by proving for instance that given a curve, the proportion of primes ℓ for which non-genericity occurs is finite or sufficiently small. Conversely, a skeptical reader could attempt to create pathological curves such that avoiding all the "bad" primes ℓ would entail considering primes sufficiently large to hamper our complexity result. Note that because of our bounds, this would require finding a family of curves such that the largest required prime ℓ grows faster than any power of $\log q$.

The question of finding a classical point-counting algorithm running in time polynomial in both g and $\log q$ being open, we wonder whether the approach of Chapter 7 has the potential for providing a small yet non-trivial family of curves for which such an algorithm exists. The first reason why the algorithm presented in Chapter 7 is exponential in g is that the multihomogeneous Bézout bound has a combinatorial factor in $O(g^{g^2})$. Indeed, even though we manage to decrease the degrees of the equations by splitting the ℓ-torsion into a direct sum of kernels of endomorphisms of degree ℓ^2, our systems still have $O(g^2)$ variables. We have reviewed the other sources of factors exponential in g, and remarked that the polynomial systems appearing in the modelling come both in number and size exponential in g. Therefore, our approach needs further insight before turning into an algorithm running in time subexponential in g, even on a particular subset of curves.

Bibliography

[1] Simon Abelard, Pierrick Gaudry, and Pierre-Jean Spaenlehauer. Improved complexity bounds for counting points on hyperelliptic curves, 2017. To appear in *Foundations of Computational Mathematics*, ArXiv preprint 1710.03448.

[2] Simon Abelard, Pierrick Gaudry, and Pierre-Jean Spaenlehauer. Counting points on genus-3 hyperelliptic curves with explicit real multiplication, 2018. To appear in the Proceedings of the ANTS-XIII Conference (Thirteenth Algorithmic Number Theory Symposium), ArXiv preprint 1806.05834.

[3] Leonard M. Adleman and Ming-Deh Huang. Counting points on curves and Abelian varieties over finite fields. *Journal of Symbolic Computation*, 32(3):171–189, 2001.

[4] Leonard M. Adleman and Ming-Deh A. Huang. *Primality testing and Abelian varieties over finite fields*. Springer, 2006.

[5] François Apéry and Jean-Pierre Jouanolou. Élimination : le cas d'une variable. *Hermann, Collection Méthodes*, 2006.

[6] A. Oliver L. Atkin and François Morain. Finding suitable curves for the elliptic curve method of factorization. *Mathematics of Computation*, 60(201):399–405, 1993.

[7] A. Oliver L. Atkin and François Morain. Elliptic curves and primality proving. *Mathematics of Computation*, 61(203):29–68, 1993.

[8] Sean Ballentine, Aurore Guillevic, Elisa Lorenzo García, Chloe Martindale, Maike Massierer, Benjamin Smith, and Jaap Top. Isogenies for point counting on genus two hyperelliptic curves with maximal real multiplication. In *Algebraic Geometry for Coding Theory and Cryptography*, pages 63–94. Springer, 2017.

[9] Stéphane Ballet, Julia Pieltant, Matthieu Rambaud, and Jeroen Sijsling. On some bounds for symmetric tensor rank of multiplication in finite fields. *Contemporary Mathematics, AMS*, 686:93–121, 2017.

[10] Razvan Barbulescu, Joppe W. Bos, Cyril Bouvier, Thorsten Kleinjung, and Peter L. Montgomery. Finding ECM-friendly curves through a study of Galois properties. In *ANTS X*, volume 1 of *The open book series*, pages 63–86, 2012.

[11] Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2004.

[12] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of the F5 Gröbner basis algorithm. *Journal of Symbolic Computation*, 70:49–70, 2015.

[13] Jens Bauch, Daniel J. Bernstein, Henry de Valence, Tanja Lange, and Christine Van Vredendaal. Short generators without quantum computers: the case of multiquadratics. In *EUROCRYPT 2017*, volume 10210 of *LNCS*, pages 27–59. Springer, 2017.

[14] Eberhard Becker, Teo Mora, Maria Grazia Marinari, and Carlo Traverso. The shape of the shape lemma. In *Proceedings of ISSAC 1994*, pages 129–133. ACM, 1994.

[15] Elwyn R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24(111):713–735, 1970.

[16] Daniel Bernstein, Peter Birkner, Tanja Lange, and Christiane Peters. ECM using Edwards curves. *Mathematics of Computation*, 82(282):1139–1179, 2013.

[17] Daniel J. Bernstein. Curve25519: new Diffie–Hellman speed records. In *PKC 2006*, volume 3958 of *LNCS*, pages 207–228. Springer, 2006.

[18] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Peter Schwabe. Kummer strikes back: new DH speed records. In *ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 317–337. Springer, 2014.

[19] Christina Birkenhake and Herbert Lange. *Complex Abelian varieties*. Springer-Verlag, 2004.

[20] G. Bisson, R. Cosset, D. Robert, et al. AVIsogenies (abelian varieties and isogenies). *Magma package for explicit isogenies between abelian varieties*, 2010.

[21] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. *Algorithmes efficaces en calcul formel*. Published by the authors, 2017.

[22] Alin Bostan, Grégoire Lecerf, Bruno Salvy, Éric Schost, and Bernd Wiebelt. Complexity issues in bivariate polynomial factorization. In *Proceedings of ISSAC 2004*, pages 42–49. ACM, 2004.

[23] Ivan Boyer. *Variétés abéliennes et jacobiennes de courbes hyperelliptiques, en particulier à multiplication réelle ou complexe*. PhD thesis, Paris 7, 2014.

[24] Richard P. Brent and Paul Zimmermann. *Modern computer arithmetic*. Cambridge University Press, 2010.

[25] Antonio Cafure and Guillermo Matera. Fast computation of a rational point of a variety over a finite field. *Mathematics of Computation*, 75(256):2049–2085, 2006.

[26] Antonio Cafure and Guillermo Matera. An effective Bertini theorem and the number of rational points of a normal complete intersection over a finite field. *Acta Arithmetica*, 130(1):19–35, 2007.

[27] David G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Mathematics of computation*, 48(177):95–101, 1987.

[28] David G. Cantor. On the analogue of the division polynomials for hyperelliptic curves. *Journal fur die reine und angewandte Mathematik*, 447:91–146, 1994.

[29] Wouter Castryck, Jan Denef, and Frederik Vercauteren. Computing zeta functions of nondegenerate curves. *International Mathematics Research Papers*, Vol. 2006, 2006.

[30] Wouter Castryck, Hendrik Hubrechts, and Frederik Vercauteren. Computing zeta functions in families of $C_{a,b}$ curves using deformation. In *ANTS 2008*, volume 5011 of *LNCS*, pages 296–311. Springer, 2008.

[31] David Volfovich Chudnovsky and Gregory Volfovich Chudnovsky. Algebraic complexities and algebraic curves over finite fields. *Journal of Complexity*, 4(4):285–316, 1988.

[32] Laurent Clozel, Michael Harris, and Richard Taylor. Automorphy for some ℓ-adic lifts of automorphic mod ℓ Galois representations. *Publications mathématiques*, 108(1):1, 2008.

[33] Henri Cohen. *A course in computational algebraic number theory*. Springer, 1993.

[34] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2005.

[35] Stéphane Collart, Michael Kalkbrener, and Daniel Mall. Converting bases with the Gröbner walk. *Journal of Symbolic Computation*, 24(3-4):465–469, 1997.

[36] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 392–407. Springer, 2000.

[37] Jean-Marc Couveignes and Tony Ezome. Computing functions on Jacobians and their quotients. *LMS Journal of Computation and Mathematics*, 18(1):555–577, 2015.

[38] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 2007.

[39] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 559–585. Springer, 2016.

[40] Jean Della Dora, Claire Dicrescenzo, and Dominique Duval. About a new method for computing in algebraic number fields. In *European Conference on Computer Algebra*, pages 289–290. Springer, 1985.

[41] Jan Denef and Frederik Vercauteren. An extension of Kedlaya's algorithm to hyperelliptic curves in characteristic 2. *Journal of Cryptology*, 19(1):1–25, 2006.

[42] Noam D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational Perspectives on Number Theory*, pages 21–76. AMS/International Press, 1998. Proceedings of a Conference in Honor of A.O.L. Atkin.

[43] Jordan S. Ellenberg. Endomorphism algebras of Jacobians. *Advances in Mathematics*, 162:243–271, 2001.

[44] Andreas Enge and Emmanuel Thomé. Computing class polynomials for abelian surfaces. *Experimental Mathematics*, 23(2):129–145, 2014.

[45] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.

[46] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). *Proceedings of ISSAC 2002*, 2002.

[47] Jean-Charles Faugère, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[48] Jean-Charles Faugère, David Lubicz, and Damien Robert. Computing modular correspondences for abelian varieties. *Journal of Algebra*, 343(1):248–277, 2011.

[49] Francesc Fité, Kiran S. Kedlaya, Víctor Rotger, and Andrew V. Sutherland. Sato–Tate distributions and Galois endomorphism modules in genus 2. *Compositio Mathematica*, 148(5):1390–1442, 2012.

[50] Gerhard Frey and Michael Müller. Arithmetic of modular curves and applications. In B. Heinrich Matzat, Gert-Martin Greuel, and Gerhard Hiss, editors, *Algorithmic Algebra and Number Theory*, pages 11–48. Springer Verlag, 1999.

[51] Matsusaburô Fujiwara. Über die obere schranke des absoluten betrages der wurzeln einer algebraischen gleichung. *Tohoku Mathematical Journal, First Series*, 10:167–171, 1916.

[52] Steven Galbraith and Raminder S. Ruprai. An improvement to the Gaudry-Schost algorithm for multidimensional discrete logarithm problems. In *IMA International Conference on Cryptography and Coding*, volume 5921 of *LNCS*, pages 368–382. Springer, 2009.

[53] Steven D. Galbraith and Pierrick Gaudry. Recent progress on the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, 78(1):51–72, 2016.

[54] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013. Third edition.

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

[56] Pierrick Gaudry. Algorithmes de comptage de points d'une courbe définie sur un corps fini, 2013.

[57] Pierrick Gaudry and Robert Harley. Counting points on hyperelliptic curves over finite fields. In *ANTS 2000*, volume 1838 of *LNCS*, pages 313–332. Springer, 2000.

[58] Pierrick Gaudry, Florian Hess, and Nigel P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *Journal of Cryptology*, 15(1):19–46, 2002.

[59] Pierrick Gaudry, David R. Kohel, and Benjamin A. Smith. Counting points on genus 2 curves with real multiplication. In *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 504–519. Springer, 2011.

[60] Pierrick Gaudry and Éric Schost. Construction of secure random curves of genus 2 over prime fields. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 239–256. Springer, 2004.

[61] Pierrick Gaudry and Éric Schost. A low-memory parallel version of Matsuo, Chao and Tsujii's algorithm. In *ANTS-VI*, volume 3076 of *LNCS*, pages 208–222. Springer Verlag, 2004.

[62] Pierrick Gaudry and Éric Schost. Genus 2 point counting over prime fields. *Journal of Symbolic Computation*, 47(4):368–400, 2012.

[63] Gerard van der Geer, Everett W. Howe, Kristin E. Lauter, and Christophe Ritzenthaler. Tables of curves with many points, 2009.

[64] Marc Giusti, Grégoire Lecerf, and Bruno Salvy. A Gröbner free alternative for polynomial system solving. *Journal of complexity*, 17(1):154–211, 2001.

[65] Shafi Goldwasser and Joe Kilian. Almost all primes can be quickly certified. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 316–329. ACM, 1986.

[66] Valery Denisovich Goppa. Algebraico-geometric codes. *Izvestiya: Mathematics*, 21(1):75–91, 1983.

[67] Michael Harris, Nick Shepherd-Barron, and Richard Taylor. A family of Calabi-Yau varieties and potential automorphy. *Annals of Mathematics*, pages 779–813, 2010.

[68] Robin Hartshorne. *Algebraic geometry*. Springer, 1977.

[69] David Harvey. Computing zeta functions of arithmetic schemes. *Proceedings of the London Mathematical Society*, 111(6):1379–1401, 2015.

[70] David Harvey and Andrew V. Sutherland. Computing Hasse–Witt matrices of hyperelliptic curves in average polynomial time. *LMS Journal of Computation and Mathematics*, 17(A):257–273, 2014.

[71] David Harvey and Andrew V. Sutherland. Computing Hasse-Witt matrices of hyperelliptic curves in average polynomial time II. *Contemporary Mathematics*, 663:127–148, 2016.

[72] Joos Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 24(3):239–277, 1983.

[73] Florian Hess. Computing Riemann–Roch spaces in algebraic function fields and related topics. *Journal of Symbolic Computation*, 33(4):425–445, 2002.

[74] Alston S. Householder. The Padé table, the Frobenius identities, and the qd algorithm. *Linear Algebra and its applications*, 4(2):161–174, 1971.

[75] Ming-Deh Huang and Doug Ierardi. Counting points on curves over finite fields. *Journal of Symbolic Computation*, 25(1):1–21, 1998.

[76] Peter Kaiser. Greatest numbers certified with Primo. `http://primes.utm.edu/primes/page.php?id=123996`.

[77] Wolfgang Kampkötter. *Explizite gleichungen für Jacobische varietäten hyperelliptischer kurven*. Universität Essen. Institut für Experimentelle Mathematik, 1991.

[78] Nicholas M. Katz and Peter Sarnak. *Random matrices, Frobenius eigenvalues, and monodromy*, volume 45. AMS, 1999.

[79] Motoko Qiu Kawakita. Certain sextics with many rational points. *Advances in Mathematics of Communications*, 11(2), 2017.

[80] Kiran S. Kedlaya. Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology. *Journal of the Ramanujan Mathematical Society*, 16(4):323–338, 2001.

[81] Kiran S. Kedlaya. Quantum computation of zeta functions of curves. *computational complexity*, 15(1):1–19, 2006.

[82] Kiran S. Kedlaya and Andrew V. Sutherland. Hyperelliptic curves, L-polynomials, and random matrices. *Contemporary Mathematics*, 14:119, 2009.

[83] Kamal Khuri-Makdisi. Linear algebra algorithms for divisors on an algebraic curve. *Mathematics of Computation*, 73(245):333–357, 2004.

[84] Steven L. Kleiman. Bertini and his two fundamental theorems. *ArXiv e-print alg-geom/9704018v1*, 1997.

[85] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of cryptology*, 1(3):139–150, 1989.

[86] Neal Koblitz. CM-curves with good cryptographic properties. In *Annual International Cryptology Conference*, pages 279–287. Springer, 1991.

[87] David R. Kohel and Benjamin A. Smith. Efficiently computable endomorphisms for hyperelliptic curves. In *ANTS VII*, volume 4076 of *LNCS*, pages 495–509. Springer Verlag, 2006.

[88] Junichi Kuroki, Masaki Gonda, Kazuto Matsuo, Jinhui Chao, and Shigeo Tsujii. Fast genus three hyperelliptic curve cryptosystems. In *The 2002 Symposium on Cryptography and Information Security, Japan—SCIS*, 2002.

[89] Yagati N. Lakshman and Daniel Lazard. On the complexity of zero-dimensional algebraic systems. In *Effective methods in algebraic geometry*, pages 217–225. Springer, 1991.

[90] Tanja Lange. Formulae for arithmetic on genus 2 hyperelliptic curves. *Applicable Algebra in Engineering, Communication and Computing*, 15(5):295–328, 2005.

[91] Daniel Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *European Conference on Computer Algebra*, pages 146–156. Springer, 1983.

[92] Daniel Lazard. Solving systems of algebraic equations. *ACM SIGSAM Bulletin*, 35(3):11–37, 2001.

[93] Hendrik W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of mathematics*, pages 649–673, 1987.

[94] Reynald Lercier and David Lubicz. Counting points on elliptic curves over finite fields of small characteristic in quasi quadratic time. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 360–373. Springer, 2003.

[95] Reynald Lercier and David Lubicz. A quasi quadratic time algorithm for hyperelliptic curve point counting. *The Ramanujan Journal*, 12(3):399–423, 2006.

[96] David Lubicz and Damien Robert. Arithmetic on Abelian and Kummer varieties. *Finite Fields and Their Applications*, 39:130–158, 2016.

[97] F. S. Macaulay. Some formulae in elimination. *Proceedings of the London Mathematical Society*, 1(1):3–27, 1902.

[98] Chloe Martindale. *Isogeny graphs, modular polynomials, and applications*. PhD thesis, PhD thesis, Universiteit Leiden, 2017. in preparation, 2018.

[99] Kazuto Matsuo, Jinhui Chao, and Shigeo Tsujii. An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields. In *ANTS 2002*, volume 2369 of *LNCS*, pages 461–474. Springer, 2002.

[100] Ernst W. Mayr and Albert R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in mathematics*, 46(3):305–329, 1982.

[101] Jean-François Mestre. Familles de courbes hyperelliptiques à multiplications réelles. In *Arithmetic algebraic geometry*, pages 193–208. Springer, 1991.

[102] Jean-François Mestre. Lettre adressée à Gaudry et Harley. *Available on http://www. math. jussieu. fr/˜ mestre*, 2000.

[103] Enea Milio. *Calcul de polynômes modulaires en dimension 2*. PhD thesis, Université de Bordeaux, 2015.

[104] Enea Milio. Computing isogenies between Jacobian of curves of genus 2 and 3. working paper or preprint, September 2017.

[105] Victor S. Miller. Use of elliptic curves in cryptography. In *CRYPTO 1985*, volume 218 of *LNCS*, pages 417–426. Springer, 1985.

[106] James S. Milne. Jacobian varieties. In *Arithmetic geometry*, pages 167–212. Springer, 1986.

[107] Shinji Miura. Algebraic geometric codes on certain plane curves. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 76(12):1–13, 1993.

[108] Peter L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264, 1987.

[109] François Morain. Primality proving using elliptic curves: an update. In *ANTS 1998*, volume 1423 of *LNCS*, pages 111–127. Springer, 1998.

[110] Alexander Morgan and Andrew Sommese. A homotopy for solving general polynomial systems that respects m-homogeneous structures. *Applied Mathematics and Computation*, 24(2):101–113, 1987.

[111] David Mumford, Chidambaran Padmanabhan Ramanujam, and Jurij Ivanovič Manin. *Abelian varieties*, volume 108. Oxford university press Oxford, 1974.

[112] Vassiliy Ilyich Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

[113] Committee on National Security Systems. Use of public standards for the secure sharing of information among national security systems, advisory memorandum. https://cryptome.org/2015/08/CNSS_Advisory_Memo_02-15.pdf, 2015.

[114] Jonathan Pila. Frobenius maps of Abelian varieties and finding roots of unity in finite fields. *Mathematics of Computation*, 55(192):745–763, 1990.

[115] Jonathan Pila. Counting points on curves over families in polynomial time. *arXiv preprint math/0504570*, 2005.

[116] Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.). *IEEE Transactions on information Theory*, 24(1):106–110, 1978.

[117] Bjorn Poonen. Using zeta functions to factor polynomials over finite fields. *arXiv preprint arXiv:1710.00970*, 2017.

[118] Matthieu Rambaud. Finding optimal Chudnovsky-Chudnovsky multiplication algorithms. In *International Workshop on the Arithmetic of Finite Fields*, pages 45–60. Springer, 2014.

[119] Joost Renes, Peter Schwabe, Benjamin Smith, and Lejla Batina. μKummer: Efficient hyperelliptic signatures and key exchange on microcontrollers. In *CHES 2016*, volume 9813 of *LNCS*, pages 301–320. Springer, 2016.

[120] Joost Renes and Benjamin Smith. qDSA: Small and secure digital signatures with curve-based Diffie–Hellman key pairs. In *ASIACRYPT 2017*, volume 10625 of *LNCS*, pages 273–302. Springer, 2017.

[121] Christophe Ritzenthaler. Point counting on genus 3 non hyperelliptic curves. In *ANTS 2004*, volume 3076 of *LNCS*, pages 379–394. Springer, 2004.

[122] Damien Robert. *Fonctions thêta et applications à la cryptographie*. PhD thesis, Université Henri Poincaré-Nancy I, 2010.

[123] Mohab Safey El Din and Éric Schost. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *Journal of the ACM*, 63(6):1– 48, 2017.

[124] Takakazu Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *Journal of the Ramanujan Mathematical Society*, 15(4):247–270, 2000.

[125] Takakazu Satoh. On p-adic point counting algorithms for elliptic curves over finite fields. In *International Algorithmic Number Theory Symposium*, volume 2369 of *LNCS*, pages 43–66. Springer, 2002.

[126] Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical programming*, 66(1-3):181–199, 1994.

[127] René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of Computation*, 44(170):483–494, 1985.

[128] René Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995.

[129] Yih-Dar Shieh. *Arithmetic aspects of point counting and Frobenius distributions.* PhD thesis, Aix-Marseille, 2015.

[130] G. Shimura. *Introduction to the arithmetic theory of automorphic functions*, volume 11 of *Publications of the Mathematical Society of Japan.* Iwanami Shoten and Princeton University Press, 1971.

[131] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 256–266. Springer, 1997.

[132] Andrew J. Sommese and Charles W. Wampler II. *The numerical solution of systems of polynomials arising in engineering and science.* World Scientific, 2005.

[133] Andrew Sutherland. A generic approach to searching for Jacobians. *Mathematics of Computation*, 78(265):485–507, 2009.

[134] Andrew V. Sutherland. *Order computations in generic groups.* PhD thesis, Massachusetts Institute of Technology, 2007.

[135] Andrew V. Sutherland. Accelerating the CM method. *LMS Journal of Computation and Mathematics*, 15:172–204, 2012.

[136] Agnes Szanto. Multivariate subresultants using Jouanolou matrices. *Journal of Pure and Applied Algebra*, 214(8):1347–1369, 2010.

[137] John Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones mathematicae*, 2(2):134–144, 1966.

[138] Walter Tautz, Jaap Top, and Alain Verberkmoes. Explicit hyperelliptic curves with real multiplication and permutation polynomials. *Canad. J. Math*, 43(5):1055–1064, 1991.

[139] Richard Taylor. Automorphy for some ℓ-adic lifts of automorphic mod ℓ Galois representations. II. *Publications mathématiques*, 108(1):183–239, 2008.

[140] Gérald Tenenbaum. *Introduction to analytic and probabilistic number theory.* Cambridge university press, 1995.

[141] Akira Terui. Recursive polynomial remainder sequence and its subresultants. *Journal of Algebra*, 320(2):633–659, 2008.

[142] Jan Tuitman. Counting points on curves using a map to $\mathbb{P}^1$, II. *Finite Fields and Their Applications*, 45:301–322, 2017.

[143] Gilles Villard. On computing the resultant of generic bivariate polynomials. In *Proceedings of ISSAC 2018.* ACM, 2018.

[144] Paul van Wamelen. Proving that a genus 2 curve has complex multiplication. *Mathematics of Computation*, 68(228):1663–1677, 1999.

[145] William C. Waterhouse and James S. Milne. Abelian varieties over finite fields. In *Proc. Sympos. Pure Math*, volume 20, pages 53–64, 1971.

[146] Annegret Weng. Constructing hyperelliptic curves of genus 2 suitable for cryptography. *Mathematics of Computation*, 72(241):435–458, 2003.

[147] Richard Zippel. *Effective polynomial computation*. Springer Verlag, 1993.

Résumé en Français

Courbes algébriques sur les corps finis et applications

Les courbes algébriques font partie du paysage mathématique depuis plus de 2000 ans, depuis l'Antiquité et les fondements de la Géométrie jusqu'à la preuve du dernier théorème de Fermat dans les années 1990. De telles courbes sont souvent décrites comme le lieu des solutions d'un systèmes polynomial et modélisent de nombreuses situations, d'où leur vaste domaine d'application y compris en dehors des Mathématiques. Dans cette thèse, nous nous concentrons sur les courbes algébriques planes, c'est-à-dire des courbes données par une équation de la forme $f(x, y) = 0$, avec f un polynome bivarié. Un point de la courbe correspond à une solution de son équation, mais il faut s'accorder sur le sens à donner à la notion de solution. Les mathématiciens grecs de l'Antiquité en savent quelque chose puisqu'ils ont dû faire face au fait que même des équations à coefficients entiers peuvent avoir des solutions irrationnelles. Il nous faut donc préciser le corps dans lequel vivent les coefficients du polynôme f, que nous appellerons le corps de base de la courbe, et considérer les points de la courbe dans la clôture algébrique de ce corps. Rien ne s'oppose toutefois à ce que certains d'entre eux appartiennent bien au corps de base, et nous les qualifierons de rationnels. Bien que le corps des réels semble particulièrement naturel (sans mauvais jeu de mots) pour étudier les corps et en particulier pour les tracer, les courbes définies sur les corps finis sont également sources d'intérêt pour leurs multiples applications. Dans cette thèse, nous considérons presque exclusivement des courbes définies sur un corps fini de caractéristique impaire, bien que nous profitons parfois de propriétés de certaines courbes qui sont des réductions modulo un nombre premier de courbes définies sur $\mathbb{Q}$.

Les courbes sur les corps finis ont historiquement trouvé leurs premières applications en théorie des nombres, et plus particulièrement pour factoriser des entiers ou tester leur primalité. En effet, l'algorithme ECM [93] est toujours compétitif par rapport aux algorithmes basé sur le crible sur les corps de nombres (NFS) pour trouver des "petits" facteurs (de taille inférieure à 83 bits). L'algorithme ECPP introduit par Goldwasser et Kilian, puis amélioré par Atkin et Morain [65, 109] est encore aujourd'hui parmi les plus rapides pour générer des certificats de primalité, il a même récemment été utilisé pour prouver la primalité d'un entier de 34987 bits [76]. Si son efficacité n'est plus à prouver, la complexité de cet algorithme reste cependant heuristique. Toujours en utilisant des courbes algébriques, de genre 2 cette fois-ci, Adleman et Huang [4] ont pu construire un algorithme de Las-Vegas pour prouver la primalité en temps polynomial. Une approche basée sur des courbes encore plus générales est évoquée pour parvenir à un algorithme de factorisation de polynômes sur des corps finis qui soit à la fois déterministe et de complexité polynomiale. On peut également mentionner l'utilisation de techniques d'interpolation sur des courbes algébriques par Chudnovsky et Chudnovsky [31] dans les années 1980 pour étudier la complexité du produit de deux polynômes sur des corps finis. Cette approche est toujours d'actualité et il existe une littérature abondante et récente [118, 9] visant à construire les meilleures courbes possibles dans cette optique. En particulier, on recherche des

courbes avec autant de points rationnels que possible, par exemple les courbes regroupées dans la base de donnée libre [63].

Il en va de même pour l'algorithme ECM dont on souhaite améliorer les performances en choisissant des familles de courbes particulièrement adéquates, soit parce qu'elles ont plus de chances d'avoir un nombre de points friable [6, 10], ou parce qu'elles permettent une arithmétique plus rapide [108, 16]. Si la factorisation d'entiers intéresse fortement les cryptanalystes, les courbes algébriques et notamment les courbes elliptiques ont également des applications constructives en cryptographie. En effet, le groupe des points rationnels d'une courbe elliptique sur un corps fini est un parfait exemple de groupe dans lequel calculer des logarithmes discrets est difficile. Contrairement à RSA et au logarithme discret dans le groupe multiplicatif des corps finis, il n'existe pas pour le moment d'attaque sous-exponentielle, ce qui permet d'opter pour des clés bien plus petites. Cela dit, même un algorithme exponentiel peut être efficace si le groupe utilisé est de petite taille ou si son cardinal est très friable [116]. Compter le nombre de points rationnels d'une courbe est donc une étape essentielle avant de décider ou non si elle respecte l'exigence de sécurité que l'on se fixe.

À mesure que l'étude des courbes se développait, d'autres objets mathématiques associés furent introduits, à l'image des nombreuses fonctions L et zeta qui occupent aujourd'hui une place centrale dans la théorie des nombres moderne. Ainsi, on trouve plusieurs exemples d'énoncés de théorie des nombres qui furent établis en prouvant des résultats de nature analytique sur des fonctions complexes, comme par exemple la conjecture de Sato-Tate. Cette conjecture concerne la distribution du nombre de points rationnels de la réduction modulo p d'une courbe elliptique définie sur $\mathbb{Q}$ lorsque p varie, et fut prouvée autour de 2005 [67, 32, 139]. Des travaux sont en cours pour formuler des conjectures similaires dans des cas plus généraux, notamment en genre 2 et 3 [49]. Pour ce faire, des algorithmes de comptage de points comme celui d'Harvey [71] sont au coeur d'expériences numériques impliquant une puissance de calcul considérable.

Chacune de ces applications a son propre contexte, de la nature des courbes utilisées à la taille du corps de définition. Dans cette thèse, nous considérons exclusivement des courbes hyperelliptiques données par un modèle imaginaire $y^2 = f(x)$, avec f un polynôme unitaire sans carré de degré impair. Le degré $\deg f = 2g + 1$ détermine le genre g de la courbe associée, qui est un paramètre important dans tout le manuscrit. Deux paramètres supplémentaires p et n déterminent la caractéristique p et la taille $q = p^n$ du corps de base de la courbe. Dans tout le manuscrit, nous utilisons la notation standard $O()$, la notation $\widetilde{O}()$ lorsque nous omettons les termes polylogarithmiques, et la notation O_g quand nous omettons également tous les termes dépendant uniquement de g (et indépendants de q). En utilisant des algorithmes rapides (voir par exemple [24]), nous partons du principe que chaque opération dans le corps fini $\mathbb{F}_q$ a un coût en $\widetilde{O}(\log q)$.

L'algorithme de Schoof

Nous venons de voir plusieurs raisons pour lesquelles connaître le nombre de points rationnels d'une courbe elliptique peut être capital. Pour ce faire, une approche est de construire des courbes ayant un nombre de points spécifié à l'avance, par exemple avec la méthode CM de [6] qui fut utilisée en cryptographie [86]. Une autre façon de procéder consiste à considérer des courbes au hasard, compter leurs points rationnels et répéter tant que le résultat n'est pas satisfaisant. Bien qu'il existe des approches élémentaires pour réaliser cette tâche, comme par exemple tester toutes les paires $(x, y) \in \mathbb{F}_q$ pour vérifier si elles satisfont l'équation de la courbe, leur complexité les rend inadaptées dans la plupart des cas. En 1985, Schoof propose le premier algorithme de

comptage de points dont la complexité est polynomiale en $\log q$ [127]. Bien qu'à cette époque son algorithme ne soit pas considéré suffisamment efficace pour une utilisation pratique, il a ouvert la voie à de nombreuses améliorations et généralisations aujourd'hui regroupées sous le terme de méthodes ℓ-adiques. Quelques années plus tard, Elkies et Atkin conçurent des améliorations [128] qui en firent un algorithme utilisable et remarquablement efficace. Sous le nom SEA (Schoof-Elkies-Atkin), cette variante de l'algorithme de Schoof permet aujourd'hui de compter les points d'une courbe elliptique et de générer des courbes cryptographiques de manière plus que satisfaisante.

L'idée de l'algorithme de Schoof est de calculer le nombre de points rationnels modulo des nombres premiers ℓ jusqu'à ce que la valeur exacte puisse être déduite en appliquant le théorème des restes chinois. En effet, les bornes de Weil impliquent que ce nombre se trouve dans un intervalle de taille $\lceil 4\sqrt{q} \rceil$ et donc que le nombre et la taille du plus grand ℓ à considérer sont tous les deux en $O(\log q)$. Pour obtenir l'information modulo ℓ, Schoof fait agir l'endomorphisme de Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ sur la ℓ-torsion, c'est-à-dire l'ensemble des points P tels que ℓP est le point à l'infini, qui est l'élément neutre pour l'addition sur les points de la courbe. Pour ℓ premier différent de la caractéristique, la ℓ torsion est en fait un espace vectoriel isomorphe à $(\mathbb{Z}/\ell\mathbb{Z})^2$. L'action de l'endomorphisme de Frobenius est alors représentée par une matrice 2×2 dont la trace détermine le nombre de points rationnels modulo ℓ. L'étape la plus coûteuse dans cet algorithme est le calcul de π dans la ℓ-torsion, qui coûte $\widetilde{O}(\ell^2 \log q)$ opérations dans le corps de base. En prenant en compte le coût de telles opérations, la taille du plus grand ℓ et le nombre de premiers ℓ à considérer, la complexité totale de l'algorithme de Schoof est en $\widetilde{O}(\log^5 q)$. Dans l'algorithme SEA, le gain de complexité est réalisé en remplaçant la ℓ-torsion par un sous groupe isomorphe à $\mathbb{Z}/\ell\mathbb{Z}$ dans lequel chaque opération coûte $\widetilde{O}(\ell \log q)$ opérations dans le corps de base, ce qui donne une complexité en $\widetilde{O}(\log^4 q)$ pour l'algorithme SEA.

Jacobiennes de courbes

Pour des applications telles que la construction de groupes cryptographiques, la notion qui généralise une courbe elliptique n'est pas directement une courbe de genre plus grand, parce que ses points rationnels ne forment pas nécessairement un groupe. Les jacobiennes de telles courbes représentent un objet plus adapté car elles ont une structure de groupe (plus précisément de variété abélienne). Il en va de même pour la ℓ-torsion qui est celle de la jacobienne et non plus de la courbe elle-même. Nous verrons qu'en genre plus grand que 2, la détermination de la ℓ-torsion surpasse l'application du Frobenius et devient l'étape dominante dans la complexité. Cette étape repose de manière cruciale sur l'arithmétique de la jacobienne.

Bien qu'il existe des algorithmes permettant d'effectuer des additions dans des jacobiennes de courbes non-hyperelliptiques, par exemple [73, 83], cette thèse se concentre sur le cas hyperelliptique parce que cela simplifie grandement l'arithmétique dans les jacobiennes associées, et notamment la façon dont on parvient à décrire la ℓ-torsion. Les éléments de jacobiennes de courbes hyperelliptiques de genre g peuvent être représentés par leurs coordonnées de Mumford, c'est-à-dire par une paire de polynômes de degrés respectifs bornés par g et $g-1$. L'addition de deux éléments sous cette forme est réalisée avec l'algorithme de Cantor [27], avec une complexité en temps et en mémoire qui est quasi-linéaire en $g \log q$. Via une exponentiation binaire, on déduit de cet algorithme un moyen efficace de multiplier des éléments d'une jacobienne hyperelliptique par des scalaires.

Comptage de points sur des courbes

Dans les années 1990, Pila [114] constate que les résultats théoriques sur lesquels repose l'algorithme de Schoof sont valides bien au-delà des courbes elliptiques. Il étend ainsi l'algorithme de Schoof au cas des variétés abéliennes et en particulier des (jacobiennes de) courbes algébriques. L'algorithme de Pila ne se contente pas de renvoyer le nombre de points rationnels, mais le polynôme caractéristique de l'endomorphisme de Frobenius ou, de manière équivalente, la fonction zeta locale de la courbe. Comme l'algorithme de Schoof, l'algorithme de Pila est polynomial en $\log q$ mais il dépend en plus de paramètres comme le genre (la dimension) de la courbe, et d'une manière exponentielle. Cet algorithme n'a pas été conçu pour être directement implémentable, mais dans le cas particulier des courbes hyperelliptiques de genre 2, l'emploi de techniques issues du calcul formel pour décrire les sous-groupes de torsion et l'action du Frobenius sur ces sous-groupes a permis à Gaudry et Harley [57] de créer et d'implémenter un analogue de l'algorithme de Schoof. Cet algorithme fut ensuite amélioré par Gaudry et Schost au point d'être suffisamment efficace pour générer une jacobienne de courbe de genre 2 de taille cryptographique [60, 62]. Notons que comme en genre 1, il est toujours possible de créer des jacobiennes avec un nombre de points fixé à l'avance via la méthode CM [146, 135, 44].

Au début des années 2000, d'autres méthodes également basées sur le calcul de l'action (d'une approximation p-adique) du Frobenius ont été développées, d'abord par Satoh [125] dans le cas elliptique. Cette méthode a ensuite été étendue dans un contexte bien plus général et de nombreux algorithmes, regroupés sous le noms de méthodes p-adiques, ont vu le jour. Ces différents algorithmes considèrent différents relèvements du Frobenius agissant sur différents espaces, comme par exemple celui de Kedlaya [80] basé sur la cohomologie de Monsky-Washnitzer qui s'applique à des courbes hyperelliptiques de genre quelconque, et son analogue en caractéristique 2 par Denef et Vercauteren [41]. D'autres extensions à des courbes de plus en plus générales ont ensuite été proposées [30, 29, 142] et font toujours l'objet d'une recherche active. En caractéristique 2, une variante de l'agorithme de Satoh fut proposée indépendamment par Mestre [102] qui propose une expression du Frobenius en termes de suite artihmético-géométrique et qui représente aujourd'hui la méthode la plus rapide pour compter les points de courbes elliptiques définies sur $\mathbb{F}_{2^n}$. Toujours dans [102], Mestre propose une généralisation de sa méthode en genre 2, et d'autres travaux l'ont ensuite étendue dans deux directions : soit en choisissant un corps de (petite) caractéristique impaire [94], soit en considérant des courbes de genre plus grand [121, 95].

Ces méthodes fournissent des algorithmes utilisables en pratique, et dont la complexité est polynomiale en g et en n, mais exponentielle en $\log p$, de sorte que les méthodes p-adiques et ℓ-adiques sont complémentaires lorsque l'un des deux paramètres g et p est petit. En revanche, il n'existe pas d'algorithme classique de comptage de points dont la complexité est polynomiale en ces deux paramètres. Notons toutefois que Kedlaya [81] a proposé un tel algorithme en exploitant des primitives quantiques et que pour une courbe définie sur $\mathbb{Q}$, Harvey [70] parvient à compter les points de ses réductions modulo tous les nombres premiers p inférieurs à une borne N en temps quasi-linéaire en N, ce qui veut dire que la complexité moyenne par p est polynomiale en p. Bien que cela ne s'applique qu'à des réductions d'une même courbe sur $\mathbb{Q}$, ces algorithmes sont particulièrement adaptés pour formuler des généralisations de la conjecture de Sato-Tate.

Dans cette thèse, lorsque nous parlons de compter les points d'une courbe ou de sa jacobienne, nous parlons en réalité de résoudre le problème suivant.

Calcul de la fonction zeta d'une courbe hyperelliptique. Étant donné q une puissance d'un nombre premier impair, un entier $g \leq 1$ et un polynôme univarié $f \in \mathbb{F}_q[X]$ de degré $2\,g+1$, soit $\mathcal{C}$ la courbe hyperelliptique associée au modèle de Weierstrass $Y^2 = f(X)$. Calculer le numérateur $P_{\mathcal{C}} \in \mathbb{Z}[T]$ de la fonction zeta locale de $\mathcal{C}$:

$$Z(\mathcal{C}/\mathbb{F}_q, T) = \exp\left(\sum_{i=1}^{\infty} \#\mathcal{C}(\mathbb{F}_{q^i}) \cdot \frac{T^i}{i}\right) = \frac{P_{\mathcal{C}}(T)}{(1-T)(1-qT)}.$$

Avec $\mathcal{C}(\mathbb{F}_{q^i})$ l'ensemble des points $\mathcal{C}$ dont les coordonnée sont dans $\mathbb{F}_{q^i}$.

Sous-groupes de torsion

Une étape clé dans les méthodes ℓ-adiques est la détermination de l'action du Frobenius sur les sous-groupes de ℓ-torsion. Dans l'algorithme de Schoof, la ℓ-torsion de la courbe est l'ensemble des points dont l'abscisse annule des polynômes ψ_ℓ de degrés $(\ell^2-1)/2$ que l'on appelle polynômes de ℓ-division. Ainsi l'action du Frobenius $\pi : (x, y) \mapsto (x^q, y^q)$ peut être calculée en répétant des étapes d'exponentiation et de réduction par les équations définissant la ℓ-torsion : $y^2 = f(x)$ et $\psi_\ell(x) = 0$. Dans un contexte plus général, Pila appelle cette étape le calcul d'une représentation de bas degré du Frobenius.

Pour les courbes elliptiques, les polynômes de division donnent une représentation simple et manipulable de la ℓ-torsion. Pour des courbes de genre supérieur, a priori, nous ne pouvons pas calculer de représentation de bas degré du Frobenius comme dans l'algorithme de Schoof car nous n'avons de telle description pour la ℓ-torsion. Il faut donc calculer une telle représentation, par exemple en calculant une base de Gröbner de l'idéal de torsion, avant de pouvoir réduire le Frobenius.

Dans cette thèse, nous suivons l'approche de Gaudry-Harley-Schost [57, 60, 62] et commençons par écrire l'équation $\ell D = 0$ dans la jacobienne. Pour ce faire, nous avons besoin d'une description de la multiplication par ℓ en tant qu'application rationnelle. Pour P un point d'une courbe hyperelliptique, il existe $2g+2$ polynômes décrivant les coordonnées de Mumford du diviseur $\ell(P - P_\infty)$. Ces polynômes ont été introduits par Cantor [28] et nommés d'après lui. En écrivant un élément D de la jacobienne comme somme formelle de points, on peut ainsi déduire une première description de la ℓ-torsion en tant qu'ensemble des solutions du système $\ell D = 0$.

Une fois ce système calculé, nous le résolvons afin d'avoir une représentation de la ℓ-torsion nous permettant de réduire le Frobenius. Cette étape est la plus coûteuse dans nos algorithmes, à la fois en théorie et en pratique. Nous apportons ainsi un soin particulier à la façon dont nous modélisons la ℓ-torsion par des systèmes polynomiaux et aux techniques de résolution que nous utilisons car elles ont un impact significatif sur les complexités et les temps de calcul de nos algorithmes de comptage de points.

Résolution de systèmes polynomiaux

Étant donnés des polynômes multivariés $f_1, \ldots, f_m$ dans $K[x_1, \ldots, x_n]$, nous voulons trouver des équations définissant l'idéal $I = \langle f_1, \ldots, f_m \rangle$ et telles qu'il soit possible d'effectuer des opérations arithmétiques dans $K[x_1, \ldots, x_n]/I$. Dans cette thèse, nous nous acquittons de cette tâche pour $I = I_\ell$, l'idéal de ℓ-torsion d'une jacobienne de courbe hyperelliptique, soit en mettant sous

forme triangulaire le système $f_1 = 0, \ldots, f_m = 0$, ou en calculant une résolution géométrique de ce système, c'est-à-dire une paramétrisation des coordonnées de ses solutions par les racines d'un polynôme univarié. Dans les deux cas, nous appelons cette opération "résoudre le système". La littérature propose de nombreuses façons de résoudre des systèmes polynomiaux et nous en utiliserons 3 selon leur complexité asymptotique ou leur efficacité pratique. Cela dépend de nombreux paramètres comme le nombre de variables ou le degré des équations définissant notre système, mais aussi de propriétés moins évidentes comme la dimension de l'idéal, son degré (le nombre de solutions dans la clôture algébrique s'il est fini), ainsi que d'éventuelles particularités structurelles du système. Dans cette thèse, nous résolvons des systèmes qui modélisent des sous-ensembles de la ℓ-torsion, donc nous savons à l'avance qu'ils sont zéro-dimensionnels et que leur degré est borné par ℓ^{2g}.

La torsion des courbes de genre 2, par exemple, est représentée par un système d'équations en deux variables, que l'on peut mettre sous forme triangulaire en calculant des résultants bivariés, qui représentent actuellement la meilleure option à la fois en termes d'efficacité et de complexité asymptotique. Dans le chapitre 6, nous modélisons la ℓ-torsion d'une jacobienne de courbe hyperelliptique de genre 3 par un système polynomial trivarié que nous mettons en forme triangulaire en calculant des résultants. Bien que cette approche donne une complexité asymptotique satisfaisante, calculer une base de Gröbner du système via l'algorithme F4 [45] s'avère bien plus efficace en pratique, aussi est-ce l'approche choisie dans nos expériences. Cela dit, bien que la complexité des algorithmes F4 et F5 [45, 46] ait été intensément étudiée [11, 12], aucune borne de complexité présente dans la littérature n'est assez fine pour nous permettre d'utiliser ces algorithmes en théorie et en pratique.

Dans le chapitre 5, nous modélisons d'une autre manière la ℓ-torsion de courbes hyperelliptiques de genre quelconque en faisant intervenir $O(g^2)$ variables au lieu de g. Cependant, il n'y a toujours que g variables dont le degré dépend de ℓ, toutes les autres ayant un degré en $O_g(1)$. Dans ce cas, l'emploi de l'algorithme de résolution géométrique de [26, 64] est dicté par la nécessité d'invoquer des résultats de complexité qui prennent en compte cette structure multihomogène si particulière.

Contributions

Dans cette thèse, nous étudions les méthodes ℓ-adiques dérivées des algorithmes de Schoof et Pila. La complexité de tels algorithmes est au cœur de ce manuscrit, et notamment la dépendance en g dans l'exposant de $\log q$. La première contribution, publiée en tant que [1], propose un algorithme de comptage de points sur les courbes hyperelliptiques en grande caractéristique dont on borne la complexité par une puissance de $\log q$ qui croît linéairement en g. Ce résultat s'inscrit dans la continuité des travaux d'Adleman et Huang [3] qui ont établi que cet exposant était polynomial en g dans le cas général, et quasi-quadratique dans le cas hyperelliptique. L'état de l'art concernant cet exposant est résumé dans la table 3. Pour atteindre une telle complexité, notre algorithme n'est guère différent de celui de Pila, mais notre analyse de complexité fait intervenir une nouvelle modélisation de la ℓ-torsion par un système polynomial structuré, comme expliqué plus haut. Cette structure est la clé de voûte de notre résultat, et l'on remarque d'ailleurs qu'en suivant notre raisonnement sans l'exploiter, on retrouve un résultat similaire à celui d'Adleman et Huang en $O\left((\log q)^{O(g^2 \log g)}\right)$. Si l'idée est naturelle, son exécution nécessite de surmonter quelques obstacles techniques et notamment de s'assurer que le système polynomial que l'on considère vérifie bien des hypothèses de généricité sur lesquelles reposent les résultats de complexité pour le calcul de résolution géométrique. Un autre obstacle est qu'en réalité notre modélisation fait intervenir un grand nombre de systèmes polynômiaux pour capturer toute la ℓ-torsion, y compris

certains éléments “spéciaux”.

Table 3: Complexité asymptotique pour calculer la fonction zeta locale d'une variété abélienne de dimension g sur $\mathbb{F}_q$

Auteurs (année)	Complexité	Contexte
Pila (1990)	$O\left((\log q)^{g^{O(g)}}\right)$	Variétés abéliennes
Huang-Ierardi (1998)	$O\left((\log q)^{g^{O(1)}}\right)$	Courbes planes
Adleman-Huang (2001)	$O\left((\log q)^{g^{O(1)}}\right)$	Variétés abéliennes
Adleman-Huang (2001)	$O\left((\log q)^{O(g^2 \log g)}\right)$	Courbes hyperelliptiques
Chapitre 5 (2017)	$\boldsymbol{O_g\left((\log q)^{O(g)}\right)}$	**Courbes hyperelliptiques**
Chapitre 7 (2018)	$\boldsymbol{\widetilde{O}_\eta\left((\log q)^8\right)}$	**Courbes hyp. avec RM**

Un autre aspect que nous étudions concerne l'utilisation pratique d'algorithmes inspirés par Schoof et Pila lorsque le genre est petit, ce qui va de pair avec la valeur exacte de l'exposant de $\log q$ à genre fixé. Bien que l'algorithme de Pila ne soit pas adapté à une implémentation directe, ce que Pila appelle une représentation réduite du Frobenius, c'est-à-dire la réduction du Frobenius modulo l'idéal de ℓ-torsion peut être calculé en pratique à l'aide de techniques standard issues du calcul formel. En genre 2, c'est précisément ce qui a été réalisé et implémenté par Gaudry, Harley et Schost dans [57, 60, 62]. Si la taille des objets à manipuler est sensiblement plus grande que dans le cas elliptique, cette approche est suffisamment efficace pour permettre la construction d'une courbe cryptographique de genre 2 définie sur un corps premier de taille 128 bits. Dans cette thèse, nous proposons une analyse heuristique concernant la faisabilité d'une telle courbe sur un corps de taille 192 bits, qui nous paraît peu probable en l'état actuel des choses. Les courbes à multiplication réelle (RM) explicite possèdent une structure supplémentaire qui permet, en genre 2, de ramener l'exposant de $\log q$ de 8 à 5 [59], atteignant ainsi une complexité semblable à l'algorithme de Schoof.

Une autre contribution de ce manuscrit s'intéresse ainsi aux courbes hyperelliptiques de genre 3 [2]. Cette fois, la taille de la ℓ-torsion rend les expériences pratiquement impossibles dès lors que ℓ dépasse 3. Cependant, pour des courbes munies d'une multiplication réelle explicite, les travaux de [59] s'étendent, moyennant quelques subtilités supplémentaires, avec une complexité asymptotique en $\widetilde{O}(\log^6 q)$, donc inférieure à celle du cas général en genre 2. Comme on pouvait s'y attendre avec une telle complexité, cet algorithme est utilisable en pratique, après quelques modifications. En particulier, nous calculons la fonction zeta locale d'une courbe hyperelliptique de genre 3 à multiplication réelle définie sur le corps premier $\mathbb{F}_{2^{64}-59}$, qui a donc une jacobienne de 192 bits. Notre algorithme s'adapte aisément en un algorithme de comptage de points sur des courbes sans multiplication réelle explicite, au prix d'une complexité bien plus grande en $\widetilde{O}(\log^{14} q)$, ce qui donne une réponse partielle aux interrogations sur la complexité de l'algorithme de Schoof-Pila en genre 3. Comme en genre 2, l'étape la plus coûteuse est la résolution du système polynomial décrivant la ℓ-torsion. Ce système est trivarié mais l'élimination successive à base de résultants est toujours suffisante pour atteindre une complexité quadratique en le degré de l'idéal. En pratique, cependant, nous mettons le système sous forme triangulaire en calculant une base de Gröbner avec les algorithme F4 [45] et FGLM [47]. Cette approche est bien plus

efficace en pratique, malgré des bornes de complexité théoriques bien plus difficiles à contrôler.

Table 4: Complexité asymptotique pour le calcul de fonctions zeta locales de courbes hyperelliptiques de genre ≤ 3

Genre	Complexité	Auteurs (année)
$g = 1$	$\widetilde{O}(\log^5 q)$	Schoof (1985)
$g = 1$	$\widetilde{O}(\log^4 q)$	Schoof-Elkies-Atkin ($\sim$ 1990)
$g = 2$	$\widetilde{O}(\log^8 q)$	Gaudry-Harley-Schost ($\sim$ 2000)
$\boldsymbol{g = 3}$	$\boldsymbol{\widetilde{O}(\log^{14} q)}$	**Chapitre 6 (2018)**
$g = 2$ avec RM	$\widetilde{O}(\log^5 q)$	Gaudry-Kohel-Smith (2011)
$\boldsymbol{g = 3}$ **with RM**	$\boldsymbol{\widetilde{O}(\log^6 q)}$	**Chapitre 6 (2018)**

Puisque la littérature présente de nombreux exemples de (familles de) courbes à multiplication réelle en genre quelconque [87, 23, 43, 101, 138], il est naturel de se demander quelles améliorations cette structure supplémentaire apporte à la complexité asymptotique lorsque g croît. Nous avons ainsi étendu les méthodes et les résultats obtenus en genre 2 et 3 pour créer un algorithm de comptage de points sur des courbes hyperelliptiques à multiplication réelle de genre arbitrairement grand. L'étape essentielle consiste à calculer non plus des résultants mais une résolution géométrique des noyaux d'endomorphismes de degré ℓ^2. Pour ce faire, nous adaptons la technique du Chapitre 5 qui était alors appliquée au noyau de la multiplication par ℓ, elle-même de degré ℓ^{2g} en tant qu'endomorphisme. Cette différence affecte notre modélisation en réduisant les degrés des équations de $O_g(\ell^3)$ à $O_g(\ell^{3/g})$. Ainsi, après vérification que les hypothèses sont toujours vérifiées et application de l'algorithme de résolution géométrique, nous atteignons une complexité en $O_\eta\left((\log q)^c\right)$, avec c une constante absolue et O_η dissimulant un terme dépendant de l'ordre par lequel la courbe a multiplication réelle (et donc également de g). Cela dit, nous insistons sur le fait que notre algorithme n'est pas polynomial en g et en $\log q$ parce que ce terme reste exponentiel en g. Nous analysons justement les raisons de cette dépendance exponentielle en g en espérant que de futurs travaux permettront de donner des bornes plus fines ou de remplacer ces étapes.

Résumé

Le comptage de points de courbes algébriques est une primitive essentielle en théorie des nombres, avec des applications en cryptographie, en géométrie arithmétique et pour les codes correcteurs. Dans cette thèse, nous nous intéressons plus particulièrement au cas de courbes hyperelliptiques définies sur des corps finis de grande caractéristique p. Dans ce cas de figure, les algorithmes dérivés de ceux de Schoof et Pila sont actuellement les plus adaptés car leur complexité est polynomiale en $\log p$. En revanche, la dépendance en le genre g de la courbe est exponentielle et se fait cruellement sentir même pour $g = 3$.

Nos contributions consistent principalement à obtenir de nouvelles bornes pour la dépendance en g de l'exposant de $\log p$. Dans le cas de courbes hyperelliptiques, de précédents travaux donnaient une borne quasi-quadratique que nous avons pu ramener à linéaire, et même constante dans le cas très particuliers de familles de courbes dites à multiplication réelle (RM).

En genre 3, nous avons proposé un algorithme inspiré de ceux de Schoof et de Gaudry-Harley-Schost dont la complexité, en général prohibitive, devient très raisonnable dans le cas de courbes RM. Nous avons ainsi pu réaliser des expériences pratiques et compter les points d'une courbe hyperelliptique de genre 3 pour un p de 64 bits.

Mots-clés: Courbes hyperelliptiques, comptage de points, méthodes ℓ-adiques.

Abstract

Counting points on algebraic curves has drawn a lot of attention due to its many applications from number theory and arithmetic geometry to cryptography and coding theory. In this thesis, we focus on counting points on hyperelliptic curves over finite fields of large characteristic p. In this setting, the most suitable algorithms are currently those of Schoof and Pila, because their complexities are polynomial in $\log q$. However, their dependency in the genus g of the curve is exponential, and this is already painful even in genus 3.

Our contributions mainly consist of establishing new complexity bounds with a smaller dependency in g of the exponent of $\log p$. For hyperelliptic curves, previous work showed that it was quasi-quadratic, and we reduced it to a linear dependency. Restricting to more special families of hyperelliptic curves with explicit real multiplication (RM), we obtained a constant bound for this exponent.

In genus 3, we proposed an algorithm based on those of Schoof and Gaudry-Harley-Schost whose complexity is prohibitive in general, but turns out to be reasonable when the input curves have explicit RM. In this more favorable case, we were able to count points on a hyperelliptic curve defined over a 64-bit prime field.

Keywords: Hyperelliptic curves, point counting, ℓ-adic methods.