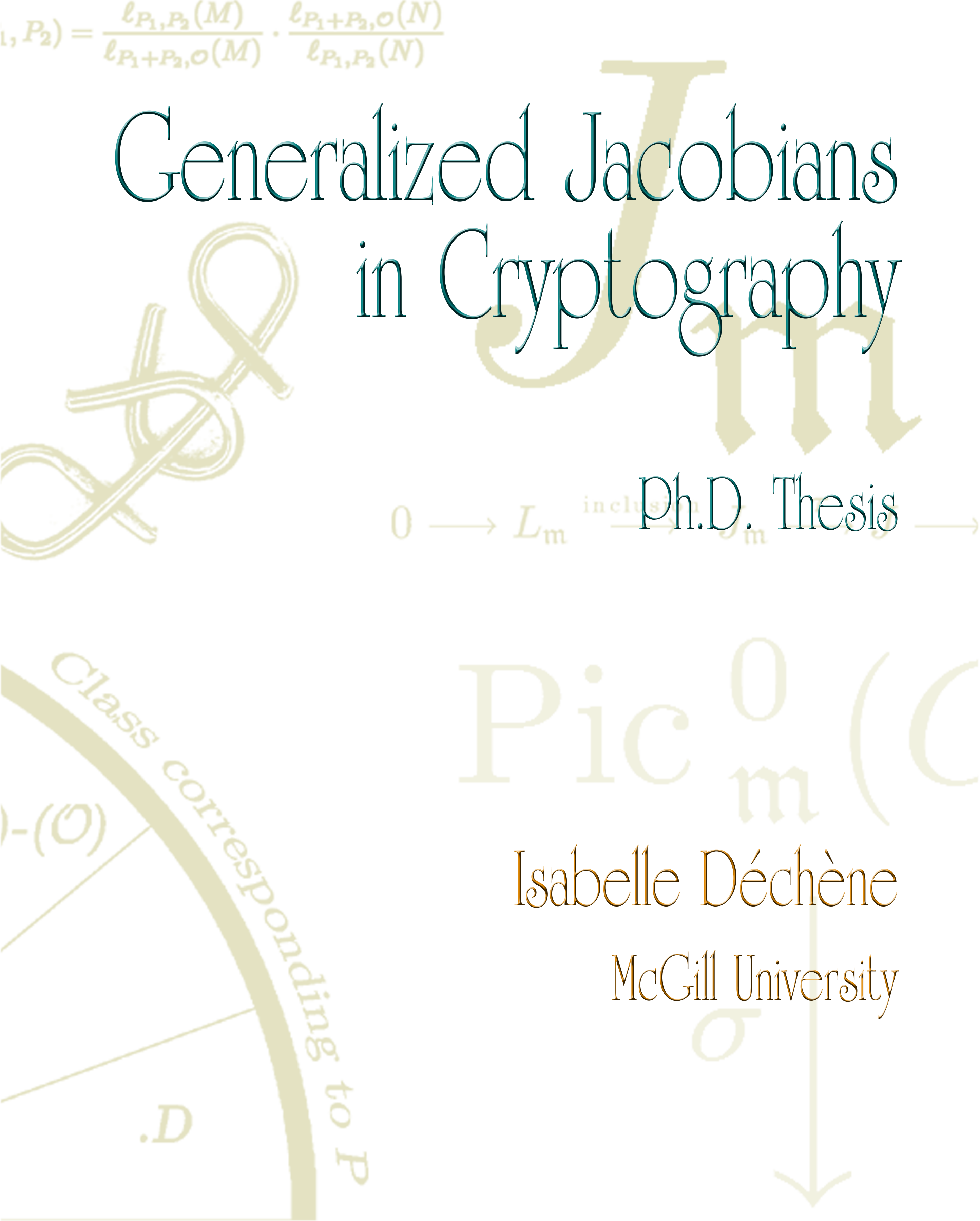


Generalized Jacobians in Cryptography

Ph.D. Thesis

Isabelle Déchène

McGill University

Generalized Jacobians in Cryptography

Ph.D. Thesis

Isabelle Déchène

Department of Mathematics and Statistics
McGill University, Montreal

A thesis submitted to McGill University in partial fulfilment
of the requirements of the degree of Doctor of Philosophy

September 2005

“C’est le temps que tu as perdu pour ta rose
qui fait ta rose si importante”

“It is the time you have devoted to your rose
that makes your rose so important”

- Antoine de Saint-Exupéry

To Daniel,
my best friend and my love,
who is always there for me.

Abstract

Groups where the discrete logarithm problem (DLP) is believed to be intractable have proved to be inestimable building blocks for cryptographic applications. They are at the heart of numerous protocols such as key agreements, public-key cryptosystems, digital signatures, identification schemes, publicly verifiable secret sharings, hash functions and bit commitments. The search for new groups with intractable DLP is therefore of great importance. The study of such a candidate, the so-called generalized Jacobians, is the object of this dissertation. The motivation for this work came from the observation that several practical discrete logarithm-based cryptosystems, such as ElGamal, the Elliptic and Hyperelliptic Curve Cryptosystems, XTR, the Lucas-based cryptosystem LUC as well as the torus-based cryptosystem CEILIDH can all naturally be reinterpreted in terms of generalized Jacobians. We next provide, from a cryptographic point of view, a global description of this family of algebraic groups that highlights their potential for applications. Our main contribution is then to introduce a new public-key cryptosystem based on the simplest nontrivial generalized Jacobian of an elliptic curve. This work thus provides the first concrete example of a semi-abelian variety suitable for DL-based cryptography.

Les groupes où le problème du logarithme discret est réputé difficile se sont avérés d'une importance capitale dans le développement d'applications cryptographiques. Ils sont au coeur de plusieurs protocoles tels les échanges de clés, les cryptosystèmes à clé publique, les signatures numériques, les procédés d'identification, les partages de secret publiquement vérifiables, les fonctions de hachage et les mises en gage de bits. La recherche de nouveaux groupes où le logarithme discret est difficile est donc d'une grande importance. L'étude de l'un de ces candidats, les Jacobiennes généralisées, fait l'objet de cette dissertation. Notre motivation vient de l'observation que plusieurs cryptosystèmes basés sur le logarithme discret, tels que ElGamal, les cryptosystèmes sur les courbes elliptiques et hyperelliptiques, XTR, le cryptosystème LUC utilisant les fonctions de Lucas ainsi que le cryptosystème CEILIDH reposant sur les tores algébriques, peuvent tous être naturellement réinterprétés en termes de Jacobiennes généralisées. En utilisant une approche cryptographique, nous présentons ensuite une description globale de cette famille de groupes algébriques mettant en lumière leur potentiel cryptographique. Notre principale contribution est alors de proposer un nouveau cryptosystème à clé publique basé sur la plus simple Jacobienne généralisée nontriviale d'une courbe elliptique. Nos recherches présentent donc le tout premier exemple d'une variété semi-abélienne pouvant concrètement être utilisée en cryptographie.

Preface: The Making of...

"Lo que vale, cuesta"
"Whatever is worthwhile, costs"

- Spanish saying

I learned in high school that one should write last what is to be read first. These lines, in other words. Well, it may have taken me an extremely long time to *really* understand why, but at least I think I know now. Indeed, I wrote so many lines in the past few months and still, I feel that I have more to say (peculiar, but true). You know, things that could not *'fit'* anywhere else, things that were simply too personal for the somehow rigid framework of a thesis. So here I am, tired but happy, taking some time to give myself the liberty to freely transgress all rules pertaining to formal scientific writing for a page or two (so yes, you have my blessing to skip this part, as it is absolutely not needed for the sequel).

Obviously, this document contains the final product of my doctoral thesis, all bright and shiny. The one thing that is missing though is how on earth did this old dream become tangible: 'The making of'.

When you liked a movie enough to take the time to listen to the extra features on a DVD, it seems that all *Making Of* have points in common: they first unveil the work of tons of people that work behind the cameras, away from the spotlights. It is also a good opportunity to show just how delicate it was to film a particular action sequence or that it took no less than forty takes to perfectly capture the emotion of the script. In a nutshell, they rarely say: *'It was a piece of cake! All fun and games!'* But since these documentaries are promotions for the movie aimed at increasing the ticket sales, it's another story to decide if they faithfully relate what really happened behind the scene...

In the present case, I (unfortunately) do not expect to become a zillionnaire any time soon by selling this thesis on eBay. I can thus ensure you that the following events really happened and that the names of the people and places have not been changed.

This thesis was written using the typesetting system LaTeX, while sipping a cup of Earl Grey in my cheap but comfy Montreal appartment, wearing a color-faded T-shirt[1], joggings and slippers (just like Kathleen Turner in the opening scene of Romancing the Stone, only much prettier). David Usher playing in background, and my lovely cats, Timide and Juliette, deeply asleep in their respective boxes[2]...

> *"So this is the story of a girl who really likes cryptography, teaching, swing dancing, laughing, home improvement, and gelato*[3]*. In the past few months, she was however hibernating and had (virtually) no time for her friends since she recently discovered something else that she likes: writing..."*

Now, that was a real surprise. I mean, I knew I loved to be in front of the class, but I had never thought that my teaching experience could ease the writing so much. Well I guess that explains the tone of this document: motivation for the problems and detailed explanations, surrounded by *'whatever works'* to make it enjoyable (or so I hope).

I know, I know, this is not what you want to hear. Okay, all right, I'll tell you. But at one condition: if you ever meet someone who is trying to find a good topic for a thesis, take a minute to tell him/her about the true story of the girl who liked gelato. Deal? Now listen carefully.

The most difficult part of this thesis was without a doubt to find a good problem to work on. Something original, ideally about elliptic curve cryptography, not too ambitious, and most importantly, something that *nobody* had done before. Hum... Was that too much to ask for?

At first, I was reading tons of papers and was overwhelmed by all these publications that seemed to be printed at the speed of light. I felt like I was running beside a train going at 100 mph and could not figure how to jump aboard. And when I tried to tackle several problems, most of the time going in circle, asking myself: *Is this really a dead end or is there a path to follow that I don't see yet? Is it wise to backtrack and try something else? Or am I simply giving up too soon?*

To come to the rescue, I have been lucky enough to have, not only one, but two incredible supervisors to help me out: Henri Darmon and Claude Crépeau. They were key actors in the cast and crew of people who helped me create what you are about to read.

[1] Like the purple and green one that says *"Camp Mathématique 1994"* (that I am simply unable to throw away).

[2] I dare to reveal these top secret details *only* because people naturally tend to think that all cryptographers work on supercomputers in a room that needs five access codes to get in...

[3] Not necessarily in this order.

So I wish to thank Claude Crépeau, who had enough faith in me for suggesting to become my co-supervisor. Even if he has so many students under his supervision, he was constantly there for me: always a phone call away when I needed help or advice, day, night, and even weekends. I also wish to thank him for letting me choose a topic that was kind of far from his speciality.

I also want to thank Henri Darmon, for his constant positive attitude and for all the encouragements he gave throughout the process. A true living encyclopedia, but all the while so humble, Henri is able to patiently explain the basic concepts with the same enjoyment as the deepest ones.

I am proud, honored and extremely grateful to have been able to work with such amazingly talented researchers, but most of all, who are also extremely kind human beings. *Claude, Henri, je vous adore!*

Huge thanks also go to the people from the Centre for Applied Cryptographic Research (CACR) at the University of Waterloo, especially Edlyn Teske and Alfred Menezes, with whom I had the pleasure to work for the last months: I have discovered in Waterloo a truly dedicated team of researchers; I learn a lot with you, and I am grateful that I could be within such a team.

To maman and papa, Denise and André, thank you for your unconditional love. Maman, thank you for letting me try my (sometimes messy) scientific experiments: from the moth balls that magically moved by themselves in a solution of vinegar and *p'tite vache*[4] to the one time when a glass bottle filled with water exploded in the freezer in the middle of the night. Papa, thank you for guiding me into discovering this world: I remember playing with you in the sand with a magnet and be amazed by the iron filings that are naturally present in the soil. And most of all, thank you for teaching me that "if something deserves to be done, then it deserves to be *well* done". Thanks also to my little sister Julie who has been my very first (and utterly patient) student, and who already knew, in grade 5, about the square root of -1. *Maman, papa, Juju: ensemble, nous formons une famille exceptionnelle.*

There are so many other people that I would like to thank. To all of you who closely or remotely contributed to the realization of this work, thank you for your time, your understanding and your generosity. A special thought goes to Tanja Lange who insisted that I attended the 2004 ECC Summer School: it was a truly memorable experience. Geneviève, I wish to thank you for your friendship, your homemade cookies that could say without a word "I am behind you", and for your wise Japanese advices. Now it is my turn to say *Gambatta koudasai!* to you. I also wish to thank the Magma team for their gracious developer's license, so that I could explore generalized Jacobians on the computer at will.

[4] This is how kids call sodium bicarbonate in Quebec, because of the cow drawing that used to appear on the box.

Finally, I wish to express my deepest gratitude to Daniel Lavoie. Dan, you were beside me at every moment, helping and supporting me in every way you possibly could (and sometimes even more). From cooking to proofreading this thesis; from fixing my (numerous!) computer problems to driving 1300 km *every* weekend to see me in Waterloo; and above all, for holding me tight, for your reassuring words and your one of a kind sense of humour, I wish to thank you. *Merci Daniel. Je t'aime.*

Isabelle Déchène
Montréal, Québec
September 2005

Contents

List of Figures

Chapter 1

Introduction

> *«People are going to steal from you. You can't stop them.*
> *But everybody has their own little personal security things - things that they think*
> *will foil the crooks, you know? In your own mind, right? ...You go to the beach,*
> *go in the water, put your wallet in the sneaker. Who's gonna know?*
> *What criminal mind could penetrate this fortress of security?*
> *"I tied a bow. They can't get through that". "I put the wallet down by the toe*
> *of the sneaker. They never look there. They check the heel, they move on".»*
>
> *- Jerry Seinfeld*

Nowadays, everyone is using security measures in their everyday lives: from the lock on the door to the car alarm or the account password, chances are that even before 9 am, most people will already have used several security mechanisms without even thinking about it. Some of them have the mandate to protect the confidentiality of information: this is where cryptography comes into play. Luckily, the implementation of cryptographic protocols are (usually) so "user-friendly" that virtually anyone can easily protect their personal data[1]. Cryptography then provides the necessary tools to avoid ad hoc methods (such as those often seen at the beach...).

1.1 The Context

For many, a day at work starts with the *coffee-and-email ritual*. Sadly, the initial excitement of the *"You've got mail"* has now faded drastically, thanks to the 22 new messages in your Inbox since 5pm yesterday. One of them perhaps contains your forgotten password while another, apparently sent by your bank, asks you to validate your personal data. In a world with 167,000,000

[1] Such an example is the freeware version of PGP (i.e. Pretty Good Privacy) available at http://www.pgpi.org. PGP is a public-key encryption program developed by Phil Zimmermann in the 1990s that now allows to encrypt email messages, transform a PC into a secure phone or encrypt the entire content of a hard drive.

users of Yahoo! Mail alone,

Who should we trust?

Happily, public-key cryptography is there to help protect us. Indeed, it was especially designed to be used by a large number of participants having access to an insecure communication channel (e.g. the Internet) in the presence of malicious parties. Loosely speaking, it allows the participants to:

- Encrypt messages that only the intended recipient can decrypt
- Affix a so-called *digital signature* to a message so that anyone can check whether it is an authentic signature or a forgery

The protocols used to achieve these tasks often rely on difficult computational problems, many of them inspired by number theory. Factoring integers and extracting discrete logarithms (DL) in a group are without a doubt the most famous hard problems used in public-key cryptography.

In a nutshell, this thesis aims at introducing generalized Jacobians (a family of groups known by mathematicians for over fifty years) as a new candidate for DL-based cryptography.

1.2 Motivation

The *sine qua non* security requirement on groups used for DL-based cryptography demands that the following computational problem be intractable:

Discrete Logarithm Problem (DLP)

Let G be a finite cyclic group generated by an element g.

$$\text{Given } h \in G, \text{ determine the smallest non-negative integer } k \text{ such that } g^k = h.$$

This integer is called the *discrete logarithm of* h (to the base g) and is denoted $\log_g h$.

Now, groups where the discrete logarithm problem is believed to be intractable are not only used to encrypt and signed messages [ElG85a, ElG85b]. They are also at the heart of various other protocols such as key agreements [DH76b], identification schemes [Sch91, Oka93], publicly verifiable secret sharings [Sta96], pseudo-random bit generators [Gen05], hash functions [CvHP92], and bit commitments [BCC88]. They are therefore inestimable building blocks for cryptographic applications.

Nevertheless, after nearly thirty years of research, only a handful of groups currently appear to be practical candidates for DL-based cryptography. This list includes the multiplicative group of a finite field, the invertible elements of $\mathbb{Z}_n$ with n a composite number, elliptic curves, Jacobians of hyperelliptic curves, algebraic tori as well as the ideal class group of an imaginary quadratic field. Another concern will always be the possibility that an efficient (classical) algorithm for solving the DLP in some (or all) of the above groups be discovered. The search for new groups with intractable DLP is therefore of great importance.

In 1985, the landmark idea of Koblitz [Kob87] and Miller [Mil86b] of using elliptic curves in public-key cryptography would, to say the least, change the perception of many on the tools of number theory that can be of practical use to cryptographers. In 1988, Koblitz [Kob89] generalized this idea by considering Jacobians of hyperelliptic curves, which then led to the broader study of abelian varieties in cryptography. Nearly fifteen years later, Rubin and Silverberg [RS03] discovered that another family of algebraic groups, namely the algebraic tori[2], also are of great cryptographic interest.

Now on one hand, Jacobians of curves (of small genus) gained the favor of many over the years, mostly because of the smaller key size needed. This attractive characteristic is in fact possible since we can easily generate curves for which there are no known subexponential-time algorithms for solving the corresponding discrete logarithm problem. On the other hand, rational algebraic tori over a finite field offer the convenient advantage of possessing a compact representation of their elements, which then decreases the amount of information needed to be exchanged.

In a nutshell, cryptographers like Jacobians of curves for their security and care about algebraic tori for their efficiency. Thus as far as we can tell, it appears that these two sub-families of algebraic groups somehow possess complementary cryptographic advantages. From a mathematical point of view, however, the overall picture looks quite different. Indeed, using a minimal background in algebraic geometry, they can both be seen as two realizations of a single concept: *generalized Jacobians*.

As a result, several existing DL-based cryptosystems, such as the ElGamal, the Elliptic and Hyperelliptic Curve Cryptosystems, XTR, the Lucas-based cryptosystem LUC as well as the torus-based cryptosystem CEILIDH all possess an underlying structure that can be naturally reinterpreted in terms of generalized Jacobians[3]. Figure 1.1 provides a simplified view of the

[2] Recall that an algebraic group defined over $\mathbb{F}_q$ which is isomorphic to $(\mathbb{G}_{\mathrm{m}})^d$ over some finite extension field is called an *algebraic torus* of dimension d over $\mathbb{F}_q$. As usual, $\mathbb{G}_{\mathrm{m}} \cong \left\{ x \in \mathbb{A}^1 \middle| x \neq 0 \right\}$ denotes the multiplicative group.

[3] The interpretation of XTR and LUC in terms of tori is due to Rubin and Silverberg [RS03, Section 7].

interrelation between the cryptosystems and their underlying structures. With this new unified approach, we could then assert that generalized Jacobians are a rich source of groups suitable for DL-based cryptography.

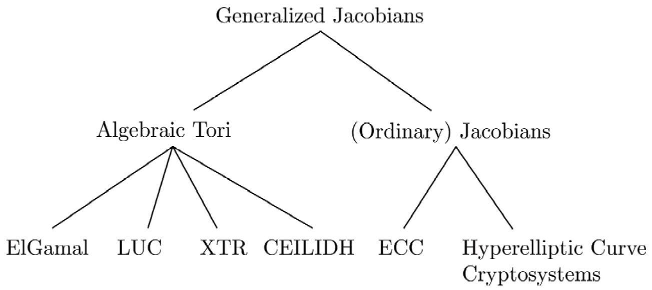


Figure 1.1: Relation between DL-based cryptosystems and generalized Jacobians

This observation then raised the following question at the heart of our research[4]:

Is it possible to use a generalized Jacobian that is neither a usual Jacobian nor an algebraic torus for DL-based cryptography?

An affirmative answer would then widen the class of algebraic groups that are of interest in public-key cryptography.

1.3 Our Work

In a word, the main contribution of this thesis is to confidently answer *yes* to the above fundamental question. This existence result was established by considering the simplest nontrivial generalized Jacobians of elliptic curves.

Before going any further, we present a brief overview of the construction of generalized Jacobian varieties [Ros52, Ros54, Ser88]. Let C be a smooth algebraic curve defined over an algebraically closed field K and $\mathfrak{m} = \sum_{P \in C} m_P(P) \in \mathrm{Div}(C)$ be an effective divisor[5], thereafter called a *modulus*. Two divisors D and D' of disjoint support with $\mathfrak{m}$ are said to be $\mathfrak{m}$-*equivalent*, and we write $D \sim_{\mathfrak{m}} D'$, if there exists an f in the function field of C such that $\mathrm{div}(f) = D - D'$ and $\mathrm{ord}_P(1-f) \geq m_P$ for each P in the support of $\mathfrak{m}$. Let $\mathrm{Pic}^0_{\mathfrak{m}}(C)$ be the group of $\mathfrak{m}$-equivalence classes of degree zero divisors having disjoint support with $\mathfrak{m}$. Then, there exists a commutative

[4] Afterall, generalized Jacobians had previously been used in coding theory [Gop88, Chapter 4], so their potential for practical applications had already been demonstrated (making them an even more attractive candidate).

[5] That is, each m_P is a nonnegative integer and only finitely many of them are nonzero.

algebraic group $J_{\mathfrak{m}}$, called the *generalized Jacobian* of C with respect to $\mathfrak{m}$, which is isomorphic to $\mathrm{Pic}^0_{\mathfrak{m}}(C)$.

The explicit family of generalized Jacobians that we considered can now be simply described as follows. Let E be a smooth elliptic curve defined over the finite field $\mathbb{F}_q$ with q elements and let $B \in E(\mathbb{F}_q)$ be a point of prime order l. Let also $\mathfrak{m} = (M) + (N)$, where M and N are distinct points of $E(\mathbb{F}_{q^r})$ such that $M, N \notin \langle B \rangle$, and $r \geq 1$ is a chosen integer. Finally, let $J_{\mathfrak{m}}$ be the generalized Jacobian of E with respect to $\mathfrak{m}$. Figure 1.2 illustrates the relationship between various structures of algebraic geometry in order to put these generalized Jacobians in perspective.

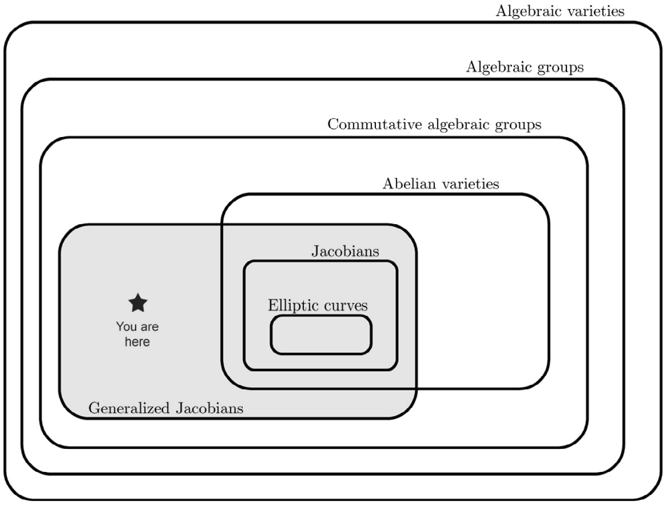


Figure 1.2: The generalized Jacobians in perspective

These test groups are in fact semi-abelian varieties which are extensions (of algebraic groups) of an elliptic curve by the multiplicative group $\mathbb{G}_{\mathrm{m}}$. Recall that a commutative algebraic group S is called a *semi-abelian variety* if there exists a short exact sequence of algebraic groups[6]

$$1 \to T \to S \to A \to 1,$$

where T is an algebraic torus and A is an abelian variety.

In order to put these groups to the test, there are several efficiency and security aspects to consider. Indeed, recall that there are four main requirements for a group G to be suitable for DL-based cryptography. Namely,

- The elements of G can be easily represented in a compact form,

[6] For information about extensions of algebraic groups, please refer to [Ser88, Chapter VII].

- The group operation can be performed efficiently,
- The DLP in G is believed to be intractable, and
- The group order can be efficiently computed.

In order to obtain a compact and convenient representation for the elements of $J_{\mathfrak{m}}$ and a group law algorithm using this representation, we first obtained an explicit bijection ψ of sets between $\mathrm{Pic}^0_{\mathfrak{m}}(E)$ and $\mathbb{G}_{\mathrm{m}} \times E$. Thus in this particular case, an element of $J_{\mathfrak{m}}$ can be viewed as a pair (k, P), where $k \in \mathbb{G}_{\mathrm{m}}$ and $P \in E$. The known addition on $\mathrm{Pic}^0_{\mathfrak{m}}(E)$ could then be used to endow, via ψ, the set $\mathbb{G}_{\mathrm{m}} \times E$ with the desired group structure. More explicitly, let (k_1, P_1) and (k_2, P_2) be elements of $J_{\mathfrak{m}}$ such that P_1, P_2, $\pm(P_1 + P_2) \notin \{M, N\}$. Then,

$$(k_1, P_1) + (k_2, P_2) = (k_1 k_2 \cdot \mathbf{c}_{\mathfrak{m}}(P_1, P_2), P_1 + P_2)\,,$$

where $\mathbf{c}_{\mathfrak{m}} : E \times E \to \mathbb{G}_{\mathrm{m}}$ is the 2-cocycle given by

$$\mathbf{c}_{\mathfrak{m}}(P_1, P_2) = \frac{\ell_{P_1,P_2}(M)}{\ell_{P_1+P_2,\mathcal{O}}(M)} \cdot \frac{\ell_{P_1+P_2,\mathcal{O}}(N)}{\ell_{P_1,P_2}(N)},$$

and $\ell_{P,Q}$ denotes the equation of the straight line passing through P and Q (tangent at the curve if $P = Q$).

As a consequence, $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is a finite subgroup of $J_{\mathfrak{m}}$ of order $(q^r - 1) \cdot l$ for which the elements are compactly represented and the group law is efficiently computable. In addition, we also described how to choose a suitable modulus, speed-up scalar multiplications and select parameters such that $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is a cyclic group.

As for security, as soon as $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is a cyclic subgroup of $J_{\mathfrak{m}}$, we obtain the following reductions among discrete logarithm problems:

The DLP in $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is at least as hard as the DLP in $\langle B \rangle \subseteq E(\mathbb{F}_q)$ and at least as hard as the DLP in $\mathbb{F}^*_{q^r}$.

Furthermore, extracting a discrete logarithm in $\mathbb{F}^*_{q^r} \times \langle B \rangle$ can always be performed by *sequentially* computing a discrete logarithm in E followed by one in $\mathbb{F}^*_{q^r}$. Moreover, it is possible to proceed in parallel when $l \nmid (q^r - 1)$, while this is still an open question in the case of curves suitable for pairing-based cryptography.

Finally, we have also investigated several scenarios involving precomputations in order to further study the DLP in $\mathbb{F}^*_{q^r} \times \langle B \rangle$. To this end, we empirically compared generalized Jacobians with the Classical Occupancy Problem. This preliminary study suggests that none of the proposed scenarios is faster than the known methods described above.

Thus from a practical point of view, these results imply that even though generalized Jacobians are newcomers in cryptography, we already know that solving their DLP cannot be easier than solving discrete logarithms in two of the most studied groups used in DL-based cryptography today.

1.4 Guided Tour of this Dissertation

Cryptographers come from various horizons, like engineering, computer science, physics and mathematics. As a result, their background knowledge greatly vary, which certainly contributes to the richness of this community. On a more down to earth consideration, it also inevitably implies that an accessible text in this domain should include a broader treatment of the underlying nuts and bolts. For this reason, we (tried to) rise to the challenge of writing a thesis that was as self-contained as possible. These lines were thus written with more than one targeted public in mind.

Chapter 2 is intended as a solid introduction to the numerous uses of discrete logarithms, written for scientists making their first steps in the universe of cryptography. From the classical Diffie-Hellman key-exchange to the elegant coin-flipping by telephone, this chapter covers the essentials of DL-based cryptography while relating its short but fascinating historical development[7].

Follows Chapter 3 on algebraic curves, which aims at allowing cryptographers having little or no background in algebraic geometry to learn more about the tools cryptographers "borrow" from algebraic geometry. More specifically, the first underlying objective is to concisely present the notions and results needed to understand the arithmetic of algebraic curves (and thus set the table for generalized Jacobians). The second wishes to give a flavor of the *methodology* followed to test the suitability of a group for DL-based cryptography. This goal is notably achieved by studying the simple hands-on example of the Pell equation[8].

Hence both Chapter 2 and 3 may be read independently from the rest of the text. We believe that Chapter 2 is accessible to motivated undergraduates, while Chapter 3 should be within reach of master's students in both mathematics and computer science.

Generalized Jacobians are finally presented in Chapter 4. In order to follow an approach by exploration, the emphasis is put on the cryptographic potential of these structures. The key ingredient in the construction of both usual and generalized Jacobians is the equivalence relation

[7] For instance, it seems that few people know that the secret-key cryptosystem of Pohlig-Hellman [PH78], which was proposed shortly before RSA [RSA78], can actually be seen as its direct ancestor.

[8] To the best of our knowledge, it is the first time that the Pell equation is used as an introduction to torus-based cryptography.

(on the divisors of the curve) one considers. Linear equivalence give rise to usual Jacobians, while $\mathfrak{m}$-equivalence characterize generalized Jacobians. Understanding the similarities and differences between them will help us choose the specific candidates we will put forward in Chapter 5. Lastly, our *coup de coeur* in this chapter is the concluding section presenting several cryptosystems falling in the spectrum of generalized Jacobians.

The (exciting) program of Chapter 5 is to introduce the first practical public-key cryptosystem based on a generalized Jacobian that is neither a torus nor a usual Jacobian. Starting from the abstract definition of generalized Jacobians in terms of divisor classes, we successively prove that all the basic requirements for a group to be suitable for DL-based cryptography are fulfilled. This therefore shows that generalized Jacobians are worth exploring towards the realization of new public-key cryptosystems.

Finally, we conclude with a quick summary in Chapter 6, which is of course followed by an extensive list of open problems for further work.

Chapter 2

The Discrete Logarithm and its Cryptographic Significance

"We stand today on the brink of a revolution in cryptography."
- Diffie & Hellman

This opening chapter aims at providing the cryptographic motivation towards the hunt for finite groups for which the group law is efficiently computable and its discrete logarithm problem seems intractable. It is really just a glimpse into the universe of cryptology and by no means a review of the literature of discrete logarithms in cryptography. Instead, we have selected classical protocols that, to our eyes, suffice to demonstrate what a powerful tool discrete exponentiation is for cryptographers. Here and there, we also tried to include a historical perspective in order to link seemingly unrelated problems (and hopefully keep awake readers who already saw this material an exponential number of times).

Everybody has an idea of what a cryptosystem is. Kids usually associate secret messages with spies[1], while adults are glad they exist so that they can safely shop online. So before we even skim over the subject, it might not be a bad idea to simply set things straight and recall the definition of a cryptosystem we will be working with:

Definition 2.1 *A* cryptosystem *is a quintuple* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, where* $\mathcal{P}$*,* $\mathcal{C}$ *and* $\mathcal{K}$ *are finite sets whose elements are respectively called* plaintexts *(or* clear texts*),* ciphertexts *and* keys*. Each key* $k \in \mathcal{K}$ *is associated with an* encryption rule $e_k : \mathcal{P} \to \mathcal{C}$ *in* $\mathcal{E}$ *and a* decryption rule $d_k : \mathcal{C} \to \mathcal{P}$ *in* $\mathcal{D}$ *such that* $d_k(e_k(m)) = m$ *for all* $m \in \mathcal{P}$*.*

[1] *'Are you a spy?'* is indeed the #1 question elementary school children ask me when I hold my workshop on secret messages.

2.1 The Holy Grail of Cryptography

Throughout this chapter, $(G, \circ)$ (or simply G), will denote a *group*. That is, a nonempty set G together with a binary operation $\circ : G \times G \to G$ satisfying

- $a \circ (b \circ c) = (a \circ b) \circ c$ for all $a, b, c \in G$ *(associativity property)*
- There is an $e \in G$ such that $e \circ a = a \circ e = a$ for all $a \in G$ *(existence of the identity)*
- For each $a \in G$, there is a $a' \in G$ satisfying[2] $a \circ a' = e$ *(existence of an inverse)*

Exactly, just how much algebraic background does one need in order to build an unbreakable cryptosystem? The integral of Hungerford's book [Hun74]? Not quite. In fact, the above three seemingly innocent properties suffice to ensure perfect secrecy.

Here is how it works. First, Alice and Bob take their favorite finite group G, say with n elements, and secretly agree on a (randomly chosen) element $k \in G$ which will serve as the key. Then, Alice chooses the message $m \in G$ she wishes to encrypt and sets the ciphertext to be $c = m \circ k$, which she then sends to Bob over an insecure channel.

An opponent, Eve, can then try to make deductions from the value c she eavesdropped. In other words, she wishes to know if there is *any information* about the message or the key leaking from the ciphertext.

Now, to the equally probable keys k_1, k_2, ..., k_n respectively correspond the potential messages $m_1 := c \circ k_1^{-1}$, $m_2 := c \circ k_2^{-1}$, ..., $m_n := c \circ k_n^{-1}$. Since G is a group, these n messages are distinct and so each element of G appears *exactly once* in this list. In other words, there are precisely n pairs (m_1, k_1), (m_2, k_2), ..., (m_n, k_n) of message/key that yields c as ciphertext. Therefore, given a ciphertext c and a uniform distribution on the keys, it is impossible for Eve to develop a bias towards or against any of the messages. That means that the knowledge of c is in fact useless to Eve, which is the best that one can hope for in a cryptosystem.

Also notice that a key *should never be used systematically.* For if $c = m \circ k$ and $c' = m' \circ k$, then Eve could eavesdrop c and c', compute $c' \circ c^{-1}$ which equals $m' \circ m^{-1}$. From the n^2 pairs of possible messages (m, m'), Eve can now narrow her search only to the n pairs satisfying $m' \circ m^{-1} = c' \circ c^{-1}$. As a result, a key should always be used only *once.*

This so-called *One-time Pad* was developed during World War I and was described by Gilbert Vernam in [Ver19, Ver26] using the letters of the English alphabet and the addition provided by the Vigenère square [Ker83]. Vernam claims that *"If [...] we employ a key composed of letters selected absolutely at random, a cipher system is produced which is* ***absolutely unbreakable****".* However, a formal proof could only be provided once Claude Shannon introduced the concept of *perfect secrecy* at the end of the 1940's [Sha48, Sha49].

[2] Notice that these three properties imply that $a' \circ a = e$ as well.

But despite its great elegance and simplicity, serious drawbacks arose in practice. For instance, the key needed to be as long as the message and since the keys were disposable, a huge amount of random (or nearly random) data needed to be generated. Some other difficulties, as we will see in the next section, were common to all secret-key cryptosystems as well.

From this point on, we will drop the cumbersome notation $(G, \circ)$ and will simply write G as a multiplicative group (that is, $a \circ b$ will now be written as ab or $a \cdot b$).

2.2 Limitations of Secret-key Cryptography

With any symmetric cryptographic system, no matter how efficient, there are certain problems that seem to be inevitable. In the late 60s, the idea that each of us would have a personal computer connected to the Internet and that we could find an ATM around every corner (respectively the so-called *'computer controlled communication network'* and *'remote cash dispensers'* of Diffie and Hellman[DH76b, p. 644]) was already in the foreseeable future of many. It was just a question of time before the need for secure communications would be required by ordinary citizens. From the government and the military to Alice and Bob, the typical user of cryptographic techniques would drastically change. However, secret-key cryptography was well-suited for a small number of participants only. Indeed, if the number of parties with no prior acquaintance was rather large, many partially or unanswered questions were left to be solved:

- *Key distribution.* The key must be known only by Alice and Bob. If they have access to a secure channel, then this problem is readily solved. But what if they don't?

- *Amount of keys.* A set of N persons want to be able to communicate two-by-two in a secure fashion. Then each of the $\binom{N}{2}$ distinct pairs of individuals have to share a key. So, a total of roughly N^2 keys must be shared and each person has to securely store $N - 1$ keys. Can this be improved?

- *Authentication and nonrepudiation (threat of dispute).* Say Bob agreed to lend money to Eve. In return, Eve sends back the encrypted message: *"I, Eve, hereby confirm that I owe 1000$ to Bob"*. Of course, Eve may wish to deny having sent such a message. But with secret-key systems, the key is also known by Bob, which means that he could have produced the message himself. How to ensure that Eve cannot deny having sent this message?

Certainly, it was conceivable that cryptographic protocols that would solve each of these problems *separately* could be designed, but who would have thought that a *single* concept could simultaneously solve them all...

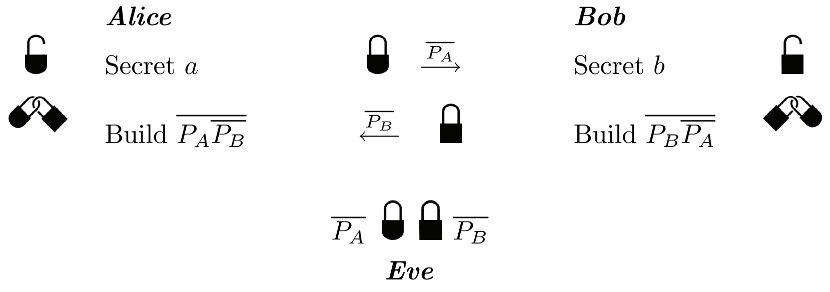
2.3 Key Agreement

For now, let's solely address the key distribution problem. Suppose that Alice and Bob only have access to an insecure channel and that they have no prior acquaintance (so that they do not already share a key), but are however able to mutually identify each other. They therefore want to agree on a key *only* by discussing over a public channel. Of course, by listening to this conversation, Eve must not be able to recover the key (and in an ideal world, not even a iota of information about it).

2.3.1 A Simple Model

But to realize such a scheme, what tools are we exactly looking for? Perhaps an easy visualization of this protocol could help. Assume that Alice has identical copies of a padlock P_A for which only she knows the secret combination a. Similarly, Bob possesses padlocks P_B for which he is the only one to know the corresponding combination b. Alice then gives a *closed* padlock $\overline{P_A}$ to Bob and he also sends a *closed* $\overline{P_B}$ to Alice:

Simple Model for a Key Agreement



Alice

Secret a

Build $\overline{P_A\overline{P_B}}$

$\xrightarrow{\overline{P_A}}$

$\xleftarrow{\overline{P_B}}$

Bob

Secret b

Build $\overline{P_B\overline{P_A}}$

$\overline{P_A}$ $\overline{P_B}$

Eve

Now, Alice can interlock P_A and $\overline{P_B}$ since she can close a P_A around the $\overline{P_B}$ received from Bob. Of course, Bob can also close a P_B around $\overline{P_A}$ and the resulting interlock owned by Alice and Bob will be *identical.* Moreover, Eve *only* has access to the two closed padlocks $\overline{P_A}$ and $\overline{P_B}$. Hence, it seems that the only way she could produce the interlock is to be able to open at least one of $\overline{P_A}$ or $\overline{P_B}$. So, in this setting, the interlock shared by Alice and Bob plays the role of the secret key.

Needless to say, padlocks are objects that are easy to close, but hard to open for anyone who does not know the secret combination. So the tool we are looking for has to be a *trapdoor one-way function*: easy to compute in one direction and hard to invert[3] *unless* you possess a sensitive piece of information, called the *trapdoor*. Of course, the trapdoor must always remain secret. Now comes the true challenge: finding such a function explicitly. In the next section, we make a brief digression in order to discuss one possible candidate and we will return to the key agreement problem in Section 2.3.3.

2.3.2 Discrete Exponentiations and Logarithms

It's no secret: cryptographic devices evolve with technology. So, after the widespread use of rotor machines[4] from the 30s to the 50s and their crucial role during World War II, they began to be replaced by cryptosystems based on *shift registers*. And with every novel approach, numerous interrogations arise. *"Given a possible state S of the register, how many shifts k were performed from the initial configuration I?"* is such a natural question. The process of recovering k from S is called *'solving a discrete logarithm problem'*. This problem can also be stated in an arbitrary group:

Discrete Logarithm Problem (DLP)
Let G be a finite cyclic group generated by an element g. Given $h \in G$, determine the smallest non-negative integer k such that $g^k = h$. This integer is called the *discrete logarithm of* h (to the base g) and is denoted $\log_g h$.

Hence, since the 1950s, discrete logarithms (DL) played a role in cryptography. As for the inverse operation, the (discrete) *exponentiation*

$$g^k := \underbrace{g \cdot g \cdot \ldots \cdot g}_{k \text{ times}}$$

can be computed much faster than the $k-1$ multiplications that the definition suggests. Actually, Indian mathematicians of circa 200 B.C. already had discovered a process that is still in use today. Their method is described in the Sanskrit book *Chandah-sûtra* of Acharya Pingala

[3] In an average-case sense. For exact definitions, please refer to [Gol01, Section 2.2].

[4] Like Enigma (German), Typex (British) or SIGABA (American).

and curiously, no trace of this rule was found outside of India for the next thousand years[5]. Incidentally, the oldest known description of the binary numbers is also attributed to Pingala.

Using today's terminology, their rule guarantees to compute g^k by performing *at most* $2\log_2 k$ group operations. This is easy to see. Let $(b_m b_{m-1} ... b_1 b_0)_2$ be the binary representation of k (with $b_m = 1$) so that $k = 2^m b_m + 2^{m-1} b_{m-1} + ... + 2b_1 + b_0$. Start with $g = g^{b_m}$ and successively compute

$$g^{b_m} \overset{\text{Square \& Multiply}}{\rightsquigarrow} (g^{b_m})^2 \cdot g^{b_{m-1}} \overset{\text{Square \& Multiply}}{\rightsquigarrow} (g^{2b_m + b_{m-1}})^2 \cdot g^{b_{m-2}} \overset{\text{Square \& Multiply}}{\rightsquigarrow}$$
$$(g^{2^2 b_m + 2b_{m-1} + b_{m-2}})^2 \cdot g^{b_{m-3}} \overset{\text{Square \& Multiply}}{\rightsquigarrow} ... \overset{\text{Square \& Multiply}}{\rightsquigarrow} (g^{2^{m-1} b_m + 2^{m-2} b_{m-1} + ... + b_1})^2 \cdot g^{b_0}.$$

As wanted, the last expression computed, $g^{2^m b_m + 2^{m-1} b_{m-1} + ... + 2b_1 + b_0}$, equals g^k. This technique is nowadays often referred to as the (left-to-right) *binary method* or *'square-and-multiply'*[6]. It hence provides an efficient algorithm [7] to perform discrete exponentiations in an arbitrary group G.

On the other hand, extracting discrete logarithms can be really easy in some groups and intractable in others. For example, in the *additive* group $\mathbb{Z}/n\mathbb{Z} = \{0, 1, 2, ..., n-1\}$ of integers modulo n, we have for $g = 1$ and any $h \in \mathbb{Z}/n\mathbb{Z}$,

$$h = \underbrace{1 + ... + 1}_{h \text{ times}} = h \cdot g,$$

so that the discrete logarithm $\log_g h = h$ is not hidden at all. But we also have that

Any two cyclic groups with the same number of elements are isomorphic.

For if C and D are two cyclic groups of order n generated by c and d respectively, the isomorphism between them is given by

$$\begin{array}{rccc} \varphi : & C & \rightarrow & D \\ & c & \mapsto & d \\ & c^2 & \mapsto & d^2 \\ & & ... & \\ & c^n & \mapsto & d^n \end{array} \tag{2.1}$$

[5] A fascinating historical account is depicted by Donald Knuth in [Knu81, Section 4.6.3], where references are given as well.

[6] If the group is written *additively* (as in the case of elliptic curves), it is sometimes also called the *'double-and-add'* method.

[7] Of course, it can be modified and improved in various ways, using signed representations, non-adjacent forms (NAF) or sliding windows for instance. See [Gor98] and [MvOV96, Sections 14.6, 14.7] for details as well as [BSS99, Section IV.2] for a comparison of several methods for elliptic curves.

This implies that for each positive integer n, *all* cyclic groups of order n are isomorphic to $(\mathbb{Z}/n\mathbb{Z}, +)$. In particular,

Every cyclic group is isomorphic to one for which solving the DLP is trivial

In algebra, we often regard two isomorphic groups as being *'the same'* since they carry the same structure. However, one must really be cautious when it comes to *computational* problems, as two isomorphic groups can behave quite differently. For example, the cost of the group operation in two isomorphic groups can greatly vary[8]. So the complexity of computational problems, like the DLP, crucially depends on the specific representation of the elements as well as the group law algorithm.

Then why don't we use the above isomorphism (2.1) to *'transport'* our problem to another group where it is easier to solve? Obviously, in order to have advantage to proceed this way in practice, the algorithm that computes this isomorphism must be *faster* than computing the discrete logarithm directly.

A really tempting instance is to try to compute an isomorphism φ from a given group $G = \langle g \rangle$ of order n (in which we want to solve DLPs) to the *additive* group $\mathbb{Z}/n\mathbb{Z}$. Let $a := \varphi(g)$, and so $\gcd(a, n) = 1$ (since a has to generate $(\mathbb{Z}/n\mathbb{Z}, +)$). Now, $\varphi(g^k) = ka$, which means that if we can solve DLPs in G, then we also know how to compute φ. Conversely, $k = \varphi(g^k) \cdot a^{-1}$ so that if we can evaluate φ, then with the simple help of the extended Euclidean algorithm (to compute a^{-1}), we can compute DLPs as well. Hence, we have that

The DLP in G is polynomial-time equivalent to explicitely computing the isomorphism φ.

So in this case, computing the isomorphism is not an easier way to proceed. However, this approach can sometimes work. This is in fact the successful idea behind the MOV[9] attack [MOV93]: to reduce the DLP for supersingular elliptic curves to the one in the multiplicative group of a finite field.

If we now go back to 1976, it was then known that the DLP in $\mathbb{F}_p^*$ appeared to be a really difficult problem (where $\mathbb{F}_p^*$ is the multiplicative group of the finite field $\mathbb{F}_p$ with a large prime number p of elements). In fact, the best known algorithms required roughly $\sqrt{p}$ operations. One such algorithm is due to Shanks [Sha71] (despite the fact that Diffie and Hellman are only citing Donald Knuth's *Art of Computer Programming* [Knu73, Exercise 5.25 with solution p.591] as reference). Another was the Pohlig-Hellman method [PH78] which was already submitted when the invited paper [DH76b] appeared, but was in fact only officially published in 1978. Thus,

[8] Just think about the relative cost of a *multiplication* in $\mathbb{F}_q^*$ compared to an *addition* in $\mathbb{Z}/(q-1)\mathbb{Z}$.

[9] Menezes-Okamoto-Vanstone

only exponential-time algorithms were known back then. In 1979, however, a subexponential-time algorithm was discovered by Adleman [Adl79]. Since then, the methods were of course diversified, improved and polished, but up to this date, no polynomial-time algorithm for solving this problem on a conventional computer is known (see Section 2.7 for details).

2.3.3 Diffie-Hellman Key Exchange Protocol

Recall that following the intuition given by the padlock analogy, we might be able to find a way to exchange a key over a public channel with the help of a trapdoor one-way function. On the other hand, we know that discrete exponentiations can be computed efficiently and that discrete logarithms seem to be hard for suitably chosen groups. These are so far the properties of a one-way function. We now need to determine the trapdoor, which has to enable its bearer to easily compute a particular instance of the DLP. But if Alice *first chooses* the value of the discrete logarithm, she can *then* easily use exponentiation in order to build the instance of the DLP that will be hard to solve for anybody but her. Thus, an easy strategy to create a trapdoor is to begin by choosing the *answer*, and then build a *tricky question* from it (just like creating a crossword puzzle). So Alice would do the following:

1. Pick the secret exponent a *(Alice first chooses her secret combination)*
2. Compute $h := g^a$ in private *(She closes her padlock)*
3. Make her challenge h public *(She challenges anybody to open it)*

Bob also performs steps 1-3 with his secret b. One last thing that needs to be done is to find how to *'interlock'* g^a and g^b such that:

1. The interlock computed by Alice and Bob must agree (they want to share the *same* key).
2. It must be (computationally) unfeasible for Eve to recover the interlock.

Knowing a is the *only* advantage that Alice has over Eve: she then *has to* use it when computing the interlock. Hence, Alice needs to combine a and g^b in a nontrivial fashion. Similarly, Bob has to combine b and g^a. Two easy candidates for the interlock are $g^{a+b} = g^{b+a}$ and $g^{ab} = g^{ba}$. The first choice is instantly ruled out since $g^{a+b} = g^a \cdot g^b$ can also be computed by Eve. As for the second choice, can one easily compute g^{ab} from g^a and g^b? The obvious strategy for Eve would be to recover a from g^a and then compute $(g^b)^a$. So we really want the DLP in G to be as hard as possible. What else can Eve do? Nothing obvious, at least. We will come back to this question shortly. But first, let's write down properly what we have so far.

From the above discussion, Alice and Bob can publicly agree on a key by first choosing a cyclic group G with generator g and then by exchanging g^a and g^b. And throughout this process, only the values of a and b need to be secret. As soon as the key $k = g^{ab}$ has been computed by both parties, Alice and Bob are free to use it with any secret-key cryptosystem they like.

Diffie-Hellman Key Exchange Protocol (DHKE)

$$\begin{array}{lcl} \textbf{\textit{Alice}} & & \textbf{\textit{Bob}} \\ \text{Private } a & \xrightarrow{g^a} & \text{Private } b \\ k = (g^b)^a & \xleftarrow{g^b} & k = (g^a)^b \end{array}$$

Once written in such a compact form, this really clever idea often seemed like *'the obvious thing to do'*. But when venturing in new territories, it was everything but obvious. In May 1975, Whitfield Diffie had the revolutionary idea of splitting the key into a public and a private part. The conference paper *'Multiuser Cryptographic Techniques'* [DH76a] was written with Martin Hellman in December that year, and still no concrete realization of the scheme was known. In the spring of 1976, Pohlig and Hellman were putting the final touch to their paper [PH78], where they used discrete exponentiation to build a secret-key cryptosystem. And in May 1976, Hellman realized how to use exponentiation to build the key exchange. This was just before the submission of the *New Directions in Cryptography*[10] and right on time for their first official public disclosure of their results at the National Computer Conference on June 8th. So a whole year had passed between the spark of genius and the explicit algorithm...

Now, the security of this elegant protocol relies on the difficulty of solving the *'Computational Diffie-Hellman Problem'*.

Computational Diffie-Hellman Problem (CDHP)
Let G be a finite cyclic group generated by an element g. Given G, g, g^a and g^b, determine g^{ab}.

As noticed above, this problem is no harder than the discrete logarithm problem. That is, CDHP$\leq_P$DLP. On the other hand, suppose that we can solve the CDHP. Then does this yields a method to solve the DLP? In general, this is an open question: we simply do not know if these two problems are polynomially equivalent. However, at CRYPTO '94, Ueli Maurer [Mau94] gave strong evidence of this equivalence[11], which was then refined with the collaboration of Stefan Wolf [MW96] at CRYPTO '96[12]. Antoine Joux and Kim Nguyen subsequently used their work

[10] Whose manuscript was received on June 3rd.

[11] using a modified version of Lenstra's elliptic curve method for factoring integers [Len87].

[12] A journal version of this work is also available [MW99].

in order to give *concrete examples* of certain elliptic curve groups where the two problems are provably equivalent [JN03].

We now briefly return to the possible attacks that Eve might try *under the assumption that she cannot solve discrete logs*. In the vast majority of the cases, we do not know whether the CDHP and the DLP are equivalent or not. This implies that under our assumption, no one has been able to devise an efficient algorithm to solve the CDHP[13], not even Eve. Hence for a *passive adversary* which merely listens to the conversation, the only known efficient attacks require solving discrete logarithms.

In the case where Eve is an *active adversary* and can *'manipulate'* the data transmitted between Alice and Bob, the situation is quite different. One possible game that Eve can play is the so-called *man-in-the-middle* attack. This is similar to the trick where in a completely dark room, Alice and Bob think that they are shaking each other hands, while in reality they are both shaking Eve's hands who is standing between them. Hence, Eve's strategy is to intercept the data and replace it with her own.

Man-in-the-middle attack

Alice		***Eve***			***Bob***
Private a	$\xrightarrow{g^a}$		Private a'	$\xrightarrow{g^{a'}}$	
	$\xleftarrow{g^{b'}}$	Private b'		$\xleftarrow{g^b}$	Private b
$k' = (g^{b'})^a$		$k' = (g^a)^{b'}$	$k'' = (g^b)^{a'}$		$k'' = (g^{a'})^b$

In doing so, Eve now shares k' with Alice and k'' with Bob. However, Alice and Bob no longer share k. Eve can then send encrypted messages to Alice using k' and chances are that Alice will believe that the message really came from Bob. And with the help of k'', Eve can also impersonate Alice to Bob. When agreeing on a key, Alice and Bob should then be able to verify that the data they received truly came from the other party. In such an *authenticated key agreement* scheme, Eve will therefore no longer be able to perform a man-in-the-middle attack.

For instance, digital signatures were used in the Station-to-station Protocol (STS) of Diffie, van Oorschot and Wiener [DvOW92] in order to modify the classical Diffie-Hellman and achieve authentication. The MTI key agreements protocols of Matsumoto, Takashima and Imai [MTI86] are modifications of the original scheme as well. Their technique exploits the idea of an *implicit key authentication* which does not rely on digital signatures.

[13] Since otherwise, this would show that the two problems are not equivalent!

There is yet another way to ensure that Eve cannot act as a man-in-the-middle. Recall that the *'partial keys'* g^a and g^b can be made public without any problem. Suppose we have a trusted (read only) public directory containing the name and corresponding partial key for each participant. Then, we can think of this directory as a *predistribution* of the keys since now, Alice can compute the key $(g^b)^a$ without the help of Bob. Of course, this is no longer a true interactive key agreement since the key shared by Alice and Bob can no longer be changed at will.

Diffie-Hellman Key Predistribution

Alice		***Bob***
Private a		Private b
Message m		
$k = (g^b)^a$	$\xrightarrow{e_k(m)}$	$k = (g^a)^b$
		$d_k(e_k(m)) = m$

Public Directory	
Name	**Key**
Alice	g^a
Bob	g^b

This predistribution scheme was also described in the landmark paper [DH76b] and this slightly different way of regarding this protocol really highlights the *public-key nature* of this algorithm.

We have here described the key-exchange in a group G, which seems to be a prerequisite to build such a scheme. However, at CRYPTO '89, Buchmann and Williams [BW90] described the first version of the DH key-exchange that did not need an underlying group structure. This surprising result was achieved by using real quadratic fields and is described in much details in the Journal of Cryptology version [BSW94], with co-author Renate Scheidler.

2.4 Public-key Cryptosystems

Following Auguste Kerckhoffs' second *'desideratum de la cryptographie militaire'* [Ker83]:

> *"Il faut qu'il (le cryptosystème) n'exige pas le secret, et qu'il*
> *puisse sans inconvénient tomber entre les mains de l'ennemi".*

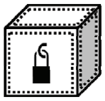
That is, the cryptosystem must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience. Therefore, *all* the security must reside in the key. So for a really long time, people thought that keeping the key *entirely secret* was a sine qua non condition to ensure secrecy. Whitfield Diffie didn't think that way. His audacious idea of splitting the key into two parts such that revealing the first part did not compromise the second truly deserved the title of a*new direction in cryptography* [DH76b].

2.4.1 A Simple Model

As with key agreements, an interesting analogy with padlocks can be made. Just as before, Bob has identical copies of a padlock P_B for which he is the only one to know the secret combination b. Bob then puts several copies of his *open* padlock at the disposal of anyone who would like to send him secured messages. Notice that the fact that open padlocks are publicly available does not compromise b. Now, if Alice wants to send a message m to Bob, she first gets Bob's open padlock P_B from a *reliable* source[14]. She then places m in a safe, locks it with P_B and sends it to Bob. Finally, Bob is the only one who can recover m since he is the unique person to know b.

Simple Model for a Public-key Cryptosystem

Alice
Private a
Get Bob's open padlock P_B
Put message m in safe,
then close P_B



⟶

Bob
Private b
Open the safe
using b
and recover m

Public Supply	
Name	**Padlocks**
Alice	🔓🔓... 🔓
Bob	🔓🔓... 🔓

This simplified view thus suggests that each user should now have a private key k, which is kept secret, and a public key K which is known to everyone. It must then be computationally infeasible to recover k from K. If Alice wishes to send a message m to Bob, she then simply looks up Bob's public key K_B from a trustable source and then encrypts m with the public encryption function $e_{K_B}(m)$. In turn, Bob can recover the plaintext by applying the decryption function $d_{k_B}(e_{K_B}(m)) = m$. Thus, each user must be able to create a pair of keys (k, K) such that $d_k(e_K(m)) = m$ for all possible messages. Of course, this has to be done by either computing K from k or by choosing them simultaneously[15]. Because of this dual key system, public-key systems are often referred to as *'asymmetric cryptographic systems'*, in opposition to symmetric key systems where the *same* key is used to encrypt and decrypt messages.

So now, Alice and Bob no longer have to share the same key. In fact, since the cryptosystem itself is publicly known, K_B is the only other piece of information that is needed in order to send messages to Bob. This implies that a newcomer can send encrypted messages to Bob without even creating keys for himself. This property certainly contrasts with secret-key cryptography and with the predistribution scheme of the previous section.

Another property of asymmetric systems is that once Alice has encrypted her message for

[14] Since otherwise, Eve might try to give her own padlock to Alice and make her believe that it is in fact Bob's padlock.

[15] Since we assumed that it is not possible to compute k from K.

Public-key Cryptography (PKC)

Alice		***Bob***
Private k_A		Private k_B
Message m	$\xrightarrow{e_{K_B}(m)}$	$d_{k_B}(e_{K_B}(m)) = m$

Public Directory	
Name	**Public Key**
Alice	K_A
Bob	K_B

Bob, she is no longer able to recover the plaintext from it. Hence, if she wants to keep a copy of the message, she should either keep a copy of the plaintext, or for more security, store $e_{K_A}(m)$ (instead of $e_{K_B}(m)$).

2.4.2 Pohlig-Hellman Secret-key Cryptosystem and RSA

Although the concept of public-key cryptography was crystal clear in the minds of their inventors, they were unfortunately unable to find a concrete scheme to include in their 1976 papers. In [DH76a], they declared with a shrug

> *"At present, we have neither a proof that public-key*
> *systems exist, nor a demonstration system"*

But as we now see, they were in fact really, really close to a positive answer. The manuscript of the Pohlig-Hellman paper [PH78] was submitted only two weeks after [DH76b][16] had been. We often think of [PH78] as being an algorithm for computing discrete logs, but the paper also contained a *secret-key* cryptosystem *based on discrete logarithms* in $\mathbb{F}_p^*$. The key comprised the two elements d and e between 1 and $p-1$ such that

$$de \equiv 1 (\bmod\, \phi(p)).$$

The encryption and decryption rules on a message m and corresponding ciphertext c were performed as follows:

$$c = m^e \bmod p \quad \text{and} \quad m = c^d \bmod p.$$

This is already similar to a PKC since the key is split in two parts. However, if the value of e is revealed, then

$$d = e^{-1} (\bmod\, \phi(p))$$

[16] Notice that even if the article of Pohlig and Hellman was submitted in June 1976, it was only officially published in January 1978.

is easy to recover since $\phi(p) = p - 1$ is trivial to compute from the public value p. Hence, the prime p has to be replaced by a *composite* integer n such that computing Euler's totient function from n is computationally unfeasible. Recall that

$$\phi(n) = \left(p_1^{\alpha_1} - p_1^{\alpha_1 - 1}\right) \cdot \left(p_2^{\alpha_2} - p_2^{\alpha_2 - 1}\right) \cdot \ldots \cdot \left(p_N^{\alpha_N} - p_N^{\alpha_N - 1}\right), \tag{2.2}$$

where $p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_N^{\alpha_N}$ is the prime factorization of n, the p_i's are distinct and each $\alpha_i > 0$. From (2.2), we see that computing ϕ is easy if the factorization of n is known.

But what if the factorisation of n is not known?

The answer to this simple question was indeed the bridge between secret and public-key cryptography. In the simplest case where $n = pq$ is the product of two distinct primes $p < q$, we have that

$$\phi(n) = (p-1)(q-1) = (p-1)\left(\frac{n}{p} - 1\right) = \frac{(p-1)(n-p)}{p} = \frac{-p^2 + np + p - n}{p},$$

and so $p^2 - (n - \phi(n) + 1)p + n = 0$. Thus, if *both* n and $\phi(n)$ are known, then

$$p = \frac{(n - \phi(n) + 1) - \sqrt{(n - \phi(n) + 1)^2 - 4n}}{2} \text{ and } q = \frac{(n - \phi(n) + 1) + \sqrt{(n - \phi(n) + 1)^2 - 4n}}{2}.$$

Hence, if Eve is able to compute $\phi(n)$ from the public value n, then she is able to factor n as well.

Enters Donald Knuth, who was at Stanford just like Diffie, Hellman and Pohlig at the time. Knuth raised the idea that since multiplication was an easy task but factorization appeared hard, it could be a good candidate for a one-way function at the heart of a public-key cryptosystem [Lev01, p.83].

So under the assumption that factorisation of $n = pq$ is computationally out of reach, Eve is unable to calculate $\phi(n)$ and there is therefore no obvious way to compute the decryption exponent d, even if e is publicly known.

However, the MIT group composed of Rivest, Shamir and Adleman was the first to put the pieces of the puzzle together. In their paper, they even acknowledge the great similarity between the two cryptosystems: *"Pohlig and Hellman study a scheme related to ours, where exponentiation is done modulo a prime number"* [RSA78, p.123]. For a concise treatment of exponentiation ciphers where Pohlig-Hellman and RSA are seen as two realizations of the same principle, see [Den82, Section 2.7].

In conclusion, the Stanford group not only invented public-key cryptography: they also set the table for its first concrete implementation.

2.4.3 ElGamal Encryption

We keep the notation of the previous section. We have just seen how the Pohlig-Hellman secret-key cryptosystem can be modified in order to yield the public-key system RSA. Now, the security of the Pohlig-Hellman scheme relies on the discrete logarithm problem in $\mathbb{F}_p^*$. For if Eve knows a plaintext-ciphertext pair (m, c), she *must* solve $c = m^e \bmod p$ (that is, compute $\log_m c$) in order to recover the secret value e. But when we convert this scheme into a PKC by replacing p by n and publishing e, the value of $\log_m c$ is now *known to everyone*, so that the resulting protocol is *no longer based on discrete logarithms*. In fact, a necessary condition for RSA to be secure is that it must be computationally infeasible to factor n (otherwise, $\phi(n)$ can be obtained and the private key d, computed). The goal of this section is then to describe a public-key cryptosystem whose security depends on the intractability of the DLP.

As usual, G is a finite cyclic group of order n generated by g. In order to be able to perform the computations, we want G and g to be publicly known and of course, we assume that the DLP in G is intractable. Now, a natural choice for Alice would be to secretly choose an exponent a as her *private key*, compute g^a and make it her *public key*. We now are in the following situation:

Towards a Discrete Logarithm Based PKC

$$\begin{array}{lcl}
\textbf{\textit{Alice}} & & \textbf{\textit{Bob}} \\
\text{Private } a & & \text{Private } b \\
\text{Encrypt } m \text{ using } g^b & \xrightarrow{c} & \text{Decrypt } c \text{ using } b \\
\text{to get ciphertext } c & & \text{and recover } m
\end{array}$$

Public Directory	
Name	**Public Key**
Alice	g^a
Bob	g^b

The big question is *how* to encrypt m using g^b. We might try the same strategy as in the previous section: start with an existing private-key system and try to convert it into a public-key one. Plus, we already know that the Diffie-Hellman key agreement uses discrete exponentiations and that once the key is exchanged, we can use any secret-key cryptosystem we like:

$$\begin{array}{lcl}
\textbf{\textit{Alice}} & & \textbf{\textit{Bob}} \\
\text{Private } a & \xrightarrow{g^a} & \text{Private } b \\
k = (g^b)^a & \xleftarrow{g^b} & k = (g^a)^b \\
\text{Encrypt } m \text{ using } k & \xrightarrow{c} & \text{Decrypt } c \text{ using } k \\
\text{to get ciphertext } c & & \text{and recover } m
\end{array}$$

First, we need to get rid of the step where Bob sends g^b to Alice. This is easy since g^b is Bob's public key and so Alice can retrieve this value directly from the directory:

Alice		***Bob***
Private a	$\xrightarrow{g^a}$	Private b
$k = (g^b)^a$		$k = (g^a)^b$
Encrypt m using k	$\xrightarrow{c}$	Decrypt c using k
to get ciphertext c		and recover m

Public Directory	
Name	**Public Key**
Alice	g^a
Bob	g^b

Next, the key employed should not depend on Alice's private key, but only on g^b. One possibility would be to use another secret element, say r, instead:

Alice		***Bob***
Private a		Private b
Secret r	$\xrightarrow{g^r}$	
$k = (g^b)^r$		$k = (g^r)^b$
Encrypt m using k	$\xrightarrow{c}$	Decrypt c using k
to get ciphertext c		and recover m

Public Directory	
Name	**Public Key**
Alice	g^a
Bob	g^b

Alice is free to choose any value of r, so she could certainly randomly pick a new one for every encryption. That way, a fresh new key k would be used each time. So Alice could try to use k as a one-time pad:

Alice		***Bob***
Private a		Private b
Randomly pick r	$\xrightarrow{g^r}$	
$k = (g^b)^r$		$k = (g^r)^b$
$c = m \cdot k$	$\xrightarrow{c}$	$c \cdot k^{-1} = m$

Public Directory	
Name	**Public Key**
Alice	g^a
Bob	g^b

As a bonus, we even get that this encryption function is truly economical since only one group operation is needed once k is known. Moreover, even though Alice and Bob share the same k, we are really in presence of a PKC: Alice computes $c = m \cdot (g^b)^r$ with the help of Bob's public key, and Bob performs $c \cdot (g^r)^{-b} = m$ with his private b. We are then ready to write down the final version:

ElGamal Public-key Cryptosystem

Alice		***Bob***
Private a, $1 \le a \le n-1$		Private b, $1 \le b \le n-1$
Randomly pick r, $1 \le r \le n-1$	$\xrightarrow{g^r}$	
$c = m \cdot (g^b)^r$	$\xrightarrow{c}$	$c \cdot (g^r)^{-b} = m$

Public Directory	
Name	**Public Key**
Alice	g^a
Bob	g^b

Finally, we turn our attention to the security aspect. Since the value of c is known by Eve, she will be able to deduce the plaintext m if and only if she can compute g^{rb}. Therefore, her task is to compute g^{rb} from g^r and g^b. In other words, she has to solve an instance of the computational Diffie-Hellman problem (p.17) in G.

This cryptosystem was presented at CRYPTO '84 by Taher ElGamal [ElG85a, ElG85b], who was also at Stanford at the time[17]. This simple and elegant scheme is easy to remember since it can be thought of as *'a key-exchange followed by a one-time pad'.* And even if these two primitives were known by cryptographers in 1976, ElGamal was the first to make the connection and to propose this *randomized* encryption method. This aspect is certainly an advantage of this cryptosystem. Notice that with deterministic encryption, Eve could tell with certainty if an observed ciphertext c is the encryption of a specific message m_0 (by means of the public encryption rule). This is no longer true here since each message now corresponds to many possible ciphertexts (depending on the choice of r).

2.5 Digital Signatures

In the previous sections, we have seen how the Diffie-Hellman key agreement provides a solution to the key distribution problem. As well, the predistribution scheme and public-key cryptosystems have the property that each of the N users now has *only one* key to keep secret, instead of the $(N-1)$ needed in a conventional secret-key setting. However, we still have to solve the authentication and nonrepudiation problem. This is the object of this section.

2.5.1 Digital Signatures from a Public-key Cryptosystem

According to Diffie and Hellman, *"Any public-key cryptosystem can be transformed into a one-way authentication system*[18]*"* [DH76b, pp.645, 650]. Loosely speaking, the idea is to turn the cryptosystem *'on its head'.* This is done as follows.

First recall that with any public-key cryptosystem, Bob can send an encrypted message to Alice with the help of her public key. Alice then uses her secret key in order to invert the process and recover the plaintext. That way, anyone can send enciphered messages to Alice but she is the only one who can decipher them.

With digital signatures, the situation is somewhat reversed. We now need the signer of the message, Alice, to be the unique individual able to produce the corresponding signature. That is, the signature cannot be forged. And just like a paper-and-pencil signature, anyone should be

[17] In addition to the cryptosystem, his paper also contains a digital signature scheme that will be presented in Section 2.5.2.

[18] a.k.a. *digital signature.*

able to verify its validity[19]. As a result, it makes perfect sense to use the private key to produce the signature and the corresponding public key to check its authenticity. Also notice that the signature must be message dependant. Otherwise, Eve could simply copy and paste Alice's signature and append it to the message of her choice. So, unlike a classical signature, Alice's digital signature on two distinct messages will look completely different (but can nevertheless be verified by anybody).

So for a given public-key cryptosystem with set of possible plaintexts $\mathcal{P}$ and ciphertexts $\mathcal{C}$, the encryption $e_A : \mathcal{P} \to \mathcal{C}$ and decryption rule $d_A : \mathcal{C} \to \mathcal{P}$ of Alice are such that

$$\begin{array}{ccccc} & \text{Encryption} & & \text{Decryption} & \\ \mathcal{P} & \xrightarrow{e_A} & \mathcal{C} & \xrightarrow{d_A} & \mathcal{P} \\ m & \longmapsto & e_A(m) & \longmapsto & d_A(e_A(m)) = m \end{array}$$

for any message $m \in \mathcal{P}$. Now, to produce the digital signature, we wish to use both the secret key and the message. But how? An easy solution would be to compute $d_A(m)$, but this can only be done if $\mathcal{P} \subseteq \mathcal{C}$. If it is the case, we can set the signature on message m to be $d_A(m)$. To verify its validity using the public key, we could then compute $e_A(d_A(m))$, which is possible if $\mathcal{C} \subseteq \mathcal{P}$. Hence, if $\mathcal{P} = \mathcal{C}$, both $d_A(m)$ and $e_A(d_A(m))$ make sense. The last step is to verify that $e_A(d_A(m)) = m$ for any $m \in \mathcal{P}$.

But with any public-key cryptosystem, the encryption rule e_A is one-to-one (for if $e_A(m) = e_A(m')$, then $m = d_A(e_A(m)) = d_A(e_A(m')) = m'$). Hence, e_A has to be onto here as well since $\mathcal{P} = \mathcal{C}$ is a finite set. Now, if $e_A(d_A(m)) = m'$, then $d_A(m) = d_A(m')$ and there are $c, c' \in \mathcal{P}$ such that $m = e_A(c)$ and $m' = e_A(c')$. So, $c = d_A(e_A(c)) = d_A(e_A(c')) = c'$, which finally implies that $m = m'$.

Hence, as long as $\mathcal{P} = \mathcal{C}$, we have that $e_A(d_A(m)) = m$ for any $m \in \mathcal{P}$ and so the signature generation and verification can be performed as follows:

$$\begin{array}{ccccc} & \text{Signature Generation} & & \text{Signature Verification} & \\ \mathcal{P} & \overset{d_A}{\to} & \mathcal{P} & \overset{e_A}{\to} & \mathcal{P} \\ m & \mapsto & d_A(m) & \mapsto & e_A(d_A(m)) = m \end{array} .$$

So given a message m, Alice can compute the corresponding signature $d_A(m)$ which she then transmits to Bob together with m. Bob then accepts Alice's signature iff $e_A(d_A(m)) = m$.

Hence, it is possible to easily produce digital signatures from a public-key cryptosystem as soon as $\mathcal{P} = \mathcal{C}$. The famous example of course being the RSA signature scheme [RSA78].

[19] In some other specific applications, it is desirable to require that the collaboration of the signer be required in order to validate signatures. These so-called *undeniable signatures* were introduced by Chaum and van Antwerpen at Crypto 89 [CA89] and once more, the DLP is at the heart of their scheme.

But what happens when $\mathcal{P} \neq \mathcal{C}$? In the original definition of a public-key cryptosystem [DH76b, p.648], it is assumed that $\mathcal{P} = \mathcal{C}$, so that explains the claim that *any* PKC could be turned into a signature scheme. Now, if $\mathcal{P} \neq \mathcal{C}$, the above construction does not work so we might have to work a little harder, as we will see in the next section.

2.5.2 ElGamal Signature

We now want to concretely see how one could build digital signatures from the ElGamal cryptosystem. Here we have that $\mathcal{P} = G$ and $\mathcal{C} = G \times G$. For simplicity, we will first work in the original setting of ElGamal, where $G = \mathbb{F}_p^*$ for a prime p and let g be a generator of G. It will then be an easy task to generalize for an arbitrary group.

The first thing to try is to naively use the private key to produce the signature:

$$\begin{array}{lcl} \textbf{\textit{Alice}} & & \textbf{\textit{Bob}} \\ \text{Pick a random } k \in G & \xrightarrow{m,\ k} & \\ \text{Compute } j := m \cdot (g^k)^a & \xrightarrow{j} & \text{Check if } j = m \cdot (g^a)^k \end{array}$$

This obviously doesn't work since Alice could have computed j as $m \cdot (g^a)^k$ without knowing a. Hence, we must *force* Alice to really use a when producing her signature. Bob could then secretly pick k, transmit g^k to Alice and challenge her to compute j.

$$\begin{array}{lcl} \textbf{\textit{Alice}} & & \textbf{\textit{Bob}} \\ & \xleftarrow{g^k} & \text{Pick a random } k \in \mathbb{Z},\ 0 < k < \operatorname{ord}(g) \\ \text{Compute } j := m \cdot (g^k)^a & \xrightarrow{m,\ j} & \text{Check if } j = m \cdot (g^a)^k \end{array}$$

For sure, Bob will be convinced that the message came from Alice. However, Bob could have produced this signature by himself simply by computing j as $m \cdot (g^a)^k$, so this approach is not good either. Instead of choosing k arbitrarily, we might be able to force Alice to compute k using her private key. For example, to solve $m = g^{ak}$, assuming that we know that $m = g^l$, we need to solve the congruence $l \equiv ak (\operatorname{mod} \operatorname{ord}(g))$, which can be done as soon as a is known and invertible (i.e. $\gcd(a, \operatorname{ord}(g)) = 1$). However, if it isn't, then finding k requires to compute the discrete logarithm $\log_{g^a} m$. Of course, in practice, recovering l from m requires to compute $\log_g m$. So instead of computing l, we could choose it first and since we need it to depend on the message, the canonical choice is to consider an equation of the form $g^m = g^{ak}$:

$$\begin{array}{lcl} \textbf{\textit{Alice}} & & \textbf{\textit{Bob}} \\ \text{Solve } m \equiv ak (\operatorname{mod} \operatorname{ord}(g)) \text{ for } k & \xrightarrow{m,\ k} & \text{Check if } g^m = (g^a)^k \end{array}$$

But once Bob knows m and k, he could solve $m \equiv ak (\operatorname{mod} \operatorname{ord}(g))$ for a and hence learn Alice's private key. Thus, the signing equation $g^m = g^{ak}$ is too simple. So from Bob's point of view, we need more than one unknown since otherwise, a can uniquely be determined. We can then try to add an extra variable t and modify the signing equation to, say, $g^m = g^{ak} \cdot g^t$. As we just said, the value of t should be unknown to Bob, so Alice would transmit g^t to Bob instead:

Alice		***Bob***
Pick a random $t \in \mathbb{Z}$, $0 < t < \operatorname{ord}(g)$	$\xrightarrow{m,\ g^t}$	
Solve $m \equiv ak + t (\operatorname{mod} \operatorname{ord}(g))$ for k	$\xrightarrow{k}$	Check if $g^m = (g^a)^k \cdot g^t$

But Alice can cheat once more: she could first pick k and compute the value of g^t by performing $(g^a)^{-k} \cdot g^m$. This tells us that we should require that Alice gives us *'a proof'* that she knows the value of t without revealing its actual value (recall that if Bob learns t (and k), he can then compute a). Hence, we could disclose only a *part of* t, just like in the key exchange protocol. So we write $t = rs$ and by revealing only g^r and s, Bob could compute g^t without knowing the value of t:

Alice		***Bob***
Pick random $r, s \in \mathbb{Z}$, $0 < r, s < \operatorname{ord}(g)$	$\xrightarrow{m,\ g^r,\ s}$	
Solve $m \equiv ak + rs (\operatorname{mod} \operatorname{ord}(g))$ for k	$\xrightarrow{k}$	Check if $g^m = (g^a)^k \cdot (g^r)^s$

On the other hand, Alice can still pick k first, compute $(g^a)^{-k} \cdot g^m$ to deduce the value of g^t, but this time, she has to transmit s as well. Hence, she has to find g^r and s such that $(g^r)^s = g^t$. An easy way out is to set $s = 1$, or any other value for which s-roots in G are efficiently computable[20]. She then sets $g^r = g^{t/s}$ and was therefore able to forge a signature on the message of her choice. Since we cannot prevent Alice from first picking m, k and get the corresponding value of $(g^r)^s$, we really have to ensure that she won't be able to choose the s she wants and then get g^r. The weakness that was exploited here is that the left-hand side of $(g^a)^{-k} \cdot g^m = (g^r)^s$ is independent of g^r, so that s was allowed to be chosen first. The only parameter of the left-hand side on which we have some freedom is k. The easiest thing is then to set $k := g^r$ and since the value of g^r was already transmitted, it even decreases the amount of data sent to Bob. The (honest) Alice would then have to derive the corresponding value of s in the last step.

The signature on message m is then the pair (g^r, s). With this last improvement, first notice that recovering a from the signing equation requires to solve an instance of the DLP. Next, we examine if a signature could be forged. If r (or g^r) is chosen first, then computing s requires to solve a DLP. Conversely, if s is fixed first, the equation

[20] For instance, square roots are easy to compute in $\mathbb{F}_p^*$ (see [Per86] for example), so taking s to be any small power of 2 would do.

ElGamal Signature Scheme

Alice		***Bob***
Secretly pick a random $r \in \mathbb{Z}$, $0 < r < \operatorname{ord}(g)$	$\xrightarrow{m}$	
such that $\gcd(r, \operatorname{ord}(g)) = 1$	$\xrightarrow{g^r}$	
Solve $m \equiv ag^r + rs (\operatorname{mod} \operatorname{ord}(g))$ for s	$\xrightarrow{s}$	Check if $g^m = (g^a)^{g^r} \cdot (g^r)^s$

$$g^m = (g^a)^x \cdot x^s \tag{2.3}$$

must be solved in G for the unknown x. So far, nobody was able to provide an efficient method to solve this kind of equations, so ElGamal's original challenge [ElG85a, p.470] still holds: *"The reader is encouraged to find a polynomial-time algorithm for solving (2.3)"*. Another possible approach to forge a signature would be to devise a process that *simultaneously* determine g^r and s. But here again, no one was able to find a feasible way to perform this task. In fact, twenty years have now passed since this signature scheme was first proposed and yet, no attack was successful at breaking it.

Notice that the ElGamal signature still keeps the same secret and public key for all users, which is quite practical. But since $\mathcal{P} \neq \mathcal{C}$, the signature-verification procedure is now really different from decryption-encryption. We saw why it didn't seem possible to stick really close to the PKC and we have complexified the verification equation step by step until we were no longer able to break it. However, using this trial and error procedure, we explored only one path, which means that there might be different verification equations that are secure and efficient as well. There are several others in fact. In the handbook [MvOV96, Notes 11.70–11.71], five alternatives are presented. For example, the equation $g^s = (g^a)^{g^r} \cdot (g^r)^m$ with the corresponding signature (g^r, s) has the advantage that the computation of s do not require to perform any inversion in G. So there is some freedom on the specific verification equation used, but the underlying idea really is the same. Of course, one must be extra cautious when playing with this equation since even a tiny modification could change the computational assumptions, and hence alter the security of the system.

One well-known variant of ElGamal is the Schnorr signature scheme [Sch91], which has the advantage of providing shorter signatures while seemingly maintaining the same level of security. And the most famous variant of ElGamal's signature is certainly the Digital Signature Algorithm (DSA) [NIoST00], which was the very first digital signature scheme to be approved by a government.

In practice, of course, the message can be quite long. So instead of producing the signature for m, what we sign is actually the message digest $h(m)$, where $h : \{0,1\}^* \to \mathbb{Z}/p\mathbb{Z}$ is a public cryptographic hash function. This has many advantages. Namely, it allows to have a fixed length for the signature, instead of having a signature twice as long as the message in the case of ElGamal. Also, in the original scheme [ElG85a, Section IV, Attack 6], given a message m and its corresponding signature (g^r, s), it is possible to produce another message u together with a valid signature (v, w). Luckily, this attack does not allow to choose u. And since h is a one-way function, it will then be impracticable to determine a message whose digest equals u. So with the help of the hash function, this particular attack cannot succeed. Thus, the use of cryptographic hash functions is not limited to the efficiency aspect, but has a role to play in the security of the scheme as well. It should therefore always be used in practice.

2.5.3 Generalized ElGamal Signature

So far, we have described the original ElGamal signature where the underlying group is $\mathbb{F}_p^*$ for a prime p. We now wish to extend this scheme to an arbitrary group G where the discrete log problem is believed to be intractable. Hence, we now consider g and m as elements of G. As discussed above, we want to sign the hash of the message, so technically, we need a hash function from G to $\mathbb{Z}/n\mathbb{Z}$, where n is the order of g. In practice, it will be easier to proceed in two steps: first provide a public message embedding f from G to $\{0,1\}^*$ and then use a well-studied hash function h as follows:

$$G \xrightarrow{f} \{0,1\}^* \xrightarrow{h} \mathbb{Z}/n\mathbb{Z}$$

If we assume that h is strongly collision resistant, then the composition $h_G := h \circ f$ will enjoy this property as well. For if we find a collision on h_G, say x and x' in G such that $h_G(x) = h_G(x')$ and $x \neq x'$, then $f(x) \neq f(x')$ (since f is one-to-one) and hence we would have found a collision on h as $h(f(x)) = h(f(x'))$.

Now, if we look at the original scheme with hash function and we try to use it *'as is'* in the group G, the only part that might not make sense is to encounter g^r as an exponent, since this is in fact an element of G instead of being an integer. But as pointed out previously, the important point is that the exponent of g^a should *depend on* g^r in order to avoid that the value of s be chosen first, which would allow a forged signature. In the case of the original algorithm, the canonical choice was to take g^r itself whereas here, the natural choice is to consider $h_G(g^r)$. Notice that this value can be computed directly from g^r by Bob, so that the data sent by Alice is unchanged. We hence obtain the following generalized scheme:

Generalized ElGamal Signature Scheme

Alice		***Bob***
Secretly pick a random $r \in \mathbb{Z}$, $0 < r < \mathrm{ord}(g)$	$\xrightarrow{m}$	
such that $\gcd(r, \mathrm{ord}(g)) = 1$	$\xrightarrow{g^r}$	
Solve $h(m) \equiv ah(g^r) + rs (\bmod \mathrm{ord}(g))$ for s	$\xrightarrow{s}$	Check if $g^{h(m)} = (g^a)^{h(g^r)} \cdot (g^r)^s$

As with the original ElGamal signature, nobody has been able to mount a successful attack on this generalized version, assuming that the DLP in G is computationally infeasible.

2.6 Groups suitable for DL-based Cryptography

We have now seen, in quite some details, three fundamental cryptographic primitives based on discrete logarithms in a group G: the Diffie-Hellman key exchange, the ElGamal public-key cryptosystem and the ElGamal signature scheme. In Section 2.8, we will give an overview of some of the numerous other applications of discrete logarithms in cryptography. In order to get the most out of these protocols, solid candidates for the group G are needed. Evidently, a good prospect has to ally *efficiency* and *security*. That is, we need the elements to be easily handled by a computer, the group operation in G to be relatively inexpensive to compute and of course, the DLP in G to be presumably intractable. In addition, our life will be made a lot easier if there is also an efficient algorithm to compute the cardinality of G.

Different applications, different needs: depending on the computing resources available, the short, medium or long term security needed or the nature of the information at stake, the choice of the group will inevitably vary. It is indeed the context that will determine what *balance* between efficiency and security is required. For instance, from smart cards to PCs to supercomputers, totally different criteria have to be filled. So in a nutshell: the longer the list of known suitable groups is, the better.

We now want to give a brief overview of the principal members on this list. Initially, Diffie and Hellman [DH76b] worked in the multiplicative group $\mathbb{F}_p^*$ of a finite field with a prime number p of elements. It was then natural to consider finite fields $\mathbb{F}_{2^n}^*$ of characteristic 2 as well and to generalize to any Galois field $\mathbb{F}_{p^n}^*$, where p is prime and n is a positive integer.

In 1985, Neal Koblitz [Kob87] and Victor Miller [Mil86b] independently proposed to use the group of points on an elliptic curve over a finite field. A remarkable fact concerning these groups is that we can efficiently generate elliptic curves for which the only known algorithms to compute their discrete logarithms are *exponential-time*. As a result, the key length can be much shorter than in a system where subexponential-time algorithms are known. Just to give an idea, the effort required to factor a 1024-bit RSA modulus or to extract a discrete logarithm in $\mathbb{F}_q^*$, where q is a 1024-bit prime, is *roughly* the same as to solve a DLP in a (suitably chosen) elliptic curve over $\mathbb{F}_p$, where p is a 160-bit prime only. Hence, elliptic curves are a good example where the efficiency/security ratio pays off: the group operation may be more expensive, but since shorter keys are needed, the overall cost makes it a competitive choice.

Elliptic curves are a sub-family of the hyperelliptic curves. In 1988, Neal Koblitz generalized his idea to create the hyperelliptic cryptosystems [Kob89]. To be more accurate, the underlying group where the discrete logarithm is presumably hard is the Jacobian of a hyperelliptic curve over a finite field. The Ph.D. thesis of Tanja Lange [Lan01] was devoted to efficiently perform the arithmetic in these groups. For security reasons, it is recommended in practice to use hyperelliptic curves of *low genus g*. Up to this date, taking g to be 1, 2 or 3 is advised (the case $g = 1$ corresponding to elliptic curves). See Section 2.7.2 for more details.

Also in 1988, Kevin McCurly suggested to use $\mathbb{Z}_n^*$, the group of invertible elements of $\mathbb{Z}_n$ where n is composite, in a modified version of the ElGamal cryptosystem [McC88]. It has been shown that breaking his scheme is at least as difficult as factoring n. Moreover, Håstad, Schrift and Shamir showed that when n is a Blum integer[21], then all bits of the discrete logarithm are individually hard and moreover, that the lower half of the bits, just like the upper half, are simultaneously hard[22] [HSS93]. So, as Schrift and Shamir puts it, *'The discrete log is very discreet'* [SS90].

One more proposal was done that year: the ideal class group of an imaginary quadratic field $\mathbb{Q}(\sqrt{D})$. That is, an element of this group is an *equivalence class* of ideals of the number ring of $\mathbb{Q}(\sqrt{D})$ (where D is a squarefree negative integer). The idea of using this structure in cryptography is due to Johannes Buchmann and Hugh Williams [BW88]. However, a subexponential-time algorithm to compute this DLP was devised by Kevin McCurley the following year [McC89]. At CRYPTO '89, Buchmann and Williams then proposed to use *real* quadratic fields instead [BW90, BSW94]: this was the first time a key-exchange was based on a structure which was *not* a group (see p.19).

At CRYPTO 2003, Karl Rubin and Alice Silverberg introduced the concept of torus-based cryptography. They described the cryptosystem CEILIDH[23] for which the underlying group is an algebraic torus over a finite field. A Scots Gaelic word, *ceilidh* is a traditional Scottish

[21] A *Blum integer* is a product $n = pq$ of two distinct prime numbers p and q satisfying $p \equiv q \equiv 3 (\mathrm{mod}\, 4)$.
[22] Under the assumption that factoring large Blum integers is an intractable problem.
[23] Pronounced *'kayley'*.

gathering and was chosen because of the acronym '**C**ompact, **E**fficient, **I**mproves on **L**UC[24] and **I**mproves on **D**iffie-**H**ellman'[25]. It has the advantage that the group elements can be represented in a really compact form and so it decreases the amount of information exchanged between Alice and Bob. Further details on this topic will be given in Section 4.6.

That concludes our brief survey of groups suitable for DL-based cryptography. The goal of this thesis is now to add one more entry to this list, namely, the *generalized Jacobians* of an algebraic curve defined over a finite field.

2.7 Solving the Discrete Logarithm Problem

Designing protocols whose security depends on the intractability of the discrete logarithm problem or searching for groups where this problem seems intractable is useless if we do not take the time to develop and refine methods to solve it. Using the state-of-the-art in these techniques will allow us to select the size of the group needed to meet the desired security parameter.

In this section, we wish to present a snapshot of some of the methods in use today. For each of them, we list its main characteristics as well as the principle behind it. By definition, such a description is neither complete nor rigorous. However, details can be found in the surveys of McCurley [McC90], Odlyzko[Odl00] and Teske [Tes01].

There are two types of algorithms that can be distinguished: the generic and the specific ones. The *generic* methods will work in nearly any cyclic group (see below) whereas *specific* methods are *'custom-made'* since they take full advantage of the representation of the group. It is therefore not surprising that specific algorithms generally perform better in practice than generic ones.

As usual, G is here a finite cyclic group of order n generated by g. So given $h \in G$, we wish to determine the smallest non-negative integer k such that $g^k = h$.

2.7.1 The Baby, the Giant and the Kangaroos

In this section, we plan to tell the tale of generic algorithms using the colorful images that have now become classics of the literature.. For generic algorithms, we really want to assume the minimum about G. That is, only the following facts can be used:

1. Each element of G is encoded as a unique binary string
2. We have access to a *black box oracle* for the group law and the inverse of elements

[24] LUC is a public key cryptosystem based on Lucas functions and which was described in [LS93].

[25] It was also named in the memory of Alice Silverberg's cat Ceilidh, to which the paper is dedicated.

These properties imply that the identity can be identified and that we are able to decide if two elements are equal or not (that is, we can perform *'equality checks'*).

At EUROCRYPT '97, Victor Shoup showed that *any* generic algorithm solving[26] the DLP in G *must* perform $\Omega(\sqrt{p})$ group operations, where p is the largest prime dividing n [Sho97b]. As a result, the performance of the generic algorithms presented below should really be seen in the light of Shoup's lower bound.

The principal generic algorithms are the rho and kangaroo methods, both due to Pollard, as well as Shanks' baby-step giant-step algorithm. And if the factorization of n is known, then one can also use the Pohlig-Hellman algorithm.

BABY-STEP GIANT-STEP. The *baby-step giant-step (BSGS)* method is due to Daniel Shanks [Sha71]. It was originally designed to compute the ideal class number of a quadratic number field. Proposed in 1971, it was hence known prior to the Diffie-Hellman key-exchange protocol. The BSGS is a deterministic generic algorithm which is in fact a time-memory trade-off of an exhaustive search. The idea behind this method is that if we set $m = \lceil \sqrt{n} \rceil$, then $h = g^k = g^{im+j}$ for some i, j such that $0 \leq i, j < m$. Thus, $h(g^{-m})^i = g^j$ and so it suffices to compute two sorted lists, one with all g^j (the *baby-steps*) and one with all $h(g^{-m})^i$ (the *giant-steps*). To get $\log_g h$, we simply find a match between the two lists. Note that the BSGS has a large memory requirement (needing the storage of $O(\sqrt{n})$ group elements) and has running time $O(\sqrt{n})$ group operations.

POHLIG-HELLMAN. As mentioned earlier, the Pohlig-Hellman[27] generic algorithm was part of the same paper as their secret-key cryptosystem [PH78]. This method requires that the factorisation of n be known. So let $n = p_1^{\alpha_1} p_2^{\alpha_2} ... p_N^{\alpha_N}$, where the p_i's are distinct primes and each $\alpha_i > 0$. Since this process requires to perform $O\left(\sum_{i=1}^{N} \alpha_i \left(\log_2 n + \sqrt{p_i} \log_2 p_i\right)\right)$ group operations to extract a logarithm (once the factorisation of n is known)[28], Pohlig-Hellman will be rather efficient if n has only small prime factors. In practice, it is thus advised to choose a group order having at least one large prime dividing it. Here is how it works: first compute $k_i := k \bmod p_i^{\alpha_i}$ for each i and then use the Chinese remainder theorem to recover k. Now, to compute each k_i, write it in base p_i as $k_i = l_0 + l_1 p_i + ... + l_{\alpha_i - 1} p_i^{\alpha_i - 1}$. Start by determining l_0 from the identity $h^{n/p_i} = (g^{n/p_i})^{l_0}$, then find l_1 using $h^{n/p_i^2} = g^{nl_0/p_i^2} \cdot (g^{n/p_i})^{l_1}$ and so on until $l_{\alpha_i - 1}$ is known.

[26] with probability bounded away from zero

[27] This algorithm was also independently discovered by R. Silver and by R. Schroeppel and H. Block, but S. Pohlig and M. Hellman were the first to publish it.

[28] See [PH78, Section IV] for a precise account of what can be simultaneously achieved in terms of running time, memory and precomputations.

POLLARD'S RHO. The ρ-method was developed by Pollard [Pol78] in 1978. It is a probabilistic algorithm based on the birthday paradox which has expected running time $O(\sqrt{n})$ group operations. It is preferred to BSGS in practice since it requires a negligible amount of storage. It can in fact be implemented in such a way that only a constant number of group elements have to be stored. The strategy here is to *recursively* define a sequence $\{x_i\}_{i\geq 0}$ of elements of G of the form $x_i = g^{a_i}h^{b_i}$ such that $x_0 \in G$ with known a_0, b_0 (e.g. $x_0 = 1$) and x_{i+1} is a function of x_i only. Since G is finite, then this sequence will eventually be *periodic* (so that a schematic representation of this sequence looks like the letter ρ). Then find[29] any two elements x_i and x_j of this sequence that are equal and such that $b_i \not\equiv b_j (\bmod n)$. Finally, k can be easily determined from $g^{a_i}(g^k)^{b_i} = g^{a_j}(g^k)^{b_j}$. In 1999, van Oorschot and Wiener [vOW99] developed a parallelized version where each of the N processors utilizes the same recurrence relation, but with a different starting point. The search for a match is carried out through *all* computed values of the processors at once (and not merely within each sequence), yielding a linear[30] speed-up.

KANGAROO METHOD. The λ method, also referred to as the *'Kangaroo method'*, is also due to Pollard and was published in the same article as the ρ method [Pol78]. It is a space efficient randomized algorithm as well, but is especially suited when we already know an interval $[a, b]$ in which the discrete logarithm k lies. In fact, it is expected to require $O(\sqrt{b-a})$ group operations and storage of $O(\log_2(b-a))$ group elements to extract a discrete log. The goal of this *'game'* is now to make the paths of the tame and the wild kangaroos collide. First, the tame kangaroo starts at position g^b and performs a set of jumps of the form g^{d_i} and then stops at position $g^{b+d_1+\ldots+d_N}$, where the travelled distances d_i are known. Then, the wild kangaroo starts at position h (or hg^δ where δ is chosen to be small) and also executes a number of jumps of known distances until the wild kangaroo meets the tame one[31], i.e. $g^{b+d_1+\ldots+d_N} = hg^{\delta+d'_1+\ldots+d'_M}$. If it doesn't happen, we simply try again starting the wild kangaroo at a different initial position. Since we kept track of the travelling distances, it is then easy to compute k. The kangaroo method has also been parallelized with a linear speed-up by van Oorschot and Wiener [vOW99] and further analysis and improvements were done by Pollard [Pol00] himself and Edlyn Teske [Tes01].

[29] Using for example Brent's algorithm [Bre80].

[30] That is, a speed-up by a factor of N.

[31] Each jump is completely determined from the current position. So if the wild kangaroo steps on a spot where the tame kangaroo once was, then from that point on, their two paths will coincide (and look like a λ).

2.7.2 Specific algorithms

"Les structures sont les armes du mathématicien" once said Bourbaki[32]. Well, apparently, they are the weapons of the cryptanalyst too. In order to develop targeted methods to solve the DLP in G, one has advantage to thoroughly exploit its structure.

INDEX-CALCULUS ALGORITHM. The idea behind the index-calculus method seems to date back to the 1920s. Kevin McCurley [McC90] indeed attributes it to Kraitchik and Cunningham. In the context of public-key cryptography, Adleman [Adl79] first described and analyzed the algorithm for $\mathbb{F}_p^*$ while Hellman and Reyneri [HR83] worked in $\mathbb{F}_{p^m}$. And with its numerous improvements over the years, the index-calculus has become one of the most powerful techniques known to solve DLPs. Index-calculus works as follows. First choose a relatively small subset $S = \{s_1, s_2, ..., s_N\} \subseteq G$ that can serve as a *'factor base'* (that is, we want to be able to write a significant proportion of the elements in G as $s_1^{\alpha_1} \cdot s_2^{\alpha_2} \cdot ... \cdot s_N^{\alpha_N}$). We then want to build a database containing the discrete logarithms $l_i := \log_g s_i$ $(1 \leq i \leq N)$. To do so, we first need to build a system of linear equations with unknowns l_1, l_2, ..., l_N. The equations are collected as follows. Pick a random exponent r. If we can find α_1, α_2, ..., α_N satisfying $g^r = s_1^{\alpha_1} \cdot s_2^{\alpha_2} \cdot ... \cdot s_N^{\alpha_N}$, then $r \equiv \alpha_1 l_1 + \alpha_2 l_2 + ... + \alpha_N l_N (\bmod n)$ is added to the list of equations. We repeat this process until this system has a unique solution (so at least N equations are needed). Solving this system will yield the values of l_1 up to l_N. Now, to compute $\log_g h$, we pick random exponents t until we can find β_1, β_2, ..., β_N such that $hg^t = s_1^{\beta_1} \cdot s_2^{\beta_2} \cdot ... \cdot s_N^{\beta_N}$. Finally, $\log_g h + t \equiv \beta_1 l_1 + \beta_2 l_2 + ... + \beta_N l_N (\bmod n)$. In practice, the index-calculus in $\mathbb{F}_p^*$ and $\mathbb{F}_{2^m}^*$ have expected running time[33] $L[p, 1/2]$ and $L[2^m, 1/3]$ respectively (using Coppersmith's improvement [Cop84] for characteristic 2). Fortunately, the power of the index-calculus method does not seem to apply to large enough subgroups of $\mathbb{F}_p^*$ of prime order[34] or to suitably chosen elliptic curves[35]. However, index-calculus can be applied to hyperelliptic curves and with the latest developments [Gau00, The03], it already performs better than the generic ρ-method for genus greater than 2.

We conclude this section by insisting on the fact that the above description is just the tip of the iceberg. Indeed, many other specific algorithms and refinements are known, such as the Gaussian integer method [COS86], the number field sieve [Gor93] or the function field sieve [Adl94]. In addition, several other tools, including the structured Gaussian elimination [Odl85],

[32] That is, *"Structures are the weapons of the mathematician"*. Created in the 1930s, *'Nicolas Bourbaki'* is in fact a pseudonym used by a group of (mainly French) mathematicians.

[33] Recall that $L[n, \alpha] := O\left(e^{(c+o(1))(\ln n)^{\alpha}(\ln \ln n)^{1-\alpha}}\right)$, where c is a positive constant.

[34] Used in the Digital Signature Algorithm (DSA).

[35] See [Mil86c] for a discussion of why the index-calculus method cannot be readily applied to elliptic curves.

the Weil [MOV93] and Tate [FMR99] pairings as well as the Weil descent [GHS02], are either at the heart of an attack or are employed to improve existing ones. Moreover, the difficulty of computing individual bits or groups of bits of a discrete logarithm is also an important issue which is addressed in [MvOV96, Section 3.9]. Lastly, Peter Shor designed polynomial-time Las Vegas algorithms for both discrete logarithms in $\mathbb{F}_p^*$ and integer factorization on a (hypothetical) quantum computer [Sho94, Sho97a, BL95]. At CRYPTO '95, Dan Boneh and Richard Lipton [BL95] used a similar method to show that the discrete logarithm problem in any finite group (where the group operation can be computed efficiently) can be solved in random quantum polynomial-time. It should therefore be kept in mind that we are everything but immune against the practical realization of polynomial-time attacks towards the discrete logarithm problem.

2.8 Versatility of Discrete Logarithms

We conclude this chapter with a selection of different applications of discrete logarithms in cryptography. Since our science aims at *securing information*, there is so much more to it than key-exchange, encryption and signatures. To reflect this reality, we chose from a wide range of applications three independent occurrences that will hopefully demonstrate what an ubiquitous and polyvalent tool discrete logarithms are for cryptographers.

2.8.1 Coin-Flipping, Bit Commitments... and Computer Games

Before betting even a single penny at the roulette of an online casino, Bob should be convinced that the winning number can't be changed after he placed his bet. But how to make sure that *they* are playing fairly? In 1981, Manuel Blum and Silvio Micali described an algorithm that could answer this question and many more. Blum [Blu82] humoristically called his own work *'a protocol for solving impossible problems'*. To describe this technique, the *coin-flipping by telephone*, nothing surpasses his own words:

> *"They (Alice and Bob) have just divorced, live in different cities, want to decide who gets the car. Bob would not like to tell Alice HEADS and hear Alice (at the other end of the line) say "Here goes... I'm flipping the coin... You lost!" "*

A fair coin-flipping can be achieved using what is called a *bit commitment scheme*. The action of *'committing to a bit'* can be described as follows: Alice first picks a bit b, either 0 or 1. She places it in a safe (whose combination is only known to her), closes it and gives it to

Bob. Once the safe is in Bob's hands, Alice cannot change her mind: she is therefore *bound* to b. Moreover, the value of b is *concealed* from Bob until Alice opens the safe for him[36].

Now, if Alice and Bob wants to virtually flip a coin, Alice begins by committing to a bit b. Bob then tries to guess what b is and publicly announces his guess b' to Alice. She then unveils b by opening her safe. Bob wins if $b = b'$, and looses otherwise. Moreover, if Alice ever refuses to open the safe, Bob could then conclude that his guess was right.

Once more, groups where the discrete logarithm problem is believed to be intractable can serve as a tool to build tangible bit commitment and coin-flipping schemes. Not surprisingly, the coin-flipping protocol originally proposed by Blum and Micali was indeed relying on the intractability of the discrete logarithm problem. In [BCC88, Sections 6.1.2 and 6.2.2], Brassard, Chaum and Crépeau describe two realizations of a bit commitment based on the discrete logarithm, one unconditionally secure for Alice and another which is unconditionally secure for Bob.

2.8.2 Secret Sharing... and National Security

When taking decisions concerning national security, an agreement among several executives is required. The well-known (and extreme) instance being the launch of a nuclear missile. In Russia, at least two of the President, the Defense Minister and the Defense Ministry have to give their consent before any action can be taken[37]. This *'two-man rule'* in fact applies in a variety of contexts, from opening the vault of a bank to shutting down a central server.

Hence, we need a way to ensure that only precise subsets of people are authorized to take a decision. To achieve this goal, a secret s could be shared among all participants in such a way that a coalition can recover s if and only if they form an authorized subset. In such a *secret sharing scheme*, each participant P_i receives a piece of information s_i (called a *share*) from a dealer.

Notice that this method could also be used by a single individual who wishes to safeguard a sensitive piece of information I: the data could be split into n parts (say such that a minimum of $n/2$ shares are required to recover I) and each piece placed at a different (secret) physical location.

A secret sharing is said to be *perfect* if pooling the shares of any unauthorized subset of participants yields absolutely no information about s. For a toy example[38], the following magic

[36] Notice the difference with a public-key cryptosystem (c.f. Section 2.4) where Bob was the one able to open the safe.

[37] See the Time Magazine of May 4,1992 on page 13.

[38] Please take note that magic squares are used here as an illustrative example only: they are easy to understand, but are not practical secret sharing schemes per se.

square

11	16	9
10	12	14
15	8	13

with positive integer entries as shares, and where $s = 36$ is the sum of any line, column or diagonal, is *not* a perfect secret sharing since, for instance, the person having share 16 *knows* that $s \geq 18$.

The concept of secret sharing was independently proposed by Adi Shamir [Sha79] and George R. Blakley [Bla79] in 1979. However, in its original formulation, the malicious Eve could provide a dummy share and hence prevent the reconstruction of s when desired. As well, a corrupted dealer could really do anything he likes depending on the bribes he received. In order to circumvent these difficulties, Chor, Goldwasser, Micali and Awerbuch [CGMA85] introduced the concept of a *verifiable secret sharing* (VSS) in 1985. In such a scheme, each participant can verify that the share they received is authentic and moreover, no one can successfully submit an invalid share when comes the time to recover s. However, it would be even better if *anyone* (and not only the participants) could verify that the shares have been distributed correctly. Markus Stadler[Sta96] introduced this notion at EUROCRYPT '96 and called such a scheme a *publicly verifiable secret sharing* (PVSS). He proposed a protocol using '*double exponentiations*', i.e. exponentiations of the form $g^{(k^n)}$ and consequently, *'double discrete logarithms'*.

2.8.3 Identification Schemes... and Your Banking Card

An *identification scheme* is a protocol that will allow Alice to prove her identity to Bob in such a way that while Bob is convinced that he is really talking to Alice, he won't in turn be able to usurp her identity. Therefore, solely providing a login and password to access email, typing a PIN to withdraw money (with a banking card with a magnetic stripe only[39]) or telling a credit card number by telephone is by no mean considered an identification scheme.

So instead of giving away all the secret information, an identification scheme usually takes the form of a *challenge-and-response* protocol. That is, Bob sends to Alice a (random) challenge which can only be answered correctly if Alice's secret information S_A is know. Alice computes her answer using S_A and sends only her answer to Bob, keeping S_A secret. Finally, he verifies if the answer is correct or not. Since a new challenge is issued each time, Bob (or an eavesdropper) will have a negligible probability to impersonate Alice.

Such a scheme can be realized with the help of a group where the DLP is believed to be intractable. For instance, Tatsuaki Okamoto [Oka93] presented at CRYPTO '92 a *provably*

[39] Yes, in North America, we are still using them.

secure modification of the Schnorr identification scheme [Sch91]. Indeed, an elegant yet subtle proof shows that the Okamoto identification scheme is as secure as the discrete logarithm problem in $\mathbb{F}_p^*$. In addition, the resulting scheme is still almost as efficient as the original version proposed by Schnorr.

Finally, even if we just saw concrete examples where the discrete logarithm problem was playing a central role, it might still not be enough to convince a sceptical friend that the DLP is present in our everyday lives. Well, let's just say that SSH (Secure Shell), SSL (Secure Socket Layer), PGP (Phil Zimmermann's Pretty-Good-Privacy) or OpenPGP all rely on the discrete logarithm problem at some level. So unless your friend still believes that computers are not part of our lives yet, that should be a massive argument.

Chapter 3

Algebraic Curves

"Think geometrically, prove algebraically."

- Silverman & Tate

Not surprisingly, abstract algebra and geometry are the two underlying branches of algebraic geometry. Loosely speaking, algebraic-geometers study, among other things, the sets of solutions of systems of algebraic equations. So algebraic geometry offers us the neccessary geometric tools to fuel our intuition, but uses the power of algebra to provide demonstrations.

In this chapter, we intend to study the necessary background on algebraic curves needed to understand generalized Jacobians. The first section on the Zariski topology will provide the basics of algebraic geometry required to define projective varieties and algebraic curves. The second section on plane curves and cryptography already gets more specific and considers three families of curves: Pell conics, elliptic and hyperelliptic curves.

There is a particular goal we wished to achieve by choosing to present each of these families. First, Pell equation makes the perfect introductory example of an algebraic curve suitable for DL-based cryptography. Indeed, this well-known equation is simple enough that within a few pages, it is possible to explain its cryptographic potential in detail. At the same time, we make the parallel with algebraic tori, so that the reader has at least one concrete example at hand.

Next come elliptic curves: we provide the fundamental properties that make them so attractive to cryptographers. Since Chapter 5 introduces a new cryptosystem based on the generalized Jacobian of an elliptic curve, it is the case we treat with the most details.

We then briefly touch upon hyperelliptic curves and introduce them as a motivation for the presentation of the theory of divisors that leads to the Picard group and the Jacobian.

Our treatment of divisors will of course emphasize the role played by principal divisors. We then present the Riemann-Roch theorem, whose power we demonstrate in the proof on the Abel-Jacobi theorem. This last result will be playing a key role in the generalized Jacobians we consider in Chapter 5.

Lastly, we recall the construction of the Picard group and provide a motivation for its use in cryptography. This will naturally lead us to the existence theorem for the Jacobian.

As a result, we believe that the content of this chapter, with the material it covers and the level of details it provides, can play the role of a self-contained introduction to the algebraic geometry underlying curve-based cryptography, as well as being a relatively brief reference for those already familiar with this material.

3.1 The Zariski Topology

Considered by many as one of the most influential mathematicians of his field in the twentieth century, Oscar Zariski studied in Italy[1] with Francesco Severi, who was the first to explicitly mention generalized Jacobians in the mid-1950s . Among the students of Zariski was Maxwell Rosenlicht, whose role in the study of generalized Jacobians is prominent, as we will see in Chapter 4.

Throughout this chapter, K will denote a perfect field. That is, every algebraic extension of K is separable. For the cryptographic applications we have in mind, notice that K will ultimately be a finite field and hence this framework is general enough for curve-based cryptography. Let also $\overline{K}$ be a fixed algebraic closure of K.

We all learned cartesian product in elementary school: we now see how the same underlying idea is used in the case of affine spaces.

Definition 3.1 *The* affine n-space over K*, denoted* $\mathbb{A}^n\left(\overline{K}\right)$ *(or simply by* $\mathbb{A}^n$ *when* K *is understood) is the set of all* n*-tuples of elements of* $\overline{K}$*:*

$$\mathbb{A}^n\left(\overline{K}\right) = \left\{ (x_1, \ldots, x_n) \middle| \, x_i \in \overline{K} \textit{ for } 1 \leq i \leq n \right\}.$$

Similarly, let

$$\mathbb{A}^n\left(K\right) = \left\{ (x_1, \ldots, x_n) \middle| \, x_i \in K \textit{ for } 1 \leq i \leq n \right\}.$$

The elements of $\mathbb{A}^n(K)$ *are called the* K-rational points *of* $\mathbb{A}^n$*. Also let* $\mathbf{0} = (0, \ldots, 0) \in \mathbb{A}^n(K)$*.*

[1] He thus embraced the Italian school of algebraic geometry, whose style was renowned to be very intuitive.

When seeing the chord-and-tangent rule on an elliptic curve for the first time, one without the appropriate background could think that *'adding a point at infinity'* seems like an artificial procedure while it is, in fact, a very natural construction. In order to see why, we need to first consider an equivalence relation on the nonzero points of $\mathbb{A}^{n+1}$. Given points $P = (x_0, \ldots, x_n)$ and Q in $\mathbb{A}^{n+1} \backslash \{\mathbf{0}\}$, we will write $P \sim Q$ if there exist a constant $\lambda \in \overline{K}^*$ such that

$$Q = (\lambda x_0, \ldots, \lambda x_n).$$

Clearly, this defines an equivalence relation on the points of $\mathbb{A}^{n+1} \backslash \{\mathbf{0}\}$. The equivalence class of the point P is denoted by $[x_0 : \ldots : x_n]$.

Definition 3.2 *The* projective n-space over K, *denoted* $\mathbb{P}^n\left(\overline{K}\right)$ *(or simply by* $\mathbb{P}^n$ *when* K *is understood) is the set of these equivalence classes. In other words,*

$$\mathbb{P}^n\left(\overline{K}\right) = \left\{ [x_0 : \ldots : x_n] \middle|\, x_i \in \overline{K} \textit{ for } 0 \leq i \leq n \textit{ and are not all zero} \right\}.$$

Similarly,

$$\mathbb{P}^n(K) = \left\{ [x_0 : \ldots : x_n] \middle|\, x_i \in K \textit{ for } 0 \leq i \leq n \textit{ and are not all zero} \right\},$$

and the elements of $\mathbb{P}^n(K)$ *are called the* K-rational points *of* $\mathbb{P}^n$*.*

We can now easily see why the *'points at infinity'* arise in a natural fashion for projective spaces. Consider for instance

$$\mathbb{P}^1 = \left\{ [x_0 : x_1] \middle|\, x_0, x_i \in \overline{K} \right\},$$

and let $P = [x_0 : x_1] \in \mathbb{P}^1$ be given. If $x_1 \neq 0$, then

$$[x_0 : x_1] = \left[\frac{x_0}{x_1} : 1\right].$$

Now if $x_1 = 0$, then $x_0 \neq 0$, from which follows that

$$[x_0 : x_1] = [1 : 0].$$

Thus, $\mathbb{P}^1$ is the union of two different types of points:

$$\mathbb{P}^1 = \left\{ [\lambda : 1] \middle|\, \lambda \in \overline{K} \right\} \cup \{[1 : 0]\}.$$

Since the set $\left\{ [\lambda : 1] \middle|\, \lambda \in \overline{K} \right\}$ is in bijection with $\mathbb{A}^1$, we can think of $\mathbb{P}^1$ as being '$\mathbb{A}^1$ *together with the extra point* $[1 : 0]$'. For this reason, $[1 : 0]$ is called a *point at infinity.*

Example 3.3 *This material makes the ideal introduction to see how easy it is to work with these concepts using* MAGMA. *Basic and fundamental instructions are shown below.*

```
> K:=GF(11);                    // Finite field (Galois field) with 11 elements
> K;
Finite field of size 11
> A2<x,y> := AffineSpace(K,2);
> A2;
Affine Space of dimension 2
Variables : x, y
> p := A2![1,2];
> p[1];                                        // 1st coordinate of the point p
1
> q:=A2![2,4];
> p eq q;
false
> P1<X,Y>:=ProjectiveSpace(K,1);
> Points(P1);                                   // Notice the point at infinity
{@ (0 : 1), (1 : 1), (2 : 1), (3 : 1), (4 : 1), (5 : 1),
   (6 : 1), (7 : 1), (8 : 1), (9 : 1), (10 : 1), (1 : 0) @}
> P2<X,Y,Z>:=ProjectiveSpace(K,2);
> P2;
Projective Space of dimension 2
Variables : X, Y, Z
> P:=P2![1,9,4];
> P;
(3 : 5 : 1)
> Q:=P2![6,10,2];
> P eq Q;
true
> P2![0,0,0];                      // At least one of X,Y or Z must be nonzero!

>> P2![0,0,0];                     // At least one of X,Y or Z must be nonzero!
     ^

Runtime error in '!': Illegal coercion
> quit;                                                         // To exit Magma
```

Take note that this example will be continued as we add more notions.

Our first goal is to turn $\mathbb{P}^n$ into a topological space. That is, we need to identify what the open sets of $\mathbb{P}^n$ are. But first, a few recalls.

Definition 3.4 A topology *on a set X is a collection $\mathcal{T}$ of subsets of X satisfying the following three properties:*

1. *$\phi, X \in \mathcal{T}$,*
2. *For any subcollection $\mathcal{S}$ of $\mathcal{T}$, $\cup_{U \in \mathcal{S}} U \in \mathcal{T}$,*
3. *For any U_1, ..., $U_n \in \mathcal{T}$, $U_1 \cap ... \cap U_n \in \mathcal{T}$.*

A set X endowed with a topology $\mathcal{T}$ is called a topological space*. The elements of $\mathcal{T}$ are called the* open sets*. A set is said to be* closed *if its complement is open.*

Before we can define the closed sets of our topology, we need a few more definitions.

Definition 3.5 *A polynomial $f(X_0, \ldots, X_n) \in \overline{K}[X_0, \ldots, X_n]$ is said to be* homogeneous of degree d *if*

$$f(\lambda X_0, ..., \lambda X_n) = \lambda^d \cdot f(X_0, ..., X_n)$$

for every constant $\lambda \in \overline{K}$.

Example 3.6 *The polynomial $f(X, Y, Z) = X^3 + Y + 1$ is not homogeneous, while $g(X, Y, Z) = X^3 + YZ^2 + Z^3$ is.*

Given $P = [x_0 : \ldots : x_n] \in \mathbb{P}^n$ and a homogeneous polynomial $f(X_0, \ldots, X_n) \in \overline{K}[X_0, \ldots, X_n]$ of degree d such that $f(P) = 0$, we have, for any $\lambda \in \overline{K}$,

$$f(\lambda x_0, \ldots, \lambda x_n) = \lambda^d \cdot f(x_0, \ldots, x_n) = \lambda^d \cdot f(P) = 0.$$

Thus, $f(P) = 0$ if and only if $f(Q) = 0$ for every point Q such that $P \sim Q$. It then makes sense to consider the zeros of f as elements of $\mathbb{P}^n$. And if we wish to consider more than one polynomial, then we can look for the points which are simultaneously zeros of all of them. This motivates the following definitions.

Definition 3.7 *An ideal of $\overline{K}[X_0, \ldots, X_n]$ is called an* homogeneous ideal *if it can be generated by homogeneous polynomials.*

Definition 3.8 *To each subset Y of $\mathbb{P}^n$, we associate the ideal $I(Y) \subseteq \overline{K}[X_0, \ldots, X_n]$ generated by the set*

$$\left\{ f \in \overline{K}[X_0, \ldots, X_n] \,|\, f \textit{ is homogeneous and } f(P) = 0 \textit{ for all } P \in Y \right\}.$$

$I(Y)$ is simply called the homogeneous ideal *of Y in $\overline{K}[X_0, \ldots, X_n]$.*

Conversely, we can also associate a subset of $\mathbb{P}^n$ to any set of homogeneous polynomials:

Definition 3.9 *Let $T \subseteq \overline{K}[X_0, \ldots, X_n]$ be a set of homogeneous polynomials. Then the set*

$$Z(T) = \{ P \in \mathbb{P}^n | \, f(P) = 0 \text{ for all } f \in T \} \, .$$

is called the zero set *of T.*

Intuitively, we are interested in solutions of one or more polynomial equations. Thus, we wish to look at the subsets of $\mathbb{P}^n$ which coincides with zero sets.

Definition 3.10 *A subset Y of $\mathbb{P}^n$ is called an* algebraic set *(or a* projective algebraic set*) if there is a set $T \subseteq \overline{K}[X_0, \ldots, X_n]$ of homogeneous polynomials such that $Y = Z(T)$.*

Definition 3.11 *A projective algebraic set is said to be* defined over K *if its homogeneous ideal can be generated by homogeneous polynomials in $K[X_0, \ldots, X_n]$.*

Definition 3.12 *Let V be a projective algebraic set defined over K. The set $V(K) = V \cap \mathbb{P}^n(K)$ is called the set of K-*rational points *of V.*

The algebraic sets turn out to have the properties of closed sets. For more information on the proof, we refer to [Har77, Proposition I.1.1].

Lemma 3.13 *The empty set and the whole space are algebraic sets. The union of a finite number of algebraic sets is an algebraic set. The intersection of any family of algebraic sets is an algebraic set.*

We therefore have a topology on $\mathbb{P}^n$:

Definition 3.14 *A subset of $\mathbb{P}^n$ is said to be* open *if its complement is an algebraic set. The topology determined by these open sets is called the* Zarisky topology *on $\mathbb{P}^n$.*

Remark that all the singletons of $\mathbb{P}^n$ are closed sets. Indeed, let $P = [x_0 : \ldots : x_n] \in \mathbb{P}^n$ be given, say with $x_k \neq 0$. Then it suffices to consider the following homogeneous polynomials:

$$\begin{aligned} f_0(X_0, \ldots, X_n) &= x_k X_0 - x_0 X_k \\ f_1(X_0, \ldots, X_n) &= x_k X_1 - x_1 X_k \\ &\vdots \\ f_n(X_0, \ldots, X_n) &= x_k X_n - x_n X_k. \end{aligned}$$

Let $T = \{f_0, f_1, \ldots, f_n\}$ and notice that $P \in Z(T)$. Conversely, let $P' = [x'_0 : \ldots : x'_n] \in Z(T)$ be given. Then $x'_k \neq 0$ since otherwise, $x'_0 = x'_1 = \ldots = x'_n = 0$ (which is forbidden in the projective space). It then follows that

$$x'_0 = \frac{x'_k}{x_k} \cdot x_0, x'_1 = \frac{x'_k}{x_k} \cdot x_1, \ldots, x'_n = \frac{x'_k}{x_k} \cdot x_n.$$

Thus, $P' = P$ (as equivalence classes) and we conclude that $Z(T) = \{P\}$. Notice that it also follows by Lemma 3.13 that any finite subset of $\mathbb{P}^n$ is closed as well.

Moreover, $\mathbb{P}^n$ equipped with the Zarisky topology fulfills the T_1 separation axiom[2]: given two distinct points P_1 and P_2, there are open sets U_1 and U_2 such that $P_1 \in U_1$ but $P_2 \notin U_1$, and $P_2 \in U_2$ but $P_1 \notin U_2$. Indeed, simply let $U_1 = \mathbb{P}^n \setminus \{P_2\}$ and $U_2 = \mathbb{P}^n \setminus \{P_1\}$.

We only need a few more recalls from topology before we can introduce projective varieties.

Definition 3.15 *Let X be a topological space and Y be a subset of X. Then a subset of X is said to be* closed in Y *if it is the intersection of Y with a closed set of X.*

Definition 3.16 *A nonempty subset Y of a topological space X is said to be* irreducible *if it cannot be written as a union $Y = Y_1 \cup Y_2$, where Y_1 and Y_2 are proper subsets closed in Y.*

Notice that this definition implies that the empty set is not considered to be irreducible.

Definition 3.17 *Let X be a topological space with topology $\mathcal{T}$. If Y is a subset of X, the collection*

$$\mathcal{T}_Y = \{Y \cap U | \, U \in \mathcal{T}\}$$

is a topology on Y, called the induced topology *(or the* subspace topology*).*

Definition 3.18 *An irreducible algebraic set of $\mathbb{P}^n$, with the induced topology, is called a* projective variety *or a* projective algebraic variety.

Definition 3.19 *The* dimension of a topological space X *is the supremum of all integers d such that there exists a chain*

$$Z_0 \subsetneqq Z_1 \subsetneqq \ldots \subsetneqq Z_d$$

of closed irreducible subsets of X.

Definition 3.20 *The* dimension *of a projective variety is its dimension as a topological space.*

[2] However, $\mathbb{P}^n$ with the Zariski topology is not Hausdorff (T_2). Recall that a topological space is *Hausdorff* if given any two distinct points P_1 and P_2, there are *disjoint* open sets U_1 and U_2 such that $P_1 \in U_1$ and $P_2 \in U_2$. See [Ful69, p.133].

And finally:

Definition 3.21 *An* algebraic curve *(or simply a* curve*) is a projective variety of dimension one.*

Thus, the only closed irreducible subsets of a curve must be points.

Remark 3.22 *For us, a curve will always be projective and irreducible.*

If the polynomial is reducible, say $f = gh$, then $f = 0$ as soon as $g = 0$ or $h = 0$. Thus, a necessary condition to have an irreducible set is that the polynomial itself be irreducible. More precisely, we have the following very useful result:

Proposition 3.23 *A projective variety of $\mathbb{P}^n$ has dimension $n - 1$ if and only if it is the zero set of a single irreducible homogeneous polynomial of positive degree.*

Example 3.24 *The variety $L \subseteq \mathbb{P}^2$ defined by*

$$L : aX + bY + cZ = 0$$

has dimension one if and only if at least one of a, b or c is nonzero.

Definition 3.25 *A* line *in $\mathbb{P}^2$ is an algebraic set given by a linear equation $aX + bY + cZ = 0$, with a, b, $c \in \overline{K}$ not all zero.*

We now continue the Magma example started earlier. This time, we learn how to define an algebraic curve.

Example 3.26 *In this example, we play with homogeneous polynomials and basic curves.*

```
> K:=GF(11);
> P2<X,Y,Z>:=ProjectiveSpace(K,2);
> Dimension(P2);
2
> P:=P2![1,9,4];
> f1:=X+2*Y+5*Z;
> Evaluate(f1,P);
7
> IsHomogeneous(P2,f1);
true
> L1 := Curve(P2,f1);
> L1;
```

```
Curve over GF(11) defined by
X + 2*Y + 5*Z
> IsProjective(L1);
true
> IsIrreducible(L1);
true
> Dimension(L1);
1
> R:=P2![0,3,1];
> R in L1;
true (0 : 3 : 1)
> Points(L1);                  // Notice that L1 contains 11+1 points in P2(K)
{@ (6 : 0 : 1), (4 : 1 : 1), (2 : 2 : 1), (0 : 3 : 1),
   (9 : 4 : 1), (7 : 5 : 1), (5 : 6 : 1), (3 : 7 : 1),
   (1 : 8 : 1), (10: 9 : 1), (8 : 10: 1), (9 : 1 : 0) @}
> f2:=6*X+Y+6*Z;
> L2 := Curve(P2,f2);
> V:=Intersection(L1,L2);
> Points(V);            // The parallel lines L1 and L2 intersect at infinity!
{@ (9 : 1 : 0) @}
```

Straight lines are the most basic examples of algebraic curves. Nonetheless, they play a central role in many nontrivial situations. Indeed, we will see in this chapter how the group law on Pell equation and on elliptic curves can be described geometrically in terms of lines. We will also see how useful they are when we work with divisors in Section 3.3. Finally, they will also be at the heart of the explicit group law for the generalized Jacobians we consider in Chapter 5. So let's just say that while progressing through this work, chances are that lines in $\mathbb{P}^2$ will become our new best friends.

Before we go any further, there is another family of curves that we wish to introduce: the lemniscates of Bernoulli. Surprisingly enough, they are at the origin of the study of elliptic curves. Indeed, from the treatment given by Michael Rosen in [Ros81], we see that the intrinsinc and fascinating connection with elliptic curves involves some of the greatest names in mathematics history.

A lemniscate is defined as the locus of points such that the product of the distances to two foci F_1 and F_2 is a constant c, the classical example being $F_1 = \big(-\sqrt{2}/2, 0\big)$, $F_2 = \big(\sqrt{2}/2, 0\big)$ and $c = 1/2$. This lemniscate is depicted over the reals in Figure 3.1.

The lemniscate has been introduced in 1680 by the French astronomer Giovanni Dominico

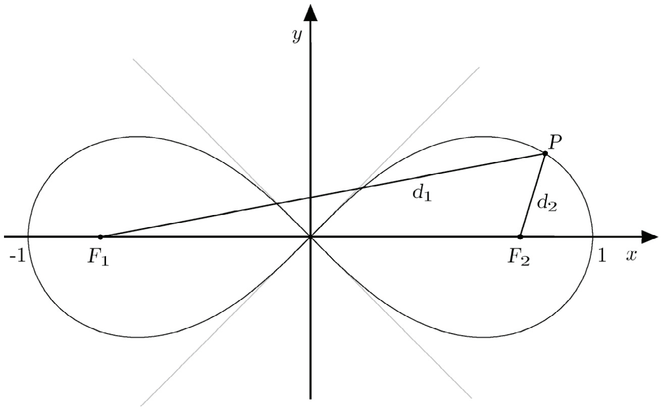


Figure 3.1: Lemniscate of Jakob Bernoulli over the real numbers

Cassini in order to illustrate the movement of the earth relative to the sun[3]. Some fifteen years later, Jakob Bernoulli independently studied various properties of this curve, and named it 'lemniscus', the Latin for '*suspended ribbon*'.

Abel, Gauss, Euler and many more... Great mathematicians who all contributed to better understand the various properties of the lemniscate. In particular, their work on its arc lenght ultimately led to the study of elliptic curves.

To conclude this section, we briefly look at maps between projective varieties. This will allow us to formally define the concept of isomorphic varieties (to say the least). But before we do so, we recall the notion of a function field. So let V, $W \subseteq \mathbb{P}^n$ be projective varieties and $\overline{K}\left(X_0, \ldots, X_n\right)$ denote the quotient field of $\overline{K}\left[X_0, \ldots, X_n\right]$ [Hun74, p.144].

Definition 3.27 *The* function field *of V, denoted $\overline{K}(V)$, is the field whose elements are rational functions f/g, where f, $g \in \overline{K}\left[X_0, \ldots, X_n\right]$ are homogeneous polynomials of the same degree such that $g(P) \neq 0$ for at least one $P \in V$. Two such functions f_1/g_1 and f_2/g_2 will be identified[4] if $f_1(P) \cdot g_2(P) = f_2(P) \cdot g_1(P)$ for every $P \in V$.*

Definition 3.28 *A map $\varphi : V \to W$ is said to be* rational *if there are $f_0, \ldots f_n \in \overline{K}(V)$ satisfying:*

For every $P \in V$, $\varphi(P) = [f_0(P) : \ldots : f_n(P)] \in W$ as soon as all f_i are defined at P.

In this case, we adopt the notation $\varphi = [f_0, \ldots, f_n]$.

[3] Incidentally, NASA has named *Cassini* its pilotless spaceship targeted at Saturn. More details can be found at http://saturn.jpl.nasa.gov/home/index.cfm.

[4] That is, considered equal in $\overline{K}(V)$.

Definition 3.29 *A rational map $\varphi = [f_0, \ldots, f_n]$ from V to W is said to be* defined *(or* regular*) at $P \in V$ if there is an $f \in \overline{K}(V)$ such that $f \cdot f_0(P), \ldots, f \cdot f_n(P)$ are defined but not all zero at P.*

Definition 3.30 *A* morphism *is a rational map $\varphi : V \to W$ that is defined at every $P \in V$.*

Definition 3.31 *The projective varieties V and W are said to be* isomorphic *if there are morphisms $\varphi : V \to W$ and $\psi : W \to V$ such that $\psi \circ \varphi$ and $\varphi \circ \psi$ are the identity maps (on V and W respectively). In this case, we write $V \simeq W$ and say that φ is an* isomorphism *(of projective varieties).*

3.2 Plane Curves and Cryptography: A Sneak Peek

Now that we have formally defined what an algebraic curve is, we can without further waiting jump right away and try give a flavor of *why* they are so useful in cryptography. This section is therefore just a glimpse into the cryptographic applications of algebraic curves. It also provides examples and motivation to keep in mind for the theory of divisors that will come next.

We follow an approach by examples and we will try as much as possible to do things explicitely, sometimes even including small MAGMA examples. We chose to touch upon three examples: Pell equation, elliptic curves and hyperelliptic curves. These families of curves will hopefully highlight the various challenges one faces in curve-based cryptography.

Throughout this section, C will denote a plane curve.

Definition 3.32 *An algebraic curve C in $\mathbb{P}^2$ is called a (projective)* plane curve *if it is the set of solutions in $\mathbb{P}^2$ to $f(X, Y, Z) = 0$, where f is a nonconstant homogeneous polynomial.*

Two other concepts are to be introduced at this point. Notice that they are not presented in the greater generality since we will only need to use them in the context of plane curves.

Definition 3.33 *A point $P = [x : y : z] \in \mathbb{P}^2$ with $z = 0$ is said to be a* point at infinity*.*

Next we introduce the notion of a singularity.

Definition 3.34 *A point $P = [x : y : z] \in C$ is said to be* singular *if the partial derivatives f_X, f_Y and f_Z all vanishes at P:*

$$\frac{\partial f}{\partial X}(P) = \frac{\partial f}{\partial Y}(P) = \frac{\partial f}{\partial Z}(P) = 0.$$

Otherwise, P is called a smooth *(or* nonsingular*) point of C. The curve C is said to be* smooth *(or* nonsingular*) if all the points $P \in C$ are smooth.*

Remark 3.35 *This intuitive notion of smoothness of a point on a curve in $\mathbb{P}^2$ we just considered can in fact be extended and formalized for an arbitrary variety V, in terms of the local ring of P on V (that is, the ring of germs of regular functions on V near P) [Har77, p. 16, 32]. We shall however only need the above definition in the sequel.*

Example 3.36 *Consider the lemniscate $C \subseteq \mathbb{P}^2$ defined by*

$$f\left(X,Y,Z\right) = \left(X^2+Y^2\right)^2 - \left(X^2-Y^2\right)Z^2,$$

whose graph over the reals ressembles the symbol at infinity '∞', as shown in Figure 3.1.

Say we are working over a field K such that $\mathrm{Char}(K) \neq 2$. A point $P = [X_P : Y_P : Z_P] \in C$ will be singular if and only if

$$\begin{aligned} \frac{\partial f}{\partial X}\left(P\right) &= 2X_P\left(2\left(X_P^2+Y_P^2\right)-Z_P^2\right) = 0, \\ \frac{\partial f}{\partial Y}\left(P\right) &= 2Y_P\left(2\left(X_P^2+Y_P^2\right)+Z_P^2\right) = 0, \textit{ and} \\ \frac{\partial f}{\partial Z}\left(P\right) &= -2Z_P\left(X_P^2-Y_P^2\right) = 0. \end{aligned}$$

If $Z_P = 0$, then the equation $f\left(X_P, Y_P, Z_P\right) = 0$ implies that $X_P^2 + Y_P^2 = 0$. Hence,

$$P = [X_P : Y_P : 0] \textit{ is singular iff } X_P^2 + Y_P^2 = 0.$$

If $Z_P \neq 0$, then we get from the third equation that $X_P^2 - Y_P^2 = 0$. So since we must have $f\left(X_P, Y_P, Z_P\right) = 0$, it follows that $X_P^2 + Y_P^2 = 0$ as well. Thus, $X_P = 0$ and we get that $P = [0:0:1]$ is the only singular point with $Z_P \neq 0$. Let's now see how this can also be done using Magma.

```
> K:=GF(13);
> P2<X,Y,Z>:=ProjectiveSpace(K,2);
> f:=(X^2+Y^2)^2-(X^2-Y^2)*Z^2;
> Lemniscate := Curve(P2,f);
> IsIrreducible(Lemniscate);
  true
> Dimension(Lemniscate);
  1
> Points(Lemniscate);
  {@ (0 : 0 : 1), (1 : 0 : 1), (12: 0 : 1), (0 : 5 : 1),
     (4 : 5 : 1), (9 : 5 : 1), (1 : 6 : 1), (12: 6 : 1),
     (1 : 7 : 1), (12: 7 : 1), (0 : 8 : 1), (4 : 8 : 1),
     (9 : 8 : 1), (5 : 1 : 0), (8 : 1 : 0) @}
> IsSingular(Lemniscate);
```

```
  true
> P:=P2![1,0,1];
> IsSingular(Lemniscate,P);
  false
> TangentLine(Lemniscate,P);
  Curve over GF(13) defined by X + 12*Z
> S:=P2![0,0,1];
> S in Lemniscate;
  true (0 : 0 : 1)
> IsSingular(Lemniscate,S);
  true
> IsCusp(Lemniscate,S);
  false
> IsNode(Lemniscate,S);
  true
> TangentLine(Lemniscate,S);
  >> TangentLine(Lemniscate,S);
                ^

  Runtime error in 'TangentLine': Argument 2 must be nonsingular point
  of argument 1
> L1:=Curve(P2,X-Y);
> L2:=Curve(P2,X+Y);
> IsTangent(Lemniscate,L1,S);
  true
> IsTangent(Lemniscate,L2,S);
  true
> SingularPoints(Lemniscate);
  {@ (0 : 0 : 1), (5 : 1 : 0), (8 : 1 : 0) @}
> IsNode(Lemniscate,P2![5,1,0]);
  true
> IsNode(Lemniscate,P2![8,1,0]);
  true
```

The lemniscate thus has a singularity at $[0:0:1]$*. For obvious reasons, this type of singularity is called a node. Remark that there are two distinct tangent lines at the origin and that in our example, the two points at infinity were also singular points.*

Remark 3.37 *There are various types of singularities of plane curves, like nodes, cusps, triple point or tacnode. See [Har77, Figure 4, p.36] for details.*

3.2.1 Pell Equation: A Case Study for Torus-based Cryptography

It is now time to study concrete curves with cryptographic applications in mind. The Pell equation has the tremendous advantage of being simple enough to be fully and concisely treated from first principles. It also makes the perfect example of how one shows that a particular group is relevant for cryptographic purposes. And most importantly, the Pell equation naturally leads to the study of algebraic tori.

The *Pell equation* over a field K is of the form

$$x^2 - Dy^2 = 1,$$

where $D \in K$ is not a square. The associated curve C, the so-called *Pell conic*, is thus a hyperbola with asymptotes

$$y = \pm \frac{x}{\sqrt{D}}.$$

Figure 3.2 illustrates the Pell conic over the real numbers.

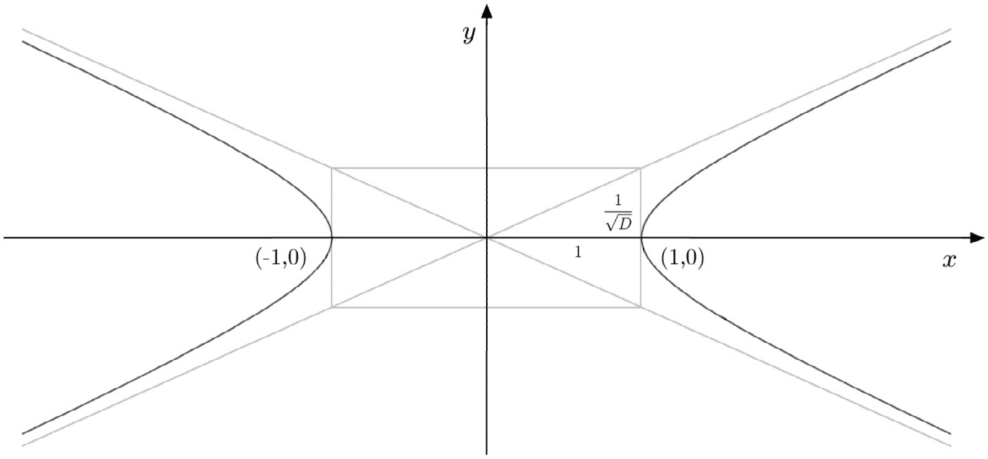


Figure 3.2: Pell conic over the real numbers

From a historical point of view, Pell equation certainly is one of the most broadly spread mathematical misunderstandings. Indeed, Pell himself had little to do with “his equation”: the confusion comes from the fact that Euler falsely attributed to Pell a method that had actually been found by another English mathematician, William Brouncker, in response to a challenge from Fermat. Besides, Brouncker's method itself might be considered as a re-discovery, since it is now known that Indians mathematicians of the 10th century A.D. (such as Jayadeva) had already developped such methods[5]. For a detailed historical approach to Pell Equation, see [Len02].

[5] Actually, one can find traces of such equations back to Ancient Greek, as seen in *The Cattle Problem*, a mathematical problem posed in the form of a poem commonly attributed to Archimedes. It is however not known whether they knew how to resolve these problems.

Now that we know a little about the history of these curves, we can officially start our cryptographic explorations. The first remark in order is that the projective equation $X^2 - DY^2 = Z^2$ associated to the Pell equation does not contain any point of the form $P = [x : y : z] \in \mathbb{P}^2(K)$ with $z = 0$. Indeed, $x^2 - Dy^2 = z^2$ yields the equation $x^2 = Dy^2$. If $y = 0$, then $x = y = z = 0$, which violates the fact that at least one coordinate must be nonzero. If $y \neq 0$, then $D = (x/y)^2$, which is also impossible since D is not a square. For this reason, we will here sometimes work directly with the equation $x^2 - Dy^2 = 1$.

Before we describe the geometric group law on this conic, we present a tiny MAGMA example.

Example 3.38 *Consider the Pell conic C defined by the equation*

$$x^2 - 5y^2 = 1$$

over $\mathbb{F}_{13}$. Using exhaustive search, we readily get that the points of $C(\mathbb{F}_{13})$ are

$$\begin{array}{ccccccc} (1,0) & (4,4) & (3,5) & (5,6) & (5,7) & (3,8) & (4,9) \\ (12,0) & (9,4) & (10,5) & (8,6) & (8,7) & (10,8) & (9,9) \end{array}$$

We now quickly show how one can also obtain these results using MAGMA*.*

```
> K:=GF(13);
> P2<X,Y,Z>:=ProjectiveSpace(K,2);
> D:=5;
> f:=X^2-D*Y^2-Z^2;
> Pell:= Conic(P2,f);
> IsIrreducible(Pell);
  true
> Dimension(Pell);
  1
> Points(Pell);
  {@ (1 : 0 : 1), (12: 0 : 1), (4 : 4 : 1), (9 : 4 : 1),
     (3 : 5 : 1), (10: 5 : 1), (5 : 6 : 1), (8 : 6 : 1),
     (5 : 7 : 1), (8 : 7 : 1), (3 : 8 : 1), (10: 8 : 1),
     (4 : 9 : 1), (9 : 9 : 1) @}
```

The Geometric Group Law

In this section, we will follow the advice of Silverman and Tate as we will try to first think geometrically and then to prove algebraically. We begin by introducing the somehow forgotten *chord-and-tangent rule* on the Pell conic, which is depicted in Figure 3.3. A good account on this topic can also be found in *Higher descent on Pell conics III: The first 2-descent* [Lem03].

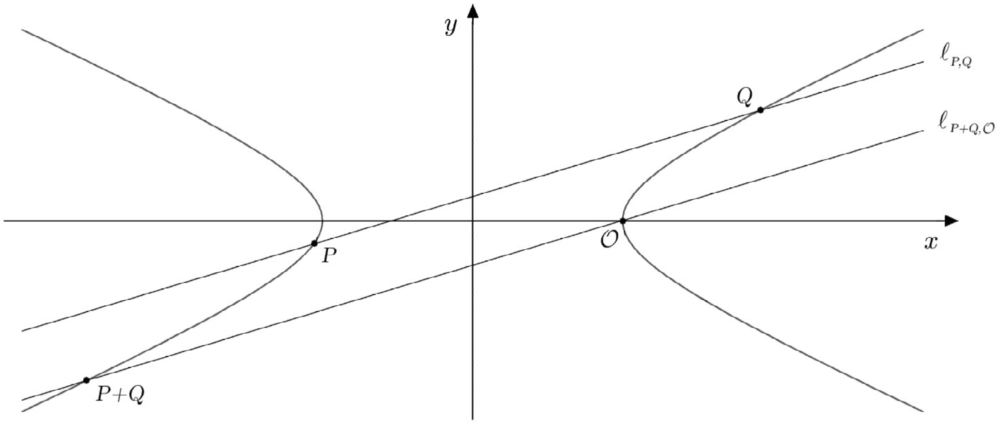


Figure 3.3: Chord-and-tangent rule on Pell conic

Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points of $C(K)$ and denote by $\ell_{P,Q}$ the line passing through P and Q if $P \neq Q$, and tangent to the curve at P if $P = Q$. The identity element of this group law will be the point $\mathcal{O} = (1, 0)$. Next consider the parallel line $\ell' = \ell_{P+Q,\mathcal{O}}$ to $\ell_{P,Q}$ that passes through $\mathcal{O} = (1, 0)$. As we will see, this line intersects C at precisely[6] one other point R. Finally, let $P + Q = R$.

Let's now compute the coordinates of R, and we of course begin with the easy cases. First, we can verify that $P+\mathcal{O} = P$ and $\mathcal{O}+Q = Q$, and so we can now assume that P, $Q \neq \mathcal{O}$. Also, if $x_1 = x_2$ and $y_1 = -y_2$, then $\ell_{P,Q}$ is a vertical line, from which follows that $P + Q = \mathcal{O}$.

Otherwise, the equation of the line $\ell_{P,Q}$ passing through P and Q is

$$y = \mathbf{m}x + \mathbf{b},$$

where

$$\mathbf{m} = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2, \\ \dfrac{x_1}{Dy_1} & \text{if } P = Q \text{ and } y_1 \neq 0. \end{cases}$$

Remark that one way of obtaining this value of $\mathbf{m}$ when $P = Q$ and $y_1 \neq 0$ is to use implicit differentiation of $x^2 - Dy^2 = 1$. Also notice that $\mathbf{m} \in K$ as soon as P, $Q \in C(K)$.

By construction, the first point of intersection of ℓ' with the hyperbola is $\mathcal{O} = (1, 0)$. Now, the equation of ℓ' is given by $y = \mathbf{m}(x - 1)$, and so we are looking for a point $R = (x_3, y_3)$ satisfying

$$x_3^2 - Dy_3^2 = 1 \text{ and } y_3 = \mathbf{m}(x_3 - 1).$$

[6] Counting multiplicities.

Thus,

$$1 = x_3^2 - Dy_3^2 = x_3^2 - D\mathbf{m}^2\left(x_3 - 1\right)^2 = x_3^2 - D\mathbf{m}^2(x_3^2 - 2x_3 + 1),$$

and so

$$(1 - D\mathbf{m}^2)x_3^2 + 2D\mathbf{m}^2x_3 - D\mathbf{m}^2 - 1 = 0. \tag{3.1}$$

But we already know that 1 is a solution of (3.1) since $\mathcal{O} = (1, 0)$ is a point of intersection of C with ℓ'. Indeed,

$$(1 - D\mathbf{m}^2)x_3^2 + 2D\mathbf{m}^2x_3 - D\mathbf{m}^2 - 1 = (x_3 - 1)\left(\left(1 - D\mathbf{m}^2\right)x_3 + D\mathbf{m}^2 + 1\right).$$

In addition, we have $1 - D\mathbf{m}^2 \neq 0$ (since otherwise, $D\mathbf{m}^2 = 1$ with $\mathbf{m} \neq 0$, and so $D = \left(1/\mathbf{m}\right)^2$, a contradiction). Thus,

$$x_3 = -\frac{D\mathbf{m}^2 + 1}{1 - D\mathbf{m}^2} = \frac{D\mathbf{m}^2 + 1}{D\mathbf{m}^2 - 1}$$

and

$$y_3 = \mathbf{m}\left(x_3 - 1\right) = \mathbf{m}\left(\frac{D\mathbf{m}^2 + 1}{D\mathbf{m}^2 - 1} - 1\right) = \frac{2\mathbf{m}}{D\mathbf{m}^2 - 1}.$$

Finally,

$$R = \left(\frac{D\mathbf{m}^2 + 1}{D\mathbf{m}^2 - 1}, \frac{2\mathbf{m}}{D\mathbf{m}^2 - 1}\right),$$

and we indeed have $R \in C\left(K\right)$.

Example 3.39 *We return to the previous example. The multiples of $P = (3, 5)$ are shown in the following table.*

$P =$	$(3, 5)$	$6P =$	$(10, 5)$	$11P =$	$(8, 7)$
$2P =$	$(4, 4)$	$7P =$	$(12, 0)$	$12P =$	$(4, 9)$
$3P =$	$(8, 6)$	$8P =$	$(10, 8)$	$13P =$	$(3, 8)$
$4P =$	$(5, 6)$	$9P =$	$(9, 9)$	$14P =$	$(1, 0)$
$5P =$	$(9, 4)$	$10P =$	$(5, 7)$		

Hence, $C\left(\mathbb{F}_{13}\right)$ is a cyclic group of order $14 = 13 + 1$; we will see in the next section that this principle always apply.

Let's now see how that could be computed with MAGMA*. Since the beginning of our series of examples, we used* MAGMA *on a command mode. Fortunately,* MAGMA *also allows us to run a program previously typed in any text editor, as illustrated in the following example.*

```
K:=GF(13);
P2<X,Y,Z>:=ProjectiveSpace(K,2);
D:=5;
f:=X^2-D*Y^2-Z^2;
```

```
Pell:= Conic(P2,f);
O:=P2![1,0,1];                                              // Identity element
P:=P2![3,5,1];                                             // Chosen Base Point
Inverse := func< P | P2![P[1],-P[2],P[3]] >;                     // Computes -P
print ‘‘-P =’’,Inverse(P);
Add:=function(P,Q)                                              // Computes P+Q
   if (P[1] ne Q[1]) then m:=(P[2]-Q[2])/(P[1]-Q[1]);
      elif (P[2] eq -Q[2]) then return O;                         // Case P =-Q
      else m:= P[1]/(D*P[2]);                        // Case P = Q and P+Q != O
   end if;
   Z3:= D*m^2-1;             // To store numerators and denominators separately
                                                         // to avoid inversions
   X3:= Z3+2;                        // To save computations since X3:= D*m^2+1
   Y3:= 2*m;
   return P2![X3,Y3,Z3];
end function;
n:=0;
Q:=P;
repeat            // Loop that prints the multiples of P until O is encountered
   n:=n+1;
   print n,‘‘ * P =’’,Q;
   Q:=Add(P,Q);
until Q eq P;             // The last entry printed is Ord(P) * P = (1 : 0 : 1)
print ‘‘Ord(P) =’’,n;
Subtract := func< P,Q | Add(P,Inverse(Q)) >;                     //Computes P-Q
print ‘‘((P+P)+P)-P =’’, Subtract(Add(Add(P,P),P),P);
```

Take note that once a ‘return’ is encountered, the last part of the function is not evaluated. We obtain the following output when the above program is run.

```
> load ‘‘C:/MAGMA/Pell.mag’’;
Loading ‘‘C:/MAGMA/Pell.mag’’
-P = (3 : 8 : 1)
1 * P = (3 : 5 : 1)
2 * P = (4 : 4 : 1)
3 * P = (8 : 6 : 1)
4 * P = (5 : 6 : 1)
5 * P = (9 : 4 : 1)
6 * P = (10 : 5 : 1)
7 * P = (12 : 0 : 1)
8 * P = (10 : 8 : 1)
9 * P = (9 : 9 : 1)
10 * P = (5 : 7 : 1)
11 * P = (8 : 7 : 1)
```

```
12 * P = (4 : 9 : 1)
13 * P = (3 : 8 : 1)
14 * P = (1 : 0 : 1)
Ord(P) = 14
((P+P)+P)-P = (4 : 4 : 1)
```

Lastly, notice that our program could have also included interactive inputs in order to change the basepoint P at will.

Everyone who is familiar with the chord-and-tangent rule on an elliptic curve will have noticed the similarities between these two geometric group laws. However, in the case of Pell equation, the formulæ can be greatly simplified[7].

Indeed, first remark that if $P = Q$ and $y_1 \neq 0$, then

$$D\mathbf{m}^2 \pm 1 = D \cdot \frac{x_1^2}{D^2 y_1^2} \pm 1 = \frac{x_1^2}{D y_1^2} \pm 1 = \frac{x_1^2 \pm D y_1^2}{D y_1^2}.$$

Hence,

$$x_3 = \frac{D\mathbf{m}^2 + 1}{D\mathbf{m}^2 - 1} = \frac{x_1^2 + Dy_1^2}{x_1^2 - Dy_1^2} = \frac{x_1^2 + Dy_1^2}{1} = x_1^2 + Dy_1^2 = x_1 x_2 + D y_1 y_2$$

and

$$\begin{aligned} y_3 &= \mathbf{m}\,(x_3 - 1) = \mathbf{m}\left(x_1^2 + Dy_1^2 - 1\right) = \mathbf{m}\left(x_1^2 + Dy_1^2 - (x_1^2 - Dy_1^2)\right) \\ &= 2\mathbf{m}Dy_1^2 = 2\frac{x_1}{Dy_1} Dy_1^2 = 2x_1 y_1 = x_1 y_2 + x_2 y_1. \end{aligned}$$

Thus,

$$R = (x_1 x_2 + D y_1 y_2, x_1 y_2 + x_2 y_1) \text{ as soon as } P = Q \text{ and } y_1 \neq 0.$$

Moreover, if $P = \mathcal{O} = (1, 0)$, then

$$R = \mathcal{O} + Q = Q = (x_2, y_2) = (x_1 x_2 + D y_1 y_2, x_1 y_2 + x_2 y_1)$$

as well and by symmetry, it also holds for the case $Q = \mathcal{O}$. That's not all. If $x_1 = x_2$ and $y_1 = -y_2$, then

$$R = \mathcal{O} = (1, 0) = \left(x_1^2 - Dy_1^2, x_1 y_2 - x_1 y_2\right) = (x_1 x_2 + D y_1 y_2, x_1 y_2 + x_2 y_1).$$

In order to show that R always equals $(x_1 x_2 + D y_1 y_2, x_1 y_2 + x_2 y_1)$, it only remains to treat the case where P, $Q \neq \mathcal{O}$, $P \neq Q$ and $x_1 \neq x_2$. First, subtracting $x_2^2 - Dy_2^2 = 1$ from

[7] The original motivation was to find points with integer coordinates, and not merely with rational coordinates. For more details, see [Lem03].

$x_1^2 - Dy_1^2 = 1$ yields $x_1^2 - x_2^2 + D\left(y_2^2 - y_1^2\right) = 0$. That is, $D\left(y_2^2 - y_1^2\right) = x_2^2 - x_1^2$. Thus, $D\left(y_2 - y_1\right)\left(y_2 + y_1\right) = \left(x_2 - x_1\right)\left(x_2 + x_1\right)$. Two cases then arise: $y_2 + y_1 = 0$ and $y_2 + y_1 \neq 0$.

If $y_2 + y_1 = 0$, then $\left(x_2 - x_1\right)\left(x_2 + x_1\right) = 0$ with $x_1 \neq x_2$. Thus, $x_2 + x_1 = 0$ and so $P = (-x_2, -y_2)$. It then follows that $\mathbf{m} = y_1 / x_1$ and

$$x_3 = \frac{D\mathbf{m}^2 + 1}{D\mathbf{m}^2 - 1} = \frac{\left(Dy_1^2 + x_1^2\right) / x_1^2}{\left(Dy_1^2 - x_1^2\right) / x_1^2} = \frac{2x_1^2 - 1}{-1} = 1 - 2x_1^2 = -x_1^2 - Dy_1^2 = x_1x_2 + Dy_1y_2$$

and

$$y_3 = \mathbf{m}\left(x_3 - 1\right) = \frac{y_1}{x_1}\left(1 - 2x_1^2 - 1\right) = -2x_1y_1 = x_1y_2 + x_2y_1,$$

which shows that $R = (x_1x_2 + Dy_1y_2, x_1y_2 + x_2y_1)$ in this case as well.

If $y_2 + y_1 \neq 0$, then $D\left(y_2 - y_1\right)\left(y_2 + y_1\right) = \left(x_2 - x_1\right)\left(x_2 + x_1\right)$ implies that

$$D\mathbf{m} = D \cdot \frac{y_2 - y_1}{x_2 - x_1} = \frac{x_2 + x_1}{y_2 + y_1}$$

and

$$D\mathbf{m}^2 \pm 1 = \frac{\left(x_1 + x_2\right)\left(y_2 - y_1\right)}{\left(y_1 + y_2\right)\left(x_2 - x_1\right)} \pm 1 = \frac{\left(x_1 + x_2\right)\left(y_2 - y_1\right) \pm \left(x_2 - x_1\right)\left(y_1 + y_2\right)}{\left(x_2 - x_1\right)\left(y_1 + y_2\right)}.$$

Therefore,

$$x_3 = \frac{D\mathbf{m}^2 + 1}{D\mathbf{m}^2 - 1} = \frac{\left(x_1 + x_2\right)\left(y_2 - y_1\right) + \left(x_2 - x_1\right)\left(y_1 + y_2\right)}{\left(x_1 + x_2\right)\left(y_2 - y_1\right) - \left(x_2 - x_1\right)\left(y_1 + y_2\right)}.$$

The numerator of this last expression can be rewritten as

$$\left(x_1 + x_2\right)\left(y_2 - y_1\right) + \left(x_2 - x_1\right)\left(y_1 + y_2\right) = 2\left(x_2y_2 - x_1y_1\right),$$

while the denominator can be expressed as

$$\left(x_1 + x_2\right)\left(y_2 - y_1\right) - \left(x_2 - x_1\right)\left(y_1 + y_2\right) = 2\left(x_1y_2 - x_2y_1\right).$$

As a result,

$$x_3 = \frac{x_2y_2 - x_1y_1}{x_1y_2 - x_2y_1}.$$

Now, notice that

$$\begin{aligned}
\left(x_1x_2 + Dy_1y_2\right)\left(x_1y_2 - x_2y_1\right) &= x_1^2x_2y_2 - x_2^2x_1y_1 + Dy_2^2 \cdot x_1y_1 - Dy_1^2 \cdot x_2y_2 \\
&= x_1^2x_2y_2 - x_2^2x_1y_1 + \left(x_2^2 - 1\right)x_1y_1 - \left(x_1^2 - 1\right)x_2y_2 \\
&= x_1^2x_2y_2 - x_2^2x_1y_1 - x_1y_1 + x_2y_2 - x_1^2x_2y_2 + x_2^2x_1y_1 \\
&= x_2y_2 - x_1y_1.
\end{aligned}$$

Finally, this implies that

$$x_3 = \frac{x_2y_2 - x_1y_1}{x_1y_2 - x_2y_1} = \frac{(x_1x_2 + Dy_1y_2)(x_1y_2 - x_2y_1)}{x_1y_2 - x_2y_1} = x_1x_2 + Dy_1y_2.$$

On the other hand,

$$y_3 = \mathbf{m}(x_3 - 1) = \frac{(y_2 - y_1)(x_1x_2 + Dy_1y_2 - 1)}{x_2 - x_1}.$$

In order to simplify this last expression, we can rewrite its numerator as follows:

$$\begin{aligned}
(y_2 - y_1)(x_1x_2 + Dy_1y_2 - 1) &= x_1x_2y_2 + y_1 \cdot Dy_2^2 - y_2 - x_1x_2y_1 - y_2 \cdot Dy_1^2 + y_1 \\
&= x_1x_2y_2 + y_1\left(x_2^2 - 1\right) - y_2 - x_1x_2y_1 - y_2\left(x_1^2 - 1\right) + y_1 \\
&= x_1x_2y_2 + x_2^2y_1 - y_1 - y_2 - x_1x_2y_1 - x_1^2y_2 + y_2 + y_1 \\
&= x_2(x_1y_2 + x_2y_1) - x_1(x_1y_2 + x_2y_1) \\
&= (x_2 - x_1)(x_1y_2 + x_2y_1).
\end{aligned}$$

And we get that

$$y_3 = \frac{(y_2 - y_1)(x_1x_2 + Dy_1y_2 - 1)}{x_2 - x_1} = \frac{(x_2 - x_1)(x_1y_2 + x_2y_1)}{x_2 - x_1} = x_1y_2 + x_2y_1.$$

We can then at last conclude that $R = (x_1x_2 + Dy_1y_2, x_1y_2 + x_2y_1)$.

We have therefore shown, using only simple (but tedious!) algebraic manipulations, that we *always* have

$$P + Q = (x_1x_2 + Dy_1y_2, x_1y_2 + x_2y_1).$$

Using this compact expression, it is now a simple matter to show that the chord-and-tangent rule on the Pell conic indeed defines a group law.

Lemma 3.40 *Let K be a field and C be the Pell conic*

$$x^2 - Dy^2 = 1,$$

where $D \in K$ is not a square. Then, $C(K)$ with the chord-and-tangent rule as binary operation is an abelian group with identity $\mathcal{O} = (1, 0)$. This group operation can be performed as follows. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ in $C(K)$ be given. Then,

$$P + Q = (x_1x_2 + Dy_1y_2, x_1y_2 + x_2y_1) \text{ and } -P = (x_1, -y_1).$$

The cost of computing this group law is then four general multiplications in K, plus a multiplication by a constant (which will be abbreviated by $4\mathbf{M} + \mathbf{C}$).

Remark 3.41 *Notice that there is no longer a large difference between addition and doubling with the simplified formulæ; such "unified formulæ" present interesting cryptographic properties, notably because they contribute to protect against side-channel attacks.*

Group Order

Now that we know that $C(\mathbb{F}_q)$ is a group, the next step is to determine its cardinality. Even without knowing it, we have parametrized the points on the curve. The natural way to proceed is thus to use the parametrization we first obtained with the chord-and-tangent rule. Indeed, we can do a projection of this conic on the y-axis, as illustrated in Figure 3.4 over the reals (notice that $1/\sqrt{D} \notin \mathbb{Q}$).

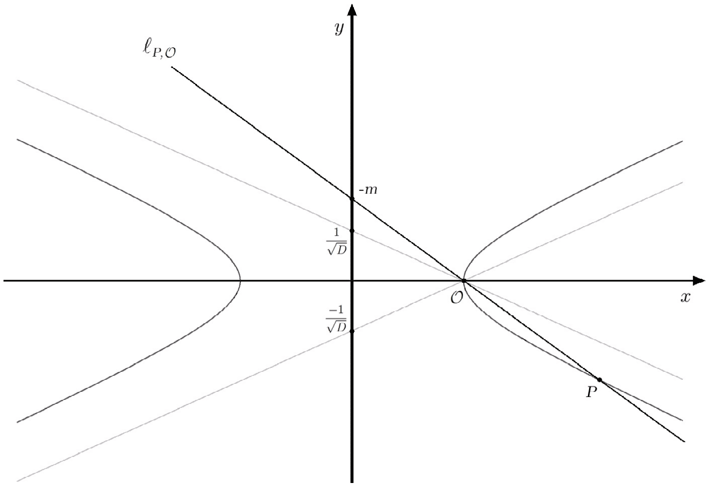


Figure 3.4: Projection of the Pell conic on the y-axis

So let $P = (x, y) \in C(\mathbb{F}_q)$ be a point different from $\mathcal{O} = (1, 0)$. If we simply apply the original addition formula that we obtained, we have that

$$P + \mathcal{O} = P, \text{ and thus that } (x, y) = \left(\frac{D\mathbf{m}^2 + 1}{D\mathbf{m}^2 - 1}, \frac{2\mathbf{m}}{D\mathbf{m}^2 - 1} \right),$$

where $\mathbf{m}$ is the slope of the straight line $\ell_{P,\mathcal{O}}$ passing through P and $\mathcal{O}$. So for each $\mathbf{m} \in \mathbb{F}_q$, we can associate a point of $C(\mathbb{F}_q)$, and conversely, $P = (x, y) \in C(\mathbb{F}_q) \setminus \{\mathcal{O}\}$ implies that $\mathbf{m} \in \mathbb{F}_q$ since

$$\mathbf{m} = \begin{cases} 0 & \text{if } P = (-1, 0), \\ \dfrac{y}{x-1} & \text{if } x \neq 1. \end{cases}$$

We thus have a mapping

$$\begin{array}{rccc} \varphi : & C(\mathbb{F}_q) & \longrightarrow & \mathbb{P}^1(\mathbb{F}_q) \\ & P = (x, y) & \longmapsto & \begin{cases} [1 : 0] & \text{if } P = \mathcal{O} \\ [y : x - 1] & \text{otherwise.} \end{cases} \end{array}$$

which can be seen to be a projection of the Pell conic on the y-axis[8]. It is a routine exercise to verify that φ is a well-defined bijection of sets with inverse

$$\begin{array}{rccc}\psi: & \mathbb{P}^1(\mathbb{F}_q) & \longrightarrow & C(\mathbb{F}_q) \\ & [\mathbf{m}:1] & \longmapsto & \left(\dfrac{D\mathbf{m}^2+1}{D\mathbf{m}^2-1}, \dfrac{2\mathbf{m}}{D\mathbf{m}^2-1}\right) \\ & [1:0] & \longmapsto & (1,0)\end{array}$$

Thus, $\#C(\mathbb{F}_q) = q+1$. Moreover, notice that both φ and ψ can be computed efficiently. So if Alice wants to send a point P on this curve to Bob, then she can do so by simply sending an element of $\mathbb{F}_q \cup \{\infty\}$ instead of transmitting the pair $(x,y) \in \mathbb{F}_q \times \mathbb{F}_q$.

Alice		Bob
Compress P by computing $P' = \varphi(P)$	$\xrightarrow{P'}$	Recover P by evaluating $\psi(P')$

It is true that in this case we don't save that much since Alice could have sent x together with one bit to specify which square root Bob should take for y. However, this idea can be generalized and applied to subgroups of $\mathbb{F}_{p^6}$ for which elements can be represented using only two elements of $\mathbb{F}_p$ instead of six: this is the cryptosystem CEILIDH. This idea of reducing the amount of information that needs to be exchanged is the main selling feature of torus-based cryptography.

Group Structure

Now that we know that we are working with a group of order $q+1$, we may wonder what its group structure is. To do so, the easiest way is to exploit the fact that the simplified group law we obtained seems quite familiar. Indeed, since D is not a square, then $\sqrt{D} \notin \mathbb{F}_q$. Thus, the polynomial $f(x) = x^2 - D = \left(x - \sqrt{D}\right)\left(x + \sqrt{D}\right)$ is irreducible over $\mathbb{F}_q$. As a result, we know that (see [Hun74, Theorem V.1.6])

$$\mathbb{F}_q\left(\sqrt{D}\right) \cong \mathbb{F}_q[x]/(f) \cong \mathbb{F}_{q^2}.$$

So given $x_1 + y_1\sqrt{D}$, $x_2 + y_2\sqrt{D} \in \mathbb{F}_q\left(\sqrt{D}\right)$, we have

$$\left(x_1 + y_1\sqrt{D}\right)\left(x_2 + y_2\sqrt{D}\right) = (x_1x_2 + Dy_1y_2) + (x_1y_2 + x_2y_1)\sqrt{D},$$

[8] This is in fact the same idea as the parametrization of the circle. See [ST92, Section I.1].

which coincides with the group law for Pell equation obtained in Lemma 3.40 . We therefore have the following one-to-one group homomorphism

$$\begin{array}{cccc} \varphi : & C\left(\mathbb{F}_q\right) & \longrightarrow & \mathbb{F}_{q^2}^* \\ & P = \left(x_1, y_1\right) & \longmapsto & x_1 + y_1\sqrt{D}. \end{array}$$

Thus, $C\left(\mathbb{F}_q\right)$ is isomorphic to a subgroup of $\mathbb{F}_{q^2}^*$ of order $q+1$. But $\mathbb{F}_{q^2}^*$ is a cyclic group of order $q^2 - 1 = (q-1)(q+1)$, and therefore contains a *unique*[9] subgroup of order $q+1$, namely

$$\left\langle \left(x_0 + y_0\sqrt{D}\right)^{q-1} \right\rangle,$$

where $x_0 + y_0\sqrt{D}$ is a generator of $\mathbb{F}_{q^2}^*$. There is therefore a natural way to view the $\mathbb{F}_q$-points on Pell conic as a subgroup of $\mathbb{F}_{q^2}^*$. So we know that $C\left(\mathbb{F}_q\right)$ *always*[10] is a cyclic group of order $q+1$.

This is where it becomes truly interesting: this subgroup of $\mathbb{F}_{q^2}^*$ in fact coincides[11] with the 1-dimensional algebraic torus $T_2\left(\mathbb{F}_q\right)$ used by Rubin and Silverberg as one of the two explicit cryptosystems described in their CRYPTO 2003 paper[12] [RS03]. More generally, we have that the $\varphi(n)$-dimensional torus $T_n\left(\mathbb{F}_q\right)$ can be identified with the unique cyclic subgroup of $\mathbb{F}_{q^n}^*$ containing $\Phi_n(q)$ elements, where φ is the Euler function and Φ_n is the n-th cyclotomic polynomial [RS03, Lemma 7]. In certain cases (e.g. if $n \geq 2$ is divisible by at most two primes), the existence of a rational parametrization allows to compactly represent the elements of $T_n\left(\mathbb{F}_q\right)$ using only $\varphi(n)$ elements of $\mathbb{F}_q$ (instead of n).

The Discrete Logarithm Problem

A cucial question that we have not yet addressed is the overall difficulty of the discrete logarithm problem in $C\left(\mathbb{F}_q\right)$. Without a doubt, we should approach this problem by thinking of $C\left(\mathbb{F}_q\right)$ as *'a subgroup of* $\mathbb{F}_{q^2}^*$*'* since the discrete logarithm problem in finite fields has been intensively studied since the birth of public-key cryptography. The first (and obvious) remark in order is that any algorithm that can extract DLPs in all of $\mathbb{F}_{q^2}^*$, index-calculus for instance, can also be used to solve DLPs in $C\left(\mathbb{F}_q\right)$. Nevertheless, it is still possible that DLPs in $C\left(\mathbb{F}_q\right)$ be faster to solve than in $\mathbb{F}_{q^2}^*$. From a cryptanalysis point of view, the two main differences between $C\left(\mathbb{F}_q\right)$ and $\mathbb{F}_{q^2}^*$ are their sizes ($\#C\left(\mathbb{F}_q\right)$ being roughly the squareroot of $\#\mathbb{F}_{q^2}^*$) and the fact that any element of $C\left(\mathbb{F}_q\right)$ can be easily and compactly represented using only one element of $\mathbb{F}_q \cup \{\infty\}$.

[9] Recall that if G is a cyclic group of order n and $k \mid n$, then G has exactly one subgroup of order k [Hun74, Ex 6, p.37].

[10] In comparison, there are many possible scenarios with elliptic curves, as we will see in the next section.

[11] I want to thank Alfred Menezes for pointing this out to me.

[12] Even if they never mentionned Pell equation in the paper!

The first index-calculus algorithm[13] especially targeted at algebraic tori was presented at CRYPTO 2005 [GV05]: Robert Granger[14] and Frederik Vercauteren had the idea of exploiting the compact representation of the elements in $T_n(\mathbb{F}_q)$ to carefully choose a new factor base (also called a *decomposition base* in that context) that would speed up the attack. Here is how they describe their algorithm:

> *"[It] exploits the compact representation of elements of rational tori. The very existence of such an algorithm shows that the lower communication cost offered by these tori, may also be exploited by the cryptanalyst."*

Section 4 of the paper is devoted to explicitly describe the attack on $T_2(\mathbb{F}_{q^m})$ when q is an odd prime power. The complexity analysis and implementation of the algorithm reveal that it is already faster than Pollard's Rho method (see Section 2.7.1) as soon as $m \geq 5$. It therefore does not yet represent a practical threat for $T_2(\mathbb{F}_p)$ when $p > 2$ is prime.

However, at the moment these lines were written down, Granger and Vercauteren were testing the algorithm using a prototype implemetation inMagma. The running times they obtained should then only be considered as upper bounds of what will actually be achieved with an optimized code. Further developments are thus soon to be expected...

To sum up, we have shown in this section that the Pell conic in fact fulfills the main requirements for a group G to be suitable for cryptographic applications. That is,

- *The elements of G can be easily represented in a compact form,*
- *The group operation can be performed efficiently,*
- *The discrete logarithm problem in G is believed to be intractable, and*
- *The group order can be efficiently computed.*

Moreover, notice that we have achieved this goal *'from scratch'*,using elementary techniques throughout. It is hoped that this case study will give a flavor of what should be expected in the sequel, as we will need to show that these conditions are also met for a generalized Jacobian of an elliptic curve in Chapter 5.

[13] For a general description of the principles behind index-calculus attacks, please refer to Section 2.7.2.

[14] I wish to thank Robert for taking the time to patiently answer my numerous questions on this subject around a cup of coffee.

3.2.2 Elliptic Curves

"The theory of elliptic curves is rich, varied, and amazingly vast" once wrote Joseph Silverman [Sil86, p.2]. Indeed, the fascination for the esthetic beauty of these curves allowed to develop the tools that would later turn out to play a key role in many applications, the most famous being the proof of Fermat's Last Theorem (FLT) [Wil95, TW95]. Their versatility is astonishing: in public-key cryptography alone, they are used for primality testing [AM93], factoring large integers [Len87], digital signatures (ECDSA) [NIoST00] and of course for encryption [Mil86b, Kob87]. The industry is now also falling under their charm: the shorter keys required for elliptic curve cryptosystems is an attractive selling feature, especially for small cryptographic devices, like smart cards, Personal Digital Assistants (PDAs) or cell phones [Mic02]. Moreover, government agencies also rely on elliptic curves to protect sensitive information:

> *"National Security Agency (NSA) selected Elliptic Curve Cryptography (ECC) as the exclusive key agreement and digital signature standard to secure sensitive but unclassified data within the U.S. government"*[15]

The goal of this section will nonetheless be very modest, as we simply wish to recall the milestones that make elliptic curves such a unique candidate for DL-based cryptography. First, there are many equivalent ways of defining elliptic curves (see [Ols73, p.173]), but perhaps the most natural for the applications we have in mind is the following:

Definition 3.42 *An* elliptic curve *is an algebraic curve of genus one together with a distinguished point $\mathcal{O} \in E$. Moreover, we say that this elliptic curve is* defined over K *if* $\mathcal{O} \in E(K)$ *and E is defined over K as an algebraic curve.*

Remark 3.43 *So strictly speaking, an elliptic curve is a* pair $(E, \mathcal{O})$. *We shall however often say 'let E be an elliptic curve' as soon as it is clear from the context which distinguished point we consider.*

Remark 3.44 *By definition, notice that if $(E, \mathcal{O})$ is an elliptic curve defined over K, then $E(K)$ is never empty as $\mathcal{O} \in E(K)$.*

Remark 3.45 *Notice that this definition does not require that the curve E be smooth.*

Of course, this definition does not say much if one is not familiar with the notion of genus. For now, let's just say that the genus of a curve is a nonnegative integer that somehow gives a

[15] From Certicom press release, June 9, 2005. Available at `http://www.certicom.com/index.php?action=company,press_archive&view=494`.

measure of its complexity. For instance, $\mathbb{P}^1$ and the Pell conic both have genus zero [Sil86, Ex. II.5.6] and are thus among the simplest curves that we can study. The formal definition of the genus is part of the Riemann-Roch theorem and requires familiarity with divisors (and so will have to wait until Section 3.3.4).

Weierstraß Equations

That being said, the very first step towards a concrete cryptographic application would be to see what the equation of an elliptic curve may look like. To do so, let's first consider the famous Weierstraß equations.

Definition 3.46 *A polynomial equation*

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3, \tag{3.2}$$

with coefficients[16] a_1*,* a_3*,* a_2*,* a_4*,* a_6 *in* $\overline{K}$ *is called a* Weierstraß equation*.*

Now, let $P = [X : Y : Z] \in \mathbb{P}^2$ satisfying (3.2) be given. If $Z = 0$, then we must have $P = [0 : 1 : 0]$. Otherwise, let $x = X/Z$, $y = Y/Z$. Thus, $P = [X : Y : Z] = [x : y : 1]$ and the equation (3.2) becomes

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \tag{3.3}$$

So for simplicity, we often write a Weierstraß equations using (3.3) instead of (3.2). We shall also follow this convention from time to time, remembering to add the point at infinity $[0 : 1 : 0]$ to the set of solutions of (3.3).

The following theorem, whose proof can be found in [Sil86, Prop. III.3.1], also relies on the Riemann-Roch theorem (see Theorem 3.79): it establishes the well-known link between Weierstraß equations and elliptic curves.

Theorem 3.47 *Any smooth curve given by a Weierstraß equation*

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

with coefficients in K *is an elliptic curve defined over* K *with distinguished point* $[0 : 1 : 0]$*.*

Conversely, if $(E, \mathcal{O})$ *is an elliptic curve defined over* K*, then there is an isomorphism* φ *from* E *onto a curve given by a Weierstraß equation with coefficients in* K*, and such that* $\varphi(\mathcal{O}) = [0 : 1 : 0]$*.*

[16] To remember these subscripts, notice that for each monomial $a_iX^jY^kZ^l$ of (3.2), we have $i + 2j + 3k = 6$.

Example 3.48 *The Fermat curve F_3 given by $A^3 + B^3 = C^3$ is an elliptic curve over the rational numbers $\mathbb{Q}$ with $\mathcal{O} = [1 : -1 : 0]$. It is isomorphic to the elliptic curve E given by the Weierstraß equation*

$$Y^2 Z = X^3 - 432 \cdot Z^3.$$

Indeed, the map

$$\begin{array}{cccc} \varphi : & F_3 & \longrightarrow & E \\ & [A : B : C] & \longmapsto & [12C : 36\,(A - B) : A + B] \end{array}$$

is easily seen to be an isomorphism with inverse

$$\begin{array}{cccc} \psi : & E & \longrightarrow & F_3 \\ & [X : Y : Z] & \longmapsto & [36Z + Y : 36Z - Y : 6X]\,. \end{array}$$

Surprisingly, this seemingly innocent observation can be used to prove the following special case of Fermat's Last Theorem[17]*:*

If $A, B, C \in \mathbb{Z}$ satisfy $A^3 + B^3 = C^3$, then $A \cdot B \cdot C = 0$.

This statement was already conjectured circa 900 A.D. by Arab mathematicians, while the very first proof was (as far as we know) provided by Fermat himself.

As we will shortly see, Weierstraß equations are really convenient in practice since, among other things, the corresponding group law algorithm can be efficiently implemented. Nevertheless, other defining equations for elliptic curves are of cryptographic interest as well. For instance, the Hessian [JQ01] and Jacobi [LS01, BJ03] families[18] can be used as (one level of) protection against side-channel analysis. In a nutshell, an attacker monitors side-channel leakage (like running time, power consumption or electromagnetic (EM) emanations) during the execution of a crypto-algorithm in the hope of recovering secret data (a private key, perhaps). It is thus somehow flattering that such physical attacks could have pure algebro-geometric countermeasures [End of digression[19]!].

From a given Weierstraß equation, it is possible (but tedious) to expliciteły write down the conditions that will ensure the smoothness of the curve directly from Definition 3.34. It is then convenient to define the following quantity before we state the smoothness condition.

Definition 3.49 *The* discriminant *of the Weierstraß equation*

$$Y^2 Z + a_1 XYZ + a_3 YZ^2 = X^3 + a_2 X^2 Z + a_4 XZ^2 + a_6 Z^3.$$

[17] See [Was03, p.36] for details.

[18] See [Hus86, Chapter 4] and [Con99, Chapter 1] for a general description of these families.

[19] For further details regarding this fascinating duel between cryptanalysts and cryptographers, see [BSS05, Chapter IV, V].

is the quantity $\Delta = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9b_2 b_4 b_6$*, where*

$$\begin{aligned} b_2 &= a_1^2 + 4a_2, \\ b_4 &= 2a_4 + a_1 a_3, \\ b_6 &= a_3^2 + 4a_6, \text{ and} \\ b_8 &= a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2. \end{aligned} \tag{3.4}$$

Lemma 3.50 *Let* E *be an elliptic curve given by* $Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$*. Then,* E *is smooth if and only if* $\Delta \neq 0$*.*

The proof of the above result can be found in [Sil86, Prop. III.1.4(a)]. Thus, one way to generate a smooth elliptic curve would be to randomly pick the a_i's until $\Delta \neq 0$. The above equations (as well as the group law algorithm) can however be simplified depending on the characteristic of K and on the j-invariant of E.

Definition 3.52 *Let* E *be a smooth elliptic curve given by the Weierstraß equation* $Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$*. The* j-invariant *of* E *is defined as*

$$j_E = \frac{\left(b_2^2 - 24b_4\right)^3}{\Delta},$$

where Δ *is the discriminant of the Weierstraß equation and* b_2*,* b_4 *are as in (3.4).*

So let E be a smooth elliptic curve defined over a field K. Let's first consider the case $\text{Char}\,(K) = 2$. Using a linear change of variables[20], it follows that E is isomorphic to a curve given by a Weierstraß equation (with coefficients in K) of the form:

$$\begin{cases} y^2 + xy = x^3 + ax^2 + b & \text{if } j_E \neq 0, \\ y^2 + cy = x^3 + dx + e & \text{if } j_E = 0. \end{cases}$$

The discriminants of these two equations respectively equal b and c^4 (and thus, $c, b \neq 0$).

Now, if $\text{Char}\,(K) \neq 2, 3$, then one can also use a linear change of variables[21] in order to get a curve given by

$$y^2 = x^3 + ax + b, \tag{3.5}$$

where $a, b \in K$ and which will be isomorphic to E. Furthermore, the discriminant of (3.5) equals $-16\left(4a^3 + 27b^2\right)$.

[20] See [Men93, Section 2.5] for explicit formulæ.

[21] See [Sil86, Section III.1] or [Men93, Section 2.4].

The Group Law

In 1835, Carl Gustav Jakob Jacobi (Jacobi *pour les intimes*) had the wonderful idea of considering a group law on cubic curves [Jac35] . Exactly 150 years elapsed before Miller and Koblitz independently put forward the use of these groups in cryptography [Mil86b, Kob87].

Interestingly enough, the Vigenère cipher, also called *Le Chiffre Indéchiffrable*[22], was still considered secure in 1835. In fact, we had to wait until 1854, three years after the death of Jacobi, before the British scientist Charles Babbage[23] finally found the weakness that allowed to break it[24].

So who knows, maybe a century from now, we will still find new applications (that are currently beyond our wildest dreams) to mathematical ideas born in Y2K...

We now proceed to describe Jacobi's idea, the so-called *chord-and-tangent rule* on a smooth elliptic curve E given by the Weierstraß equation

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3, \tag{3.6}$$

with coefficients in $\overline{K}$ and distinguised point $\mathcal{O} = [0 : 1 : 0]$. We will here follow the advice of Silverman and Tate, starting with a geometrical description of the method, which can then be easily translated into equations.

Just as with the Pell conic, the group law on the elliptic curve E can be described in terms of straight lines. So let P, $Q \in E$ be given. The point $P+Q \in E$ is obtained as follows. First draw the line $\ell_{P,Q}$ passing through P and Q. In the case where $P = Q$, $\ell_{P,P}$ simply is the tangent line to E at P, as depicted (over the reals) in Figure 3.5. Now consider the points of $\mathbb{P}^2$ which lies both on $\ell_{P,Q}$ and E. Since $\ell_{P,Q}$ has degree 1 and E has degree 3, Bézout's theorem[25] tells us that there are exactly 3 such points (counting intersection multiplicities). But we already know two of them: P and Q. Thus let R be the third such point of intersection (notice that it is possible that it coincides with P or Q). Now, draw the line $\ell_{R,\mathcal{O}}$ passing through R and $\mathcal{O}$. Apply Bézout's theorem once again to get that the intersection of E with $\ell_{R,\mathcal{O}}$ consists of precisely 3 points: R, $\mathcal{O}$, and S, say. We then define the sum of P and Q to be equal to S, as illustrated in Figure 3.5.

Remark 3.53 *Since $P + Q = S$, we therefore have that $(P + Q) + R = S + R$. So let's now evaluate $S + R$. By construction, the line $\ell_{R,S}$ intersects E at R, S and $\mathcal{O}$. Now, the* line at

[22] That is, '*the unbreakable cipher*'.

[23] Mr. Babbage also devised the blueprint of the modern computer, invented the speedometer, was the first to realize that the year's weather influenced the width of a tree ring, and much, much more [Sin99, Chapter 2].

[24] The historical perspective of this quest is revealed in [Sin99, Chapter 2].

[25] See [Ful69, Section 5.3] or [Har77, Corollary I.7.8].

infinity $\ell_{\mathcal{O},\mathcal{O}}$ *given by* $Z = 0$ *intersects* E *at* $\mathcal{O}$ *with multiplicity 3 since equation (3.6) reduces to* $X^3 = 0$ *when* $Z = 0$. *Thus,* $S + R = \mathcal{O}$, *and so* $(P + Q) + R = \mathcal{O}$. *As a result:*

$$\textit{The three points of intersection of a line with } E \textit{ sum up to } \mathcal{O}. \tag{3.7}$$

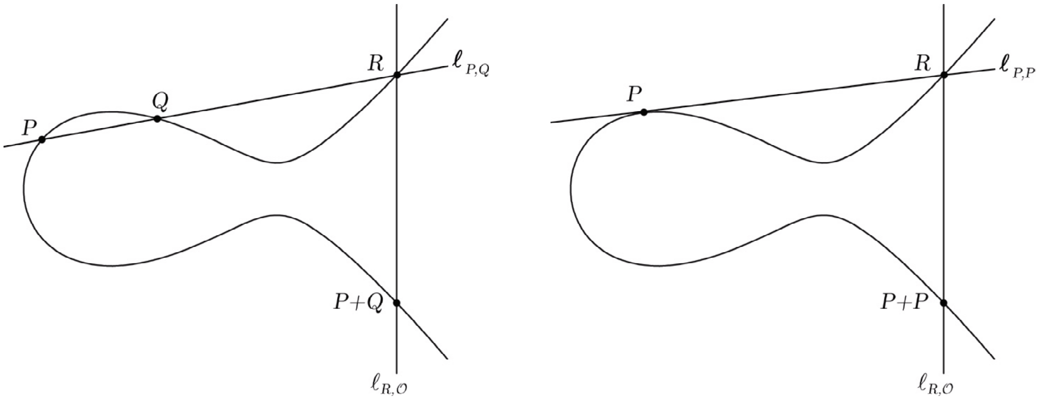


Figure 3.5: Chord-and-tangent rule on an elliptic curve

If there was a Hall of Fame of the most popular figures inspired by number theory, the chord-and-tangent rule would be there for sure: from T-shirt designs[26] to book covers [ST92, Was03], it is omnipresent. But beyond this pretty picture, a powerful and deep theory is hidden.

But we are not there yet. Indeed, from the description of the chord-and-tangent rule, it is not even obvious that this in fact defines a group law. In particular, everyone who ever tried to visualize the associativity property (which involves 8 straight lines) or work out by hand the details using the corresponding equations[27] realizes (after a few pages) just how tedious it may become. However, if we are willing to wait until we introduce divisors in Section 3.3, then it will suddendly appear so clear why it actually forms an abelian group. But for now, let's just state the desired result [Sil86, Prop. III.2.2].

Theorem 3.54 *Let* E *be a smooth elliptic curve given by a Weierstraß equation with coefficients in* $\overline{K}$ *and distinguised point* $\mathcal{O} = [0 : 1 : 0]$. *Then,* E *with the chord-and-tangent rule as binary operation forms an abelian group with identity* $\mathcal{O} = [0 : 1 : 0]$. *Furthermore, if* E *is defined over* K, *then* $E(K)$ *is a subgroup of* E.

We now proceed to obtain easy to implement explicit formulæ for this group operation. Let's

[26] Curious? See `http://www.crm.umontreal.ca/act/theme/theme_1998-1999_fr.html`.

[27] See Section 2.4 of [Was03] for full details (12 pages).

treat the easy cases first. Since $\mathcal{O}$ is the identity element, we know that $\mathcal{O}+\mathcal{O}=\mathcal{O}$, $P+\mathcal{O}=P$ and $\mathcal{O}+P=P$ for any $P\in E$.

In the case where P, $Q\neq\mathcal{O}$, we can assume without loss of generality that $P=[X_P:Y_P:1]$ and $Q=[X_Q:Y_Q:1]$. The line $\ell_{P,Q}$ is given by an equation of the form

$$aX+bY+cZ=0, \tag{3.8}$$

with a, b, c not all zero. We now want to get the coordinates of the third point R that simultaneously satisfy (3.6) and (3.8).

First, $\mathcal{O}=[0:1:0]$ is the only point on E with $Z=0$, so we can start by checking whether it satisfies (3.8) or not (notice that this will be the case if and only if $b=0$). If so, then $R=\mathcal{O}$, which implies by (3.7) that $P+Q=\mathcal{O}$.

Otherwise, $b\neq 0$ and we can assume without loss of generality that $R=[X_R:Y_R:1]$. When $Z=1$, equations (3.6) and (3.8) become

$$\begin{cases} Y^2+a_1XY+a_3Y=X^3+a_2X^2+a_4X+a_6, \\ Y=\mathbf{m}X+\mathbf{b}, \end{cases} \tag{3.9}$$

where $\mathbf{m}=-a/b$ and $\mathbf{b}=-c/b$ are respectively the slope and Y-intercept of $\ell_{P,Q}$ and

$$\begin{cases} a=Y_Q-Y_P \\ b=X_P-X_Q \\ c=X_QY_P-X_PY_Q \end{cases} \text{if } P\neq Q \text{ and } \begin{cases} a=3X_P^2+2a_2X_P+a_4-a_1Y_P \\ b=-\left(2Y_P+a_1X_P+a_3\right) \\ c=-X_P^3+a_4X_P+2a_6-a_3Y_P \end{cases} \text{if } P=Q.$$

It is a high school exercise to get these values of a, b and c when $P\neq Q$, while implicit differentiation of $(Y+a_1X+a_3)\,Y=X^3+a_2X^2+a_4X+a_6$ is used to get the slope of the tangent line to E at P: from $(Y'+a_1)\,Y+(Y+a_1X+a_3)\,Y'=3X^2+2a_2X+a_4$, we get that

$$(2Y+a_1X+a_3)\,Y'=3X^2+2a_2X+a_4-a_1Y.$$

Now, substituting $Y=\mathbf{m}X+\mathbf{b}$ in (3.3) yields a cubic equation in X looking like this:

$$X^3+\left(a_2-a_1\mathbf{m}-\mathbf{m}^2\right)X^2+\left(a_4-2\mathbf{mb}-a_1\mathbf{b}-a_3\mathbf{m}\right)X+\left(a_6-\mathbf{b}^2-a_3\mathbf{b}\right)=0. \tag{3.10}$$

Since X_P, X_Q and X_R all satisfy this equation, the left-hand side of (3.10) must equal $(X-X_P)\cdot(X-X_Q)\cdot(X-X_R)$ and can thus be rewritten as

$$X^3-(X_P+X_Q+X_R)\,X^2+(X_PX_Q+X_PX_R+X_QX_R)\,X-X_PX_QX_R. \tag{3.11}$$

Equating the coefficient of X^2 in (3.10) and (3.11) yields

$$X_R=\mathbf{m}^2+a_1\mathbf{m}-a_2-X_P-X_Q \text{ and } Y_R=\mathbf{m}X_R+\mathbf{b}.$$

It only remains to find the coordinates of $S = [X_S : Y_S : Z_S]$. First notice that $S \neq \mathcal{O}$ since otherwise,

$$\mathcal{O} = (P + Q) + R = S + R = \mathcal{O} + R = R$$

by Remark 3.53. So without loss of generality, $Z_S = 1$. Since the equation of the line $\ell_{R,\mathcal{O}}$ is $X - X_R Z = 0$, we therefore have $X_S = X_R$. Moreover,

$$Y_R^2 + a_1 X_R Y_R + a_3 Y_R = X_R^3 + a_2 X_R^2 + a_4 X_R + a_6 = X_S^3 + a_2 X_S^2 + a_4 X_S + a_6 = Y_S^2 + a_1 X_S Y_S + a_3 Y_S$$

since both R and S are on E. As a result,

$$\left(Y_R + Y_S + a_1 X_R + a_3\right)\left(Y_R - Y_S\right) = 0,$$

and thus

$$Y_S = -\left(Y_R + a_1 X_R + a_3\right) = -\left(\mathbf{m} + a_1\right) X_R - \mathbf{b} - a_3.$$

In accordance with Theorem 3.54, remark that if E is defined over K and P, $Q \in E(K)$, then the above explicit formulæ show that $S \in E(K)$ as well.

Lastly, we derive equations for the (additive) inverse of $P = [X_P : Y_P : 1]$. We are thus looking for a point $P' = [X_{P'} : Y_{P'} : 1]$ such that $P + P' = \mathcal{O}$. From Remark 3.53, the third point of intersection of $\ell_{P,P'}$ with E is then $\mathcal{O}$. The equation of $\ell_{P,P'}$ is therefore given by $X - X_P Z = 0$, which implies that $X_{P'} = X_P$ and $Y_{P'} = -\left(Y_P + a_1 X_P + a_3\right)$. And here again, if E is defined over K and $P \in E(K)$, notice that $P' \in E(K)$ as well.

For future reference, let's compactly summarize these results.

Theorem 3.55 *Let E be a smooth elliptic curve given by the Weierstraß equation*

$$Y^2 Z + a_1 XYZ + a_3 YZ^2 = X^3 + a_2 X^2 Z + a_4 XZ^2 + a_6 Z^3$$

with coefficients in $\overline{K}$ and distinguished point $\mathcal{O} = [0 : 1 : 0]$. Let $P = [X_P : Y_P : 1], Q = [X_P : Y_P : 1] \in E$ be given. Then, the inverse of P is

$$-P = \left[X_P : -Y_P - a_1 X_P - a_3 : 1\right].$$

So if $Q = -P$, then $P + Q = \mathcal{O}$. Otherwise, $P + Q = [X_S : Y_S : 1]$, where

$$X_S = \mathbf{m}^2 + a_1 \mathbf{m} - a_2 - X_P - X_Q,\ Y_S = -\left(\mathbf{m} + a_1\right) X_S - \mathbf{b} - a_3$$

and

$$\mathbf{m} = \frac{Y_Q - Y_P}{X_Q - X_P} \text{ and } \mathbf{b} = \frac{X_Q Y_P - X_P Y_Q}{X_Q - X_P} \text{ if } P \neq Q, \text{ and}$$
$$\mathbf{m} = \frac{3X_P^2 + 2a_2 X_P + a_4 - a_1 Y_P}{2Y_P + a_1 X_P + a_3} \text{ and } \mathbf{b} = \frac{-X_P^3 + a_4 X_P + 2a_6 - a_3 Y_P}{2Y_P + a_1 X_P + a_3} \text{ when } P = Q.$$

Remark 3.56 *We have here expressed the points to add in the form $[X : Y : 1]$, the so-called* affine coordinates*. Take note that various other representations for the points of E are also possible, like the homogeneous projective coordinates, Jacobian coordinates, Chudnovsky Jacobian coordinates, modified Jacobian coordinates, mixed coordinates, etc. The choice of a coordinate system for a specific implementation will depend on several factors, like the relative cost of a finite field inversion to that of a multiplication. For a detailed account of these coordinate systems, we refer to [CF05, Section 13.2-13.3].*

Before we go any further, let's see some of the basic built-in MAGMA commands for elliptic curves.

Example 3.57 *As shown in the following self-explanatory example, it is really easy to work with elliptic curves in* MAGMA*. In fact, this is the software we used for our prototype implementation of generalized Jacobians of Chapter 5.*

```
> K:=GF(7);
> E:=EllipticCurve([K|1,4]);
> E;                                 // Gives the details on the elliptic curve
Elliptic Curve defined by y^2 = x^3 + x + 4 over GF(7)
> Discriminant(E);
3
> #E;                                               // Number of points in E(K)
10
> Points(E);                                        // Lists the points of E(K)
{@ (0 : 1 : 0), (0 : 2 : 1), (0 : 5 : 1), (2 : 0 : 1), (4 : 3 : 1),
   (4 : 4 : 1), (5 : 1 : 1), (5 : 6 : 1), (6 : 3 : 1), (6 : 4 : 1) @}
> IsCyclic(AbelianGroup(E));       // Outputs ‘true’ iff E(K) is a cyclic group
true
> E!0;                                                     // Point at infinity
(0 : 1 : 0)
> P:=E![5,1,1];                    // Sets P equal to the point [5:1:1] in E(K)
> P[1];                                         // Outputs the X-coordinate of P
5
> P+P;
(6 : 3 : 1)
> 2*P;
(6 : 3 : 1)
> -P;                                               // Computes the inverse of P
(5 : 6 : 1)
> Order(P);                                           // Computes the order of P
10
> for i in [1..Order(P)] do                       // Computes the multiples of P
```

```
for> print(i*P);
for> end for;
(5 : 1 : 1)
(6 : 3 : 1)
(0 : 2 : 1)
(4 : 3 : 1)
(2 : 0 : 1)
(4 : 4 : 1)
(0 : 5 : 1)
(6 : 4 : 1)
(5 : 6 : 1)
(0 : 1 : 0)
> Q:=Random(E);                  // Q is a pseudo-randomly chosen point in E(K)
> Q;
(4 : 4 : 1)
> Log(P,Q);                     // Computes the discrete log of Q to the base P
6
> 6*P eq Q;                             // Checks the correctness of the answer
true
```

It is therefore a child's play to explore discrete logarithms with the help of MAGMA.

So we now know that an elliptic curve naturally possesses an abelian group structure. Is that all one can say? In fact, we can also emphasize that the formulæ used to compute inverses and sums of points really are *'nice'* functions. This idea can be formalized as follows.

Definition 3.58 *Let A be a nonsingular projective variety. Suppose that A is also an abelian group with identity $\mathcal{O} \in A$ and that the addition law $\oplus : A \times A \to A$ and inverse map $\ominus : A \to A$ are morphisms. Then, $(A, \mathcal{O}, \oplus, \ominus)$ is said to be an* abelian variety*.*

Remark 3.59 *In most cases, we simply say 'A is an abelian variety' when the underlying group structure is understood.*

The following result establishes the fundamental equivalence between nonsingular elliptic curves and abelian varieties of dimension one. The proof of this result can be found in the excellent article *An elementary proof that elliptic curves are abelian varieties* of Loren D. Olson [Ols73, Theorem 9 and Corollary 11].

Theorem 3.60 *A nonsingular elliptic curve is an abelian variety of dimension one. Conversely, an abelian variety of dimension one is a nonsingular elliptic curve.*

There are various techniques that one can use to check that the explicit equations of Theorem 3.55 indeed define morphisms. One way to proceed is to use the so-called *'translation maps'*, as used by Silverman in the proof of [Sil86, Theorem III.3.6]. A less elegant (but equally informative) approach is to roll up our sleeves and play with explicit equations, as outlined in [Sil86, Remark III.3.6.1].

Indeed, this direct technique allows us to modify the classical equations of the group law in order to get *'unified point addition formulæ'*. Informally, a unified formula enjoys the property that the corresponding group law algorithm for computing $P+Q$ does not contain conditional statements that treats the case $P=Q$ separately. Unified formulæ are therefore interesting countermeasures against side-channel attacks. Recently, Éric Brier, Marc Joye and the author proposed the following family of unified formulæ [rBDJ]:

Theorem 3.61 *Let E be a smooth elliptic curve given by the Weierstraß equation*

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

with coefficients in a field K and let $P=[X_P : Y_P : 1]$, $Q=[X_Q : Y_Q : 1] \in E(K)$ be given such that $Q \neq -P$. Moreover, let $f \in K[X_1, Y_1; X_2, Y_2]$ be a given polynomial and define

$$\begin{aligned} g(X_1,Y_1;X_2,Y_2) &= X_1^2 + X_1X_2 + X_2^2 - a_1Y_1 + a_2(X_1+X_2) + a_4 + (Y_1 - Y_2)f(X_1,Y_1;X_2,Y_2), \\ h(X_1,Y_1;X_2,Y_2) &= Y_1 + Y_2 + a_1X_2 + a_3 + (X_1 - X_2)f(X_1,Y_1;X_2,Y_2). \end{aligned}$$

If $h(X_P,Y_P;X_Q,Y_Q)$ or $h(X_Q,Y_Q;X_P,Y_P)$ is nonzero, then

$$\mathbf{m} = \begin{cases} \dfrac{g(X_P,Y_P;X_Q,Y_Q)}{h(X_P,Y_P;X_Q,Y_Q)} & \text{if } h(X_P,Y_P;X_Q,Y_Q) \neq 0, \\ \dfrac{g(X_Q,Y_Q;X_P,Y_P)}{h(X_Q,Y_Q;X_P,Y_P)} & \text{if } h(X_Q,Y_Q;X_P,Y_P) \neq 0 \end{cases} \tag{3.12}$$

is well-defined and $P+Q = [X_{P+Q} : Y_{P+Q} : 1]$, where

$$X_{P+Q} = \mathbf{m}^2 + a_1\mathbf{m} - a_2 - X_P - X_Q \text{ and } Y_{P+Q} = (X_P - X_{P+Q})\,\mathbf{m} - Y_P - a_1X_{P+Q} - a_3. \tag{3.13}$$

Moreover, the following condition is sufficient for f to be defined:

$$\text{If } f(X_P,Y_P;X_Q,Y_Q) + f(X_Q,Y_Q;X_P,Y_P) = a_1, \text{ then } X_P = X_Q. \tag{3.14}$$

In fact, there are infinitely many f satisfying (3.14) for all P, $Q \in E(K)$, $Q \neq -P$.

Example 3.62 *If* $\mathrm{Char}(K) \neq 2,3$*, then we saw that E can be taken to have an equation of the form $Y^2Z = X^3 + aXZ^2 + bZ^3$. Thus, $f=1$ satisfies (3.14) for all P, $Q \in E(K)$ such that*

$Q \neq -P$ and the corresponding value of $\mathbf{m}$ is given by

$$\mathbf{m} = \begin{cases} \dfrac{(X_P + X_Q)^2 - X_P X_Q + Y_P - Y_Q + a}{Y_P + Y_Q + X_P - X_Q} & \textit{if } Y_P + Y_Q \neq X_Q - X_P, \\ \dfrac{(X_P + X_Q)^2 - X_P X_Q - Y_P + Y_Q + a}{Y_P + Y_Q - X_P + X_Q} & \textit{otherwise.} \end{cases}$$

Alternatively, a fresh value of f could also be chosen each time two points are added. As a result, the side-channel information leaking when computing $P + Q$ would also depend on the choice of f. For further details on the choice of f, we refer to [rBDJ].

Hasse, Deuring, Schoof, and Friends

If we want to use an elliptic curve E defined over a finite field $\mathbb{F}_q$ for cryptographic applications, we first would like to know how many bits q should have in order for $\#E(\mathbb{F}_q)$ to be of a convenient size. That is, large enough to counter attacks based on generic algorithms for the DLP (like the Pollard Rho method), and at the same time not disproportionately large since we want the computations to be performed efficiently.

In 1921, Emil Artin conjectured in his thesis [Art21] that $\#E(\mathbb{F}_q)$ should be of the order of magnitude of $q + 1$ and could never be less than $q + 1 - 2\sqrt{q}$ nor greater than $q + 1 + 2\sqrt{q}$. His intuition was indeed right, and a decade later, Helmut Hasse [Has33] was able to provide a formal proof[28] [Sil86, Theorem V.1.1].

Theorem 3.63 (Hasse) *Let E be an elliptic curve defined over $\mathbb{F}_q$. Then,*

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

But even if q is chosen to have, say, 160 bits and we are guaranteed that $E(\mathbb{F}_q)$ has minimal cardinality $q + 1 - 2\sqrt{q}$, we still need to make sure that $\#E(\mathbb{F}_q)$ has at least one large prime factor to resist the Pohlig-Hellman attack.

In fact, there is a deterministic polynomial-time algorithm[29], initially due to René Schoof, to compute the exact value of $\#E(\mathbb{F}_q)$ [Sch85]. Subsequently, Neal Koblitz specifically treated the case of characteristic two [Kob90]. In short, over the past twenty years, Schoof's idea was improved by several authors, including the work of Elkies and Atkin. The resulting method is now often referred to as the *Schoof-Elkies-Atkin* or *SEA algorithm* [30]. For an up to date account

[28] This result was generalized by André Weil in 1948 for curves of higher genus. See [Wei48] for the original exposition.

[29] The original algorithm was described for $\mathrm{Char}(\mathbb{F}_q) \neq 2, 3$ and required $O(\log^9 q)$ bit operations.

[30] One can easily get a sense of the efficiency of this algorithm using MAGMA since it already contains an implementation of this method.

of the point counting techniques for elliptic and hyperelliptic curves, we refer to Chapter 17 of [CF05].

Thus say we first fix the value of q and then randomly choose an elliptic curve E over $\mathbb{F}_q$ until $\#E(\mathbb{F}_q)$ is of the form $h \cdot l$, where l is a large prime and h is a small integer[31] called the *cofactor*. These are indeed requirements that are found in the standards for elliptic curve cryptography [IEE99, NIoST00, CR00]. It is in fact a highly nontrivial task to show that this simple method is a relatively efficient procedure to generate such curves. The proof relies in part on a result of Max Deuring [Deu41] that establishes the close connection between Kronecker class numbers and the problem of counting, up to isomorphism, the number $N_{q,n}$ of elliptic curves E over $\mathbb{F}_q$ such that $\#E(\mathbb{F}_q) = n$, where n is a given integer in the Hasse interval $\left(q+1-2\sqrt{q}, q+1+2\sqrt{q}\right)$. Very informally, Deuring's result implies that given $n = q + 1 + t$, where $t \in \mathbb{Z}$ is such that $|t| \leq 2\sqrt{q}$, we have that $N_{q,n}$ is roughly equal to $\sqrt{4q - t^2} \Big/ \pi$, which is represented graphically in Figure 3.6 [Sch04]. A concise account of Deuring's result can also be found in [Len87].

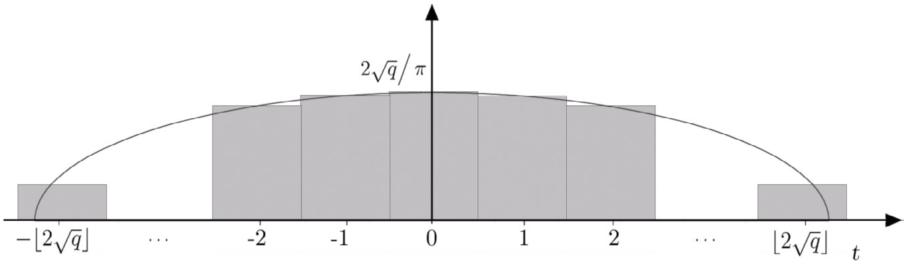


Figure 3.6: Visual interpretation of Deuring's result

Lastly, let's briefly mention that it is also sometimes possible to first choose the exact number $h \cdot l$ of points we want and then construct an elliptic curve matching this requirement. This is the so-called *complex multiplication, or CM method*, for which a description can be found in [CF05, Chapter 18].

3.2.3 Hyperelliptic Curves

The sole objective of this (outrageously) short section is to provide a motivation for the study of divisors, Picard groups and Jacobians that will come next. Indeed, hyperelliptic curves provide the perfect example of a family of curves of cryptographic interest where the group law is not directly defined on the set of points of the curve.

We want to emphasize that we will thus simply touch upon the topic of hyperelliptic curve

[31] Usually, we choose $h = 1, 2, 3$ or 4.

cryptography (HECC), and consequently, that our treatment will unfortunately not do justice nor reflect the true value of using these curves for cryptographic purposes. We thus do not want to leave the impression that hyperelliptic curves are not relevant in cryptography, as the truth is quite the opposite! For all details[32], the interested reader is urged to refer to the excellent *Handbook of Elliptic and Hyperelliptic Curve Cryptography* [CF05].

In the crypto community, the term *'hyperelliptic curves'* often refers to *imaginary quadratic hyperelliptic curves*, and we shall follow this convention as well. For simplicity, we also present the equation defining the curve in affine form, in accordance with most of the litterature of HECC.

Definition 3.64 *A* hyperelliptic curve of genus g *over a field* K *is an algebraic curve* C *given by an equation of the form*

$$y^2 + h\left(x\right)y = f\left(x\right),$$

where $h,\ f \in K\left[x\right]$*,* $\deg\left(f\right) = 2g+1$*,* $\deg\left(h\right) \leq g$*, and* f *is a monic polynomial.*

To ensure that C is smooth, it suffices to verify that the partial derivatives $2y+h$ and $f'-h'y$ do not simultaneously vanish at any point of $C\left(\overline{K}\right)$.

Remark 3.65 *Thus, notice that an elliptic curve can also be seen as a hyperelliptic curve of genus one.*

In order to provide a visual aid for this definition, Figure 3.7 presents an example of the graph of a hyperelliptic curve of genus two over the reals.

From this graph, it is at first really tempting to try to use ad hoc methods in the hope of defining the equivalent of the chord-and-tangent rule for elliptic curves. However, the set of points on a hyperelliptic curve of genus $g \geq 2$ per se do *not* form a group. But all is not lost since we can still use this set of points to turn it into a group. The clever way to proceed is to consider the divisors on C in order to build the so-called *Picard group,* $\mathrm{Pic}^0\left(C\right)$, of the curve. In turn, the *Jacobian* $J(C)$ of the curve will be a certain abelian variety isomorphic (as abelian groups) to $\mathrm{Pic}^0\left(C\right)$. So in a nutshell, the Jacobian of a hyperelliptic curve is the group we are using to do discrete logarithm-based cryptogaphy.

We conclude this section by providing an *avant-goût* of what the group law on the Jacobian look like. Figure 3.8 represents a hyperelliptic curve of genus two over the reals. With the notation for divisors that will be introduced at the beginning of next section, we have in this

[32] The 808 pages of this work is a truly complete account of the state-of-the-art in curve cryptography.

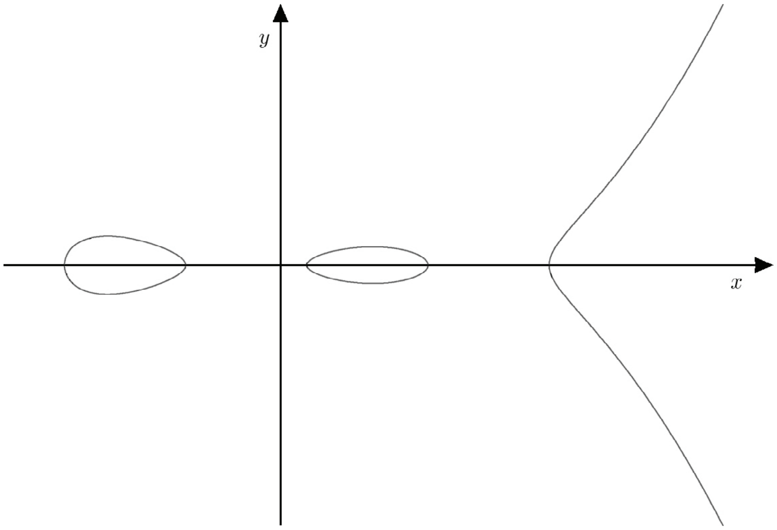


Figure 3.7: An example of a hyperelliptic curve of genus 2 over the reals

example that

$$(P_1) + (P_2) - 2\,(\mathcal{O}) \quad + \quad (Q_1) + (Q_2) - 2\,(\mathcal{O}) \quad = \quad (R_1) + (R_2) - 2\,(\mathcal{O})\,.$$

3.3 Divisors

As outlined in the previous section in the case of hyperelliptic curves, divisors will be the tool we need to turn a *set* into a *group*. Roughly speaking, and just to give an idea, let's just say for now that a divisor is a concise and convenient way of keeping track of the zeros and poles of functions.

3.3.1 Basic Concepts

So we are now ready to describe how we can create a group out of a set of elements. The starting point will be to consider a *free abelian group*. This process is in fact very natural, as demonstrated in the following high school level example (which can easily be omitted by anyone familiar with free abelian groups).

Example 3.66 *A stamp collector takes his passion quite seriously. To each collectible corresponds a unique identification code. It is then an easy matter to write in a compact form an up*

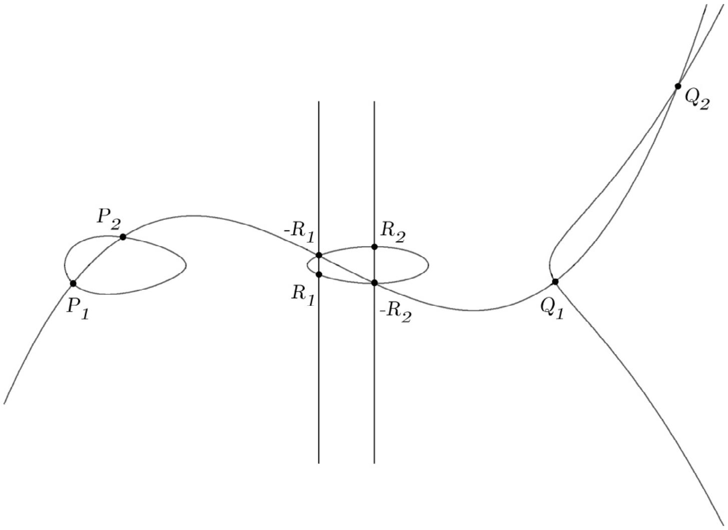


Figure 3.8: Visual interpretation of the group law on a hyperelliptic curve of genus two

to date inventory of his collection. At a glance, he can easily see what to buy and what can be traded as well as updating the state of his collection after each transaction.

Date	**Operation**	C_{5732}	E_{176}	F_{2098}	S_{54}	...
Oct. 9	Inventory	4	0	5	1	...
Oct. 10	Transaction	0	0	-2	1	...
Oct. 10	Inventory	4	0	3	2	...

For quick reference, this last state could also be symbolized by the shorthand $4(C_{5732})+3(F_{2098})+2(S_{54})+....$ *So we started with a* set *consisting of the different stamps and we ended up with a* group *where a typical element consists of a list of integers, one for each stamp.*

Formally, let S be a set (not necessarily finite) and let G be the collection of formal sums of the form

$$\sum_{A\in S} n_A(A)$$

where each n_A is an integer and finitely many of them are nonzero. The natural addition rule

$$\sum_{A\in S} m_A(A) + \sum_{A\in S} n_A(A) = \sum_{A\in S}(m_A + n_A)(A)$$

turns G into a group with identity $\sum_{A\in S} 0(A)$, denoted $\mathbf{0}$ (notice the difference between 0 and $\mathbf{0}$). The group G is called the *free abelian group* on S.

Now, let C be our favorite algebraic curve, defined over a perfect field K, for which we are collecting the points as a hobby. We then want to consider formal sums of the form

$$\sum_{P \in C} n_P(P),$$

where each n_P is an integer and finitely many of them are nonzero. Call such a sum a *divisor* on C. The free abelian group generated by the points of C is called the *divisor group* of C and is denoted by $\mathrm{Div}(C)$.

The *degree* of a divisor D is the integer

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

which is a *finite* sum of integers. The *divisors of degree zero* form a subgroup of $\mathrm{Div}(C)$, which we denote by

$$\mathrm{Div}^0(C) = \{\, D \in \mathrm{Div}(C) |\, \deg(D) = 0\} .$$

The *support* of D is defined as the (finite) set of points P such that n_P is nonzero:

$$\mathrm{Supp}\,(D) = \{\, P \in C |\, n_P \neq 0\} .$$

We say that D is *prime* to D' if D and D' have disjoint supports. Furthermore, D is called an *effective* (or *positive*) divisor when all $n_P \geqslant 0$. Lastly, we will write $D \geq D'$ when $D - D'$ is an effective divisor.

Now, let σ in $\mathrm{Gal}(\overline{K}/K)$, the Galois group of $\overline{K}$ over K, be given. Then, for any point $P = [x_0 : \ldots : x_n]$, we let $P^\sigma = [x_0^\sigma : \ldots : x_n^\sigma]$. If $D \in \mathrm{Div}(C)$, we also define

$$D^\sigma = \sum_{P \in C} n_P(P^\sigma).$$

Lastly, a divisor D is said to be *rational over* K (or *defined over* K) if $D^\sigma = D$ for all σ in $\mathrm{Gal}(\overline{K}/K)$. The *group of divisors defined over* K will be denoted $\mathrm{Div}_K\,(C)$. Similarly, $\mathrm{Div}_K^0(C)$ is the *group of degree zero divisors defined over* K.

3.3.2 Discrete Valuations

We will now build a discrete valuation on the function field $\overline{K}(C)$. Before we do so, we first recall some concepts.

Definition 3.67 *A* discrete valuation *(also called an* order function*) on a field F is a surjective map $v : F \to \mathbb{Z} \cup \{\infty\}$ such that:*

$$\begin{array}{rl} (i) & v(a) = \infty \text{ if and only if } a = 0. \\ (ii) & v(a \cdot b) = v(a) + v(b). \\ (iii) & v(a + b) \geq \min\left(v(a), v(b)\right). \end{array}$$

The valuation ring *of F is $R = \{a \in F | v(a) \geq 0\}$. Lastly, the ring R is called a* discrete valuation ring *(abbreviated* DVR*).*

Now to each smooth point of C, we will associate a discrete valuation, ord_P, that will basically tell us whether a function $f \in \overline{K}(C)$ has a zero or a pole at P, and if so, will also give the multiplicity.

Proposition 3.68 *Let C be an algebraic curve defined over K and let $P \in C$ be smooth point. Then the function*

$$\mathrm{ord}_P : \overline{K}(C) \longrightarrow \mathbb{Z} \cup \{\infty\},$$

which maps 0 to ∞, and $f \neq 0$ to its order of vanishing at P, *is a* discrete valuation. *Namely,*

$$\begin{aligned} \text{if } \mathrm{ord}_P\left(f\right) < 0, &\text{ then } f \text{ has a pole of order } -\mathrm{ord}_P\left(f\right) \text{ at } P, \\ \text{if } \mathrm{ord}_P\left(f\right) = 0, &\text{ then } f \text{ is defined and nonzero at } P, \\ \text{if } \mathrm{ord}_P\left(f\right) > 0, &\text{ then } f \text{ has a zero of order } \mathrm{ord}_P\left(f\right) \text{ at } P. \end{aligned}$$

It therefore follows that the following two (intuitively clear) properties hold:

$$\mathrm{ord}_P(f \cdot g) = \mathrm{ord}_P(f) + \mathrm{ord}_P(g) \text{ and } \mathrm{ord}_P(f + g) \geq \min\left(\mathrm{ord}_P(f), \mathrm{ord}_P(g)\right).$$

For further details on these discrete valuations, please refer to [Sil86, Section II.1].

3.3.3 Principal Divisors

It will now be convenient to associate a divisor to each function $f \in \overline{K}(C)$. The idea is to *'make a list'* where each entry is a point $P \in C$ together with $\mathrm{ord}_P(f)$, the order of f at P. A convenient way to do so is to consider the formal sum $\sum_{P \in C} \mathrm{ord}_P(f)(P)$. The following proposition from [Sil86, Proposition II.1.2] then ensures that this indeed defines a divisor.

Proposition 3.69 *Let C be a smooth algebraic curve defined over K and $f \in \overline{K}(C)$ be given. Then, there are a finite number of points of C at which f has a zero or pole. Moreover, if f has no poles, then f is constant (that is, $f \in \overline{K}$).*

We can now formally define the divisor we associate to the function f.

Definition 3.70 *Let C be a smooth algebraic curve defined over K. The* divisor of a function $f \in \overline{K}(C)^*$ *is*

$$\operatorname{div}(f) = \sum_{P \in C} \operatorname{ord}_P(f)(P).$$

Definition 3.71 *A divisor $D \in \operatorname{Div}(C)$ is said to be* principal *if there is an $f \in \overline{K}(C)^*$ such that $D = \operatorname{div}(f)$.*

The following basic properties of principal divisors, from [Sil86, Proposition II.3.1], will prove to be truly useful throughout this dissertation.

Proposition 3.72 *Let C be a smooth algebraic curve defined over K and $f, g \in \overline{K}(C)^*$ be given. Then,*

(i) $\operatorname{div}(f) = \mathbf{0}$ *if and only if $f \in \overline{K}^*$.*
(ii) $\deg(\operatorname{div}(f)) = 0$*. That is, all principal divisors have degree zero.*
(iii) $\operatorname{div}(f \cdot g) = \operatorname{div}(f) + \operatorname{div}(g)$*.*
(iv) $\operatorname{div}\left(\frac{f}{g}\right) = \operatorname{div}(f) - \operatorname{div}(g)$*.*
(v) $\operatorname{div}(f^n) = n \cdot \operatorname{div}(f)$ *for all integers $n \geq 1$.*

Definition 3.73 *Let* $\operatorname{Princ}(C) = \{ D \in \operatorname{Div}(C) |\, D$ *is principal*$\}$ *denote the set of principal divisors on C.*

Remark 3.74 *The above proposition in fact also shows that* $\operatorname{Princ}(C)$ *is a subgroup of* $\operatorname{Div}^0(C)$*.*

Moreover, notice that given a principal divisor $D = \operatorname{div}(f)$, the function $f \in \overline{K}(C)^*$ is only determined up to multiplication by a nonzero element of $\overline{K}$. Indeed, if $g \in \overline{K}(C)^*$ is such that $D = \operatorname{div}(f) = \operatorname{div}(g)$, then

$$\mathbf{0} = D - D = \operatorname{div}(f) - \operatorname{div}(g) = \operatorname{div}\left(\frac{f}{g}\right),$$

from which follows that $f/g \in \overline{K}^*$. Thus, $f = c \cdot g$ for some $c \in \overline{K}^*$.

Example 3.75 *All divisors of degree zero on $\mathbb{P}^1$ are principal. Indeed, let $D = \sum n_P(P) \in$* $\operatorname{Div}\left(\mathbb{P}^1\right)$ *be given such that* $\deg(D) = 0$*. For each $P = [x_P : y_P] \in \mathbb{P}^1$, the function $y_P X - x_P Y$ will vanish at P only. Thus, $D = \operatorname{div}(f)$ where*

$$f = \prod_{P \in \mathbb{P}^1} (y_P X - x_P Y)^{n_p}.$$

The key observation here is to notice that we have $f \in K(\mathbb{P}^1)$ since $\deg(D) = 0$*.*

This simple example, as we will later see, is a key difference between elliptic curves and curves of genus zero, such as $\mathbb{P}^1$.

We now define an equivalence relation on divisors, which is the first step towards the construction of the Jacobian of a curve. In this chapter, we will only consider *linearly* equivalent divisors, as opposed to, say, *algebraically* or *numerically* equivalent divisors. An overview of these equivalence relations can be found in [Die85, Section VII.7]. In the next chapter, we will see how to modify the definition of linear equivalence in order to construct generalized Jacobians (c.f. Section 4.2).

Definition 3.76 *Let D_1, $D_2 \in \mathrm{Div}(C)$ be given. If $D_1 - D_2$ is a principal divisor, then we say that D_1 and D_2 are* linearly equivalent*, and we write $D_1 \sim D_2$.*

Lastly, given a divisor D and a function f, we formalize the idea of *'evaluating f at D'*.

Definition 3.77 *Let C be a smooth algebraic curve defined over K. Let $D = \sum_{P \in C} n_P(P) \in \mathrm{Div}(C)$ and $f \in \overline{K}(C)^*$ be given such that D and $\mathrm{div}(f)$ have disjoint supports. We then define*

$$f(D) = \prod_{P \in C} f(P)^{n_P} = \prod_{P \in \mathrm{Supp}(D)} f(P)^{n_P}.$$

Notice that this is a finite product since finitely n_P's are nonzero by definition.

3.3.4 The Riemann-Roch Theorem

Initially demonstrated as Riemann's inequality [Ful69, Section 8.3], the Riemann-Roch theorem has gained its current form following the work of Gustav Roch, himself a student of Riemann, during the 1850s. This much celebrated theorem is one of the most important tool in the algebraic geometry of curves.

To each divisor, we now associate a subset of $\overline{K}(C)$ as follows.

Definition 3.78 *Let $D \in \mathrm{Div}\,(C)$ be given. We let*

$$\mathcal{L}(D) = \left\{ f \in \overline{K}(C)^* \,\middle|\, \mathrm{div}\,(f) \geq -D \right\} \cup \{0\}\,.$$

The set $\mathcal{L}(D) \subseteq \overline{K}(C)$ is in fact a finite-dimensional $\overline{K}$-vector space [Sil86, Proposition II.5.2 (b)]. Its dimension will be denoted $l(D) = \dim_{\overline{K}} \mathcal{L}(D)$.

Technically, we would need to formally introduce canonical divisors (and thus differential forms[33]) in order to fully appreciate the scope of Riemann-Roch. However, the only result that

[33] Indeed, a divisor $K_C \in \mathrm{Div}\,(C)$ is said to be *canonical* if there is a nonzero differential form ω on C such that $K_C \sim \mathrm{div}\,(\omega)$.

we will need in this work is a corollary of this theorem that does not involve canonical divisors. We have then settled for stating the general result for completeness and to refer to Section II.4 of [Sil86] for details about differentials and canonical divisors.

Theorem 3.79 (Riemann-Roch) *Let C be a smooth algebraic curve and K_C be a canonical divisor on C. Then, there is a nonnegative integer g such that*

$$l\,(D) - l\,(K_C - D) = \deg\,(D) - g + 1$$

for all $D \in \operatorname{Div}(C)$. The integer g is called the genus *of C.*

The classical proof of Brill and Noether was reproduced by Fulton in [Ful69, Section 8.6]. As for the proof of the following corollary, see [Sil86, Corollary II.5.5(c)].

Corollary 3.80 *Let C be a smooth algebraic curve of genus g and let $D \in \operatorname{Div}(C)$ be given. If $\deg\,(D) > 2g - 2$, then $l\,(D) = \deg\,(D) - g + 1$.*

3.3.5 The Abel-Jacobi Theorem

We now see a very interesting application of the theory of divisors. At the same time, it will be a good occasion to get used to work with principal divisors, as this will be much needed to get to understand generalized Jacobians.

As pointed out by Jean-Pierre Serre, the Abel-Jacobi Theorem is of the utmost importance since *"The theory of the usual Jacobian has its source in the theorems of Abel and Jacobi"* [Ser88, p.108]. As a bonus, the proof of this result is quite enlightening, one part being a sequence of intuitive deductions, while the other demontrates the powerfulness of the Riemann-Roch theorem. We will thus take the time of going through this proof.

But before we do so, we will need to make a fundamental observation about the principal divisors on an elliptic curve. This simple exercise merely requires to play with secant and tangent lines.

Let E be a smooth elliptic curve defined over K and let P, $Q \in E$ be given. Let also R be the third point of intersection of E with the straight line $\ell_{P,Q}$ passing through P and Q. Finally, let $\ell_{P+Q,\mathcal{O}}$ be the line passing through $P+Q$ and $\mathcal{O}$. For quick reference, the chord-and-tangent rule has been reproduced in Figure 3.9.

Remark 3.81 *In the sequel, we will often abuse notation and identify $\ell_{P,Q}$ with both the* line *passing through P and Q and the* function *defining this line. From the context, it should however be clear as to which notion we are referring to.*

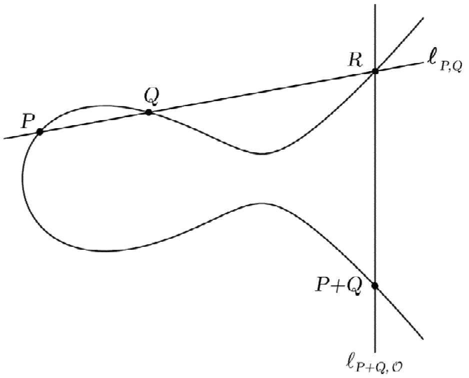


Figure 3.9: The chord-and-tangent rule and its interpretation in terms of divisors

Now, $\ell_{P,Q}$ will have zeros at P, Q and R only, which must leave a pole of order 3 at the point at infinity. In other words,

$$\operatorname{div}\left(\frac{\ell_{P,Q}}{Z}\right) = (P) + (Q) + (R) - 3(\mathcal{O}).$$

Similarly, we have that

$$\operatorname{div}\left(\frac{\ell_{P+Q,\mathcal{O}}}{Z}\right) = (R) + (P+Q) - 2(\mathcal{O}).$$

Thus,

$$\operatorname{div}\left(\frac{\ell_{P,Q}}{\ell_{P+Q,\mathcal{O}}}\right) = \operatorname{div}\left(\frac{\ell_{P,Q}}{Z}\right) - \operatorname{div}\left(\frac{\ell_{P+Q,\mathcal{O}}}{Z}\right) = (P) + (Q) - (P+Q) - (\mathcal{O}).$$

We have therefore shown:

Lemma 3.82 *Let E be a smooth elliptic curve defined over K and let P, $Q \in E$ be given. Then,*

$$\operatorname{div}\left(\frac{\ell_{P,Q}}{\ell_{P+Q,\mathcal{O}}}\right) = (P) + (Q) - (P+Q) - (\mathcal{O}), \tag{3.15}$$

where ℓ_{P_1,P_2} denotes the equation of the line passing through P_1 and P_2 (tangent at the curve if $P_1 = P_2$).

We can now repeatedly use this result as follows. In the case where $P = Q$, then the above identity reads as

$$\operatorname{div}(g_2) = 2(P) - (2P) - (\mathcal{O}), \text{ where } g_2 = \frac{\ell_{P,P}}{\ell_{P+P,\mathcal{O}}} \in \overline{K}(C)^*.$$

Next, $3P = P + 2P$, and so we can easily find a function $g_3 \in \overline{K}(C)^*$ such that

$$\operatorname{div}(g_3) = (P) + (2P) - (3P) - (\mathcal{O}).$$

As a result,

$$\operatorname{div}(g_2 \cdot g_3) = 3(P) - (3P) - 2(\mathcal{O}).$$

Likewise, $4P = P + 3P$ and we thus know a $g_4 \in \overline{K}(C)^*$ satisfying

$$\operatorname{div}(g_4) = (P) + (3P) - (4P) - (\mathcal{O}).$$

And this implies that

$$\operatorname{div}(g_2 \cdot g_3 \cdot g_4) = 4(P) - (4P) - 3(\mathcal{O}).$$

This recursive process shows that for any integer $k \geq 1$, there is a function $f_k \in \overline{K}(C)^*$ such that

$$\operatorname{div}(f_k) = k(P) - (kP) - (k-1)(\mathcal{O}).$$

Notice that when $k = 1$, we can simply let $f_1 = 1$.

Lemma 3.83 *Let E be a smooth elliptic curve defined over K and $P \in E$ be given. Then for any integer $k \geq 1$, there is a $f_k \in \overline{K}(C)^*$ such that*

$$\operatorname{div}(f_k) = k(P) - (kP) - (k-1)(\mathcal{O}). \tag{3.16}$$

So the moral of the story is that we can already see how it is possible to express many divisors of *degree zero* as a divisor of a function, simply by playing with straight lines.

Abel's Theorem

Given a smooth elliptic curve defined over K, we may then wonder:

To what extent can a divisor of degree zero be expressed as a divisor of a function?

So let's challenge ourselves and try to express an arbitrary divisor of degree zero as a divisor of a function. Thus, let

$$D = a_1(P_1) + a_2(P_2) + ... + a_m(P_m) - b_1(Q_1) - b_2(Q_2) - ... - b_n(Q_n) + c(\mathcal{O}) \in \operatorname{Div}^0(E) \tag{3.17}$$

be given such that $a_1, \ldots, a_m > 0$ and $b_1, \ldots, b_n > 0$. At this point, we simply want to know whether or not this can be achieved (and thus we won't take efficiency considerations into account here).

The obvious thing to do first is to use identity (3.16) for each P_i and Q_j. So we know that there are functions f_i and g_j satisfying

$$\begin{array}{l} a_i(P_i) = (a_iP_i) + (a_i - 1)\,(\mathcal{O}) + \operatorname{div}(f_i) \text{ for } 1 \leq i \leq m, \text{ and} \\ b_j(Q_j) = (b_jQ_j) + (b_j - 1)\,(\mathcal{O}) + \operatorname{div}(g_j) \text{ for } 1 \leq j \leq n. \end{array}$$

Substituting in (3.17) then yields

$$D = (a_1P_1) + ... + (a_mP_m) - (b_1Q_1) - ... - (b_nQ_n) + (n-m)\,(\mathcal{O}) + \operatorname{div}\left(\frac{f_1f_2...f_m}{g_1g_2...g_n}\right), \quad (3.18)$$

where we used the fact that $a_1 + \ldots + a_m - b_1 - \ldots - b_n = -c$ since D has degree zero. So we now have all the coefficients of the (a_iP_i) equal to one, and those of (b_1Q_1) equal to -1. Recall that our goal is to replace terms on the right hand side as much as possible by divisors of functions. An easy simplification is to read (3.15) as

$$(a_1P_1) + (a_2P_2) = (a_1P_1 + a_2P_2) + (\mathcal{O}) + \operatorname{div}(h_1)$$

for some $h_1 \in \overline{K}(C)^*$ and substitute in (3.18) to get

$$\begin{aligned} D \;\; = \;\; & (a_1P_1 + a_2P_2) + (a_3P_3) + ... + (a_mP_m) - (b_1Q_1) - (b_2Q_2) - ... - (b_nQ_n) \\ & + (n-m+1)\,(\mathcal{O}) + \operatorname{div}\left(\frac{f_1f_2...f_m}{g_1g_2...g_n} \cdot h_1\right). \end{aligned}$$

We can then repeat this process to decrease the number of terms in the right until we hit

$$D = (a_1P_1 + a_2P_2 + ... + a_mP_m) - (b_1Q_1 + b_2Q_2 + ... + b_nQ_n) + \operatorname{div}(f) \quad (3.19)$$

for some $f \in \overline{K}(C)^*$. So, as soon as $a_1P_1 + a_2P_2 + ... + a_mP_m = b_1Q_1 + b_2Q_2 + ... + b_nQ_n$, we know that D is principal. Moreover, we were actually able to keep track of *all* functions involved so that it is technically possible to explicitely write down the function whose divisor is D, if such a function exists. And just like that, we have rediscovered Abel's theorem:

Theorem 3.84 (Abel) *Let E be a smooth elliptic curve defined over K and $D = \sum_{P \in E} n_P(P) \in$ $\operatorname{Div}^0(E)$ be given.*

$$\textit{If } \sum_{P \in E} n_P P = \mathcal{O}, \textit{ then } D \textit{ is principal.}$$

Notice that $\sum_{P \in E} n_P P$ is a finite sum of points of E, and not a divisor.

Jacobi's Theorem

We here keep the notation of the previous section and we further let

$$\begin{aligned} P &= a_1P_1 + a_2P_2 + ... + a_mP_m \text{ and} \\ Q &= b_1Q_1 + b_2Q_2 + ... + b_nQ_n. \end{aligned}$$

Does the converse of Abel's theorem holds? That is, if D is a principal divisor, then does it imply that $P = Q$? First, we already know that all principal divisors have degree zero by Proposition 3.72. Next, by the method used to prove Abel's theorem, we also know that the arbitrary D we started with can always be written as

$$D = (P) - (Q) + \operatorname{div}(f)$$

for some $f \in \overline{K}(C)^*$. Therefore,

$$D \textit{ is principal if and only if } (P) - (Q) \textit{ is principal.}$$

It thus suffices to show:

Lemma 3.85 *Let E be a smooth elliptic curve defined over K and P, $Q \in E$ be given. Then,*

$$(P) - (Q) \textit{ is principal if and only if } P = Q.$$

Proof. Let's begin by the easy implication and assume that $P = Q$. Then, $(P) - (Q) = \mathbf{0} = \operatorname{div}(1)$, and so $(P) - (Q)$ is principal.

As for the converse, we now assume that $(P) - (Q)$ is principal. Then, there is a f in the function field of E such that $\operatorname{div}(f) = (P) - (Q)$. Hence, it suffices to show that f is constant in order to get the desired result. Now, $\operatorname{div}(f) \geq -(Q)$ and $\mathbf{0} \geq -(Q)$ so that $\mathcal{L}((Q))$ contains both f and the constant functions. But since $\deg((Q)) > 2g - 2$, we can apply the corollary of the Riemann-Roch theorem (see Corollary 3.80) to get that $l((Q)) = \deg((Q)) - g + 1 = 1$. We then conclude that f is constant. Hence, $(P) - (Q) = \operatorname{div}(f) = \mathbf{0}$, which implies that $P = Q$.□

And so we have proved the converse of Abel's theorem, which was originally due to Jacobi.

Theorem 3.86 (Jacobi) *Let E be a smooth elliptic curve defined over K and let $D = \sum_{P \in E} n_P(P) \in \operatorname{Div}^0(E)$ be given.*

$$\textit{If } D \textit{ is principal, then } \sum_{P \in E} n_P P = \mathcal{O}.$$

Lastly, we take the time to re-write the complete result we just shown.

Theorem 3.87 (Abel-Jacobi) *Let E be a smooth elliptic curve defined over K and let $D = \sum_{P \in E} n_P(P) \in \operatorname{Div}(E)$ be given. Then,*

$$D \textit{ is principal if and only if } \deg(D) = 0 \textit{ and } \sum_{P \in E} n_P P = \mathcal{O}.$$

We therefore have an easy criterion to decide if two divisors are linearly equivalent:

Corollary 3.88 *Let E be a smooth elliptic curve defined over K and let*

$$D_1 = \sum_{P \in E} n_P(P), D_2 = \sum_{P \in E} m_P(P) \in \mathrm{Div}(E)$$

be given. Then,

$$D_1 \sim D_2 \text{ if and only if } \deg(D_1) = \deg(D_2) \text{ and } \sum_{P \in E} n_P P = \sum_{P \in E} m_P P.$$

3.4 The Picard Group

3.4.1 Cryptographic Motivation

In 1985, Koblitz [Kob87] and Miller [Mil86b] independently proposed to use the group of points of an elliptic curve as an alternative to the multiplicative group of a finite field used by ElGamal [ElG85a]. Now, elliptic curves posses the remarkable property that its Jacobian coincide with the points of the curve themselves. Hence, we have the choice of understanding the group of points on an elliptic curve in two ways:

- In terms of points, tangent and secant lines, or
- As the natural abelian variety isomorphic to the zero part of the Picard group.

There is no need to say that the first interpretation is by far the simplest and that the chord-and-tangent rule can be understood by anybody. Then why should we even consider the second interpretation? Well, suppose that one hopes to find *another* family of groups suitable for DL-based cryptography. In that case, if we are only aware of the first interpretation, we might try to vary the curve and try to find one for which the points do form a group. Unfortunately, like we already mentionned for the case of hyperelliptic curves, there is no guarantee that a natural group structure exists on the points of the chosen curve.

However, if one starts with *any* smooth curve C, then by construction, the Jacobian of C, $J(C)$, always is (among other things) a good old abelian group. Naively, we could say that constructing the Jacobian of a curve is a clever process that builds a group for which the building blocks forming each element are the points of C. It then becomes a natural process to vary C and look for a $J(C)$ where the computations can be done efficiently and where the DLP seems intractable. Recall that the hunt for such curves was already open in 1985 when elliptic curves made their appearance. Two years later, David Cantor showed how to explicitely compute in the Jacobian of hyperelliptic curves [Can87] and shortly after, Koblitz proposed to use them in cryptography [Kob89].

3.4.2 Construction of the Picard Group

So our task is to start with a smooth curve C for which the points do not necessarily form a group, and use them to build a group out of it. In other words, we want to interpret the group law on an elliptic curve in terms of divisors in such a way that this process could be applied to other curves as well.

Let E be a smooth elliptic curve. The very first step is to know which divisor will play the role of a point $P \in E$. A natural candidate is of course the divisor (P) (that is, $n_P = 1$ is the only nonzero coefficient). We should normally have that the point at infinity corresponds to the identity element of $\mathrm{Div}(E)$, which is not currently the case since $(\mathcal{O}) \neq \mathbf{0}$. An easy fix-up is to associate the divisor $(P) - (\mathcal{O})$ to the point P, which we will informally denote by $P \leftrightsquigarrow (P) - (\mathcal{O})$. Now, let's see what happens when we add two points $P, Q \in E$. Let $R := P + Q$. Following our association,

$$\begin{array}{ccc} E & & \mathrm{Div}(E) \\ P & \leftrightsquigarrow & (P) - (\mathcal{O}) \\ Q & \leftrightsquigarrow & (Q) - (\mathcal{O}) \\ P + Q & \leftrightsquigarrow & (P) + (Q) - 2(\mathcal{O}) \end{array}$$

and hence the point R should correspond to the divisor $(P) + (Q) - 2(\mathcal{O})$ as well, which means that we really want to view the divisors $(R) - (\mathcal{O})$ and $(P) + (Q) - 2(\mathcal{O})$ as representing the *same* point. We therefore want to define an explicit *equivalence relation* on divisors in order to resolve this ambiguity.

But before that takes us too far afield, let's remark that we really don't need to consider all divisors here. Indeed, what do the divisors $(P) - (\mathcal{O})$, $(Q) - (\mathcal{O})$ and $(P) + (Q) - 2(\mathcal{O})$ have in common? Well, they all have degree zero. Since the divisors of degree zero, $\mathrm{Div}^0(C)$, form a subgroup of $\mathrm{Div}(C)$, we can hereafter only work with degree zero divisors.

We now proceed to determine this equivalence relation. Starting from the association $P \leftrightsquigarrow (P) - (\mathcal{O})$, we wish to know what are the other members of $\mathrm{Div}^0(E)$ that also correspond to P:

$$\begin{array}{ccc} E & & \mathrm{Div}^0(E) \\ P & \leftrightsquigarrow & (P) - (\mathcal{O}) \\ \mathcal{O} & \leftrightsquigarrow & D \\ P + \mathcal{O} & \leftrightsquigarrow & (P) - (\mathcal{O}) + D \end{array}$$

We could thus express $\mathcal{O}$ as a sum of points of E, say $\mathcal{O} = P_1 + \ldots + P_k$ where the points are not necessarily distinct. Then,

$$\begin{array}{ccc} E & & \mathrm{Div}^0(E) \\ P & \leftrightsquigarrow & (P) - (\mathcal{O}) \\ P_1 + \ldots + P_k & \leftrightsquigarrow & (P_1) + \ldots + (P_k) - k(\mathcal{O}) \\ P + P_1 + \ldots + P_k = P & \leftrightsquigarrow & (P) + (P_1) + \ldots + (P_k) - (k+1)\,(\mathcal{O}) \end{array}$$

Therefore, we want to require that a divisor $\sum_{P\in E} n_P(P)$ be equivalent to $(P)-(\mathcal{O})$ precisely when $\sum_{P\in E} n_P P = P$.

More generally, given two divisors

$$D_1 = a_1(P_1) + a_2(P_2) + ... + a_m(P_m) \text{ and } D_2 = b_1(Q_1) + b_2(Q_2) + ... + b_n(Q_n)$$

in $\operatorname{Div}^0(E)$, we can let

$$P = a_1P_1 + a_2P_2 + ... + a_mP_m \text{ and } Q = b_1Q_1 + b_2Q_2 + ... + b_nQ_n.$$

Thus,

$$\begin{array}{ccc} E & & \operatorname{Div}^0(E) \\ P & \leftrightsquigarrow & D_1 \\ Q & \leftrightsquigarrow & D_2 \end{array}$$

and we will want that D_1 be equivalent to D_2 if and only if $P = Q$. Interesting. This characterization in fact turns out to be closely related to the Abel-Jacobi theorem. Indeed, recall that Corollary 3.88 states that $P = Q$ is a necessary and sufficient condition to have $D_1 \sim D_2$ (since D_1 and D_2 both have degree zero). The equivalence relation we were looking is thus no other than the linear equivalence of divisors.

We can then remove the ambiguity in our correspondence by considering the quotient group $\operatorname{Div}^0(E)/\operatorname{Princ}(E)$. As this group is defined in terms of points and functions (and does not involve the group law on the elliptic curve), it can then be defined for a general curve as well.

Definition 3.89 *Let C be a smooth algebraic curve over K. The group* $\operatorname{Div}(C)/\operatorname{Princ}(C)$ *is called the* Picard group *or the* divisor class group *of C and is denoted by* $\operatorname{Pic}(C)$. *The* degree zero part of the Picard group, $\operatorname{Pic}^0(C)$, *is simply* $\operatorname{Div}^0(C)/\operatorname{Princ}(C)$. *Furthermore, the class of a divisor $D \in \operatorname{Div}^0(C)$ in $\operatorname{Pic}^0(C)$ will be donoted by $[D]$.*

It is therefore possible to vary the curve and study the group $\operatorname{Pic}^0(C)$ from a cryptographic point of view. For elliptic curves, it is now a routine exercise with the tools at hand to prove the following result[34].

Proposition 3.90 *Let E be a smooth elliptic curve over K. Then the map*

$$\begin{array}{ccc} E & \rightarrow & \operatorname{Pic}^0(E) \\ P & \mapsto & [(P)-(\mathcal{O})] \end{array}$$

[34] The proof can also be found in [Sil86, Proposition III.3.4].

is a group isomorphism with well-defined inverse

$$\begin{array}{rcl} \mathrm{Pic}^0(E) & \to & E \\ \left[\sum_{P\in E} n_P(P)\right] & \mapsto & \sum_{P\in E} n_P P. \end{array}$$

Of course, the structure of $\mathrm{Pic}^0(C)$ can be as rich as an elliptic curve, but could also be quite trivial in some cases.

Example 3.91 *As we saw in example 3.75 , all divisors of degree zero on $\mathbb{P}^1$ are principal. Hence,* $\mathrm{Pic}^0\left(\mathbb{P}^1\right)$ *is the trivial group with only one element.*

3.4.3 The Jacobian

So far, we know that we can start with a smooth curve C (for which the set of points does not necessarily form a group) and build the group $\mathrm{Pic}^0(C)$. We can however go one step further as it turns out that $\mathrm{Pic}^0(C)$ is naturally isomorphic to an abelian variety.

Theorem 3.92 *Let C be a smooth algebraic curve of genus g defined over an algebraically closed field. Then, there exists an abelian variety $J(C)$ of dimension g and an isomorphism of groups*

$$\varphi : \mathrm{Pic}^0(C) \to J(C).$$

The variety $J(C)$ is called the Jacobian of C.

The proof of this result can be found in [Sil94, Proposition III.2.6]. For a more complete treatment, please refer to [Wei48]. As well, take note that an explicit construction of the Jacobians of hyperelliptic curves is given in Mumford's Tata lectures on Theta II [Mum84, Chapter IIIa].

For cryptographic applications, we of course do not work in all of $J = J(C)$, but in a finite subgroup. If C is defined over a perfect field K, then we can consider the subset of J whose elements are of the form $\varphi\left([D]\right)$, where D is a divisor defined over K (that is, $D^\sigma = D$ for every $\sigma \in \mathrm{Gal}\left(\overline{K}/K\right)$). When C is understood, we often denote this set by $J\left(K\right)$ and we have that $J\left(K\right)$ is a subgroup of J. Lastly, the elements of $J\left(K\right)$ are called the *K-points of J*. More details can be found in [CF05, Section 4.4.4].

Having a structure of an abelian variety to work with is certainly an attractive feature for cryptographic applications. However, it could just be as interesting to consider a *wider* family of algebraic varieties. For instance, an algebraic group is, loosely speaking, a variety (affine or projective) that is also a good old group and for which the addition and inverse maps are also morphisms. As a result, the cryptographic potential of commutative algebraic groups are worth exploring.

Definition 3.93 *Let G be an algebraic variety. Suppose that G is also a group with identity $\mathcal{O} \in G$ and that the addition law $\oplus : G \times G \to G$ and inverse map $\ominus : G \to G$ are morphisms. Then, $(G, \mathcal{O}, \oplus, \ominus)$ is said to be an* algebraic group*, or a* group variety*. Also, G is said to be a* commutative algebraic group *if the underlying group is abelian.*

For instance, elliptic curves are commutative algebraic groups. Two other fundamental examples of commutative algebraic groups are the *additive group*

$$\mathbb{G}_{\mathrm{a}} \cong \mathbb{A}^1$$

and the *multiplicative group*

$$\mathbb{G}_{\mathrm{m}} \cong \left\{ x \in \mathbb{A}^1 \middle| \, x \neq 0 \right\},$$

which will be at the forefront of the explicit cryptosystem that we will construct in Chapter 5.

More generally, as we will shortly see, generalized Jacobians are commutative algebraic groups that are not, in general, abelian varieties. *À propos*, we can now say that we have the appropriate background needed to explore the cryptographic potential of generalized Jacobians, which is the object of next chapter.

Chapter 4

Generalized Jacobians and Cryptography

"What makes discrete log based cryptosystems work
is that they are based on the mathematics of algebraic groups.
An algebraic group is both a group and an algebraic variety.
The group structure allows you to multiply and exponentiate.
The variety structure allows you to express all elements
and operations in terms of polynomials, and therefore
in a form that can be efficiently handled by a computer."

- Rubin & Silverberg

This chapter aims at introducing generalized Jacobians in the context of cryptography. Surprisingly, in order to use these structures in practice, only a minimum of results from this theory are needed. This will allow us to quickly focus on concrete applications (and hopefully not get lost in technical details). This will indeed be possible since the underlying ideas behind the construction of both (ordinary) Jacobians and generalized Jacobians truly are the same. Namely,

1. Start with your favorite algebraic curve
2. Consider its divisors of degree zero
3. (Cleverly) define an equivalence relation on them
4. Find a canonical representative for each class

The first two steps are identical in both approaches. Now, for generalized Jacobians, a new equivalence relation needs to be defined. This crucial step will ensure the very existence of the commutative algebraic groups we are looking for: the *generalized Jacobians*.

Francesco Severi was the first to explicitly mention generalized Jacobians in his work *'Funzioni quasi abeliane'* of 1947 [Sev47, Chapter II], where an extensive bibliography can also be found. His treatment was however limited to the case where the base field was the field of complex numbers. In 1950, Maxwell Rosenlicht had just completed his thesis *'Equivalence Concepts on an Algebraic Curve'* under the supervision of Oscar Zariski at Harvard. His dissertation contained the construction and properties of generalized Jacobians in the most global setting. His trilogy of articles [Ros52, Ros54, Ros75] published in the Annals of Mathematics contains the essential of the results on generalized Jacobians. Another excellent reference is *'Groupes algébriques et corps de classes'* [Ser75] of Jean-Pierre Serre, which provides the necessary background on algebraic curves as well[1].

Throughout this chapter, and in order to avoid confusion, the term *Jacobian* alone will denote the *'usual'* Jacobian as defined in the last chapter (see Section 3.4.3), whereas the qualifier *'generalized'* will always be explicitely employed when referring to *generalized Jacobians*. Finally, most of our notation concerning generalized Jacobians will follow Serre's exposition [Ser88, Chapter V].

4.1 Motivation

We here wish to give a flavor as to *'why'* generalized Jacobians are worth considering for cryptographic applications. The following observations, half rigourous, half heuristic, have in fact been the motivation behind our research on this subject. It is hoped that sharing these first ideas right from the start will highlight the cryptographic potential of these structures.

In what follows, let C be a curve defined over a finite field $\mathbb{F}_q$. As usual, let J denote its Jacobian variety and $J(\mathbb{F}_q)$ be the finite subgroup consisting of the $\mathbb{F}_q$-points of J. We will also assume that we chose C such that the discrete logarithm problem in $J(\mathbb{F}_q)$ is believed to be intractable (so we might think of C as being a carefully chosen elliptic or hyperelliptic curve, for example).

As its name suggests, generalized Jacobians will be defined in such a way that the usual Jacobian will be subsumed under the new concept. A natural way to proceed is to modify the equivalence relation on the divisors of C such that it coincides with linear equivalence in some specific cases. Since we need the new equivalence classes to form a group (with operation induced from the formal addition of divisors), it will also be required that the set of divisors

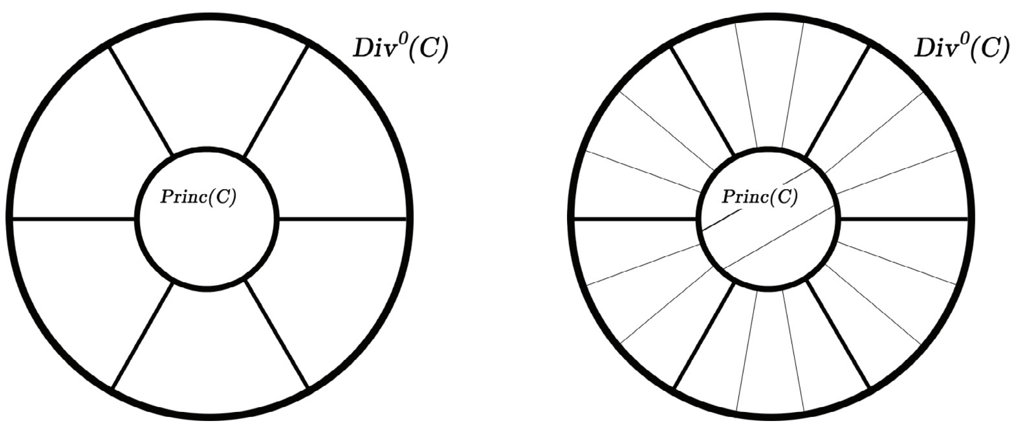
[1] For those uncomfortable with *la langue de Molière*, an english translation [Ser88] is also available.

equivalent to the zero divisor **0** forms a subgroup. More precisely, the new equivalence relation, called $\mathfrak{m}$-*equivalence*[2], will enjoy the following property:

If two divisors are $\mathfrak{m}$*-equivalent, then they are linearly equivalent as well.* (4.1)

This implies that each $\mathfrak{m}$-equivalence class will be a subdivision of an original divisor class. In the following schematic representation of a Jacobian versus a generalized Jacobian, the bold lines represent the divisor classes under linear equivalence while the thin lines show the subdivisions obtained when considering $\mathfrak{m}$-equivalence classes. We are therefore in the presence of the following *'before-and-after'* makeover:

Usual Jacobian ***Generalized Jacobian***



Linear equivalence:
$D \sim D'$ *iff* $\exists f \in \overline{K}(C)^*$ *such that* $D - D' = \mathrm{div}(f)$

$\mathfrak{m}$***-equivalence:***
$D \sim_{\mathfrak{m}} D'$ *iff* $D \sim D'$, *plus a condition to be determined*

Figure 4.1: Similarities between usual and generalized Jacobians

The idea of having these two equivalence relations, one being a *'refinement'* of the other, is somehow like the task of delivering mail on a street with appartment buildings. At a higher level, we can view all individuals living in one building as *'sharing the same class'*, while at a smaller scale, we could just as well define new classes according to the occupants of each appartment. And just like with condition (4.1), the persons sharing the same appartment *must* also live in

[2] To be completely rigorous, we should state that $\mathfrak{m}$-equivalence will be defined on divisors having disjoint support with $\mathfrak{m}$. The complete details will be given when we formally define $\mathfrak{m}$-equivalence in Section 4.2.

the same building. Following this analogy, notice that we already know how to provide a unique street address to each building and it thus remains to determine how to systematically attach an appartment number to each subdivision.

The plan is now to see how requirement (4.1) alone already gives us a feeling of the cryptographic properties of generalized Jacobians. In fact, regardless of the precise definition of $\mathfrak{m}$-equivalence, it is immediately possible to deduce some interesting arithmetic properties of these algebraic groups. The following observations will of course often rely on our prior knowledge about the Jacobian.

REPRESENTATION OF ELEMENTS. In order to identify a $\mathfrak{m}$-equivalence class, it suffices to specify a divisor class modulo linear equivalence together with an extra piece of information that will uniquely identify in which subdivision it lies. We can then see an element of the generalized Jacobian as a pair, where the first component is an element of the Jacobian and the second is a label that specifies the subdivision.

GROUP LAW. We now turn our attention to the group law algorithm, since it is at the heart of any cryptographic application using a group structure. We claim that the group operation on the generalized Jacobian will carry all the information needed to perform the addition on the Jacobian. In other words, suppose that we completely forgot how to add two elements P and Q of J but somehow managed to remember the group operation on the generalized Jacobian $J_{\mathfrak{m}}$. From the construction of the Jacobian, we know that P and Q respectively correspond to divisor classes (under linear equivalence) with representative D_P and D_Q, say. We could then use the group law on the generalized Jacobian to compute a divisor D_R such that $D_P + D_Q \sim_{\mathfrak{m}} D_R$.

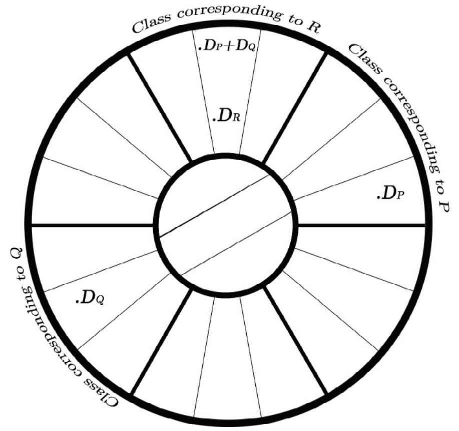


Figure 4.2: Group law on a generalized Jacobian

By (4.1), this implies that $D_P + D_Q \sim D_R$ as well and so the last step is to recover the element R of J corresponding to the equivalence class (modulo linear equivalence) of D_R. Finally, we get that $P + Q = R$, as wanted. Hence, it follows that the group law on the Jacobian can be inferred from the one of the generalized Jacobian. Of course, we are not making any affirmation concerning the efficiency of this reduction. What really matters here is that if we start with a (cryptographically) rich addition on J, it would be surprising to end up with a useless addition on $J_\mathfrak{m}$ (a more precise affirmation will be made later). The above remarks also tells us that we should expect the cost of the explicit group law on $J_\mathfrak{m}$ to be at least as high as the one on J. The gap in efficiency between these two group laws will inevitably depend on *'how much effort'* is required to determine in which subdivision (i.e. $\mathfrak{m}$-equivalence class) a given divisor lies.

GROUP ORDER & POINT COUNTING. Let $J_\mathfrak{m}(\mathbb{F}_q)$ denote the subset of $J_\mathfrak{m}$ formed by $\mathfrak{m}$-equivalence classes whose divisors are $\mathbb{F}_q$-points. Since $J(\mathbb{F}_q)$ is a group, then so is $J_\mathfrak{m}(\mathbb{F}_q)$. Moreover, if we assume that the number s of $\mathfrak{m}$-equivalence classes within one divisor class (under linear equivalence) is finite, then $\#J_\mathfrak{m}(\mathbb{F}_q) = s \cdot \#J(\mathbb{F}_q)$ will be finite as well. We therefore officially designate $J_\mathfrak{m}(\mathbb{F}_q)$ as our chosen candidate for a new group potentially suitable for cryptographic applications. Notice that the order of $J_\mathfrak{m}(\mathbb{F}_q)$ will in general be a composite number. So in practice, we will want s or $\#J(\mathbb{F}_q)$ to possess at least one large prime factor in order to thwart the Pohlig-Hellman attack (see Section 2.7.1) on discrete logarithms of $J_\mathfrak{m}(\mathbb{F}_q)$. In addition, suppose that we choose the curve C such that $\#J(\mathbb{F}_q)$ can be determined in polynomial-time. Then, the cardinality of $J_\mathfrak{m}(\mathbb{F}_q)$ can be efficiently computed if and only if s can be efficiently determined as well.

ORDER OF ELEMENTS AND GENERATORS. The obvious statement is that by Lagrange's theorem, the order of an element of $J_\mathfrak{m}(\mathbb{F}_q)$ must divide $s \cdot \#J(\mathbb{F}_q)$. We can however go one step further. Let $A \in J_\mathfrak{m}(\mathbb{F}_q)$, D be a representative of the $\mathfrak{m}$-equivalence class associated to A and let $P \in J(\mathbb{F}_q)$ be the element corresponding to the linear divisor class of D. Denote by l the order of P in $J(\mathbb{F}_q)$. Then, requirement (4.1) implies that the order of A has to be a multiple of l. Moreover, if A is a generator of $J_\mathfrak{m}(\mathbb{F}_q)$, then P will have no choice but to generate all of $J(\mathbb{F}_q)$.

DISCRETE LOGARITHMS. We saw that the group laws on $J(\mathbb{F}_q)$ and on $J_\mathfrak{m}(\mathbb{F}_q)$ are closely related, and so that raises the possibility that their discrete logarithms could be linked as well. Here is a heuristic argument in the case where $J_\mathfrak{m}(\mathbb{F}_q)$ is a cyclic group. Recall that we are working under the hypothesis that the DLP in $J(\mathbb{F}_q)$ is computationally infeasible. *'Can the DLP on $J_\mathfrak{m}(\mathbb{F}_q)$ be easy?'*, should now (hopefully) be on everybody's lips. So let's assume that it is and see what happens. First, let A be a generator of $J_\mathfrak{m}(\mathbb{F}_q)$ and let D, P, and l be as above.

As mentioned earlier, P will generate $J(\mathbb{F}_q)$ so we can try to solve an instance $Q = kP$ of the discrete logarithm in $J(\mathbb{F}_q)$. Let D_Q be a representative of the class (modulo linear equivalence) associated to Q. So in particular, we have that

$$kD = \underbrace{D + D + ... + D}_{k \text{ times}} \sim D_Q.$$

Moreover, all sums of the form $(k + nl)D$, where n is a non-negative integer, will be linearly equivalent to D_Q as well. Hence, among them, there will be a (smallest) n_0 such that $(k + n_0 l)D \sim_{\mathfrak{m}} D_Q$ since A was a generator of $J_{\mathfrak{m}}(\mathbb{F}_q)$). See Figure 4.3 for a tiny example with $l = 7$, $k = 5$ and $n_0 = 2$.

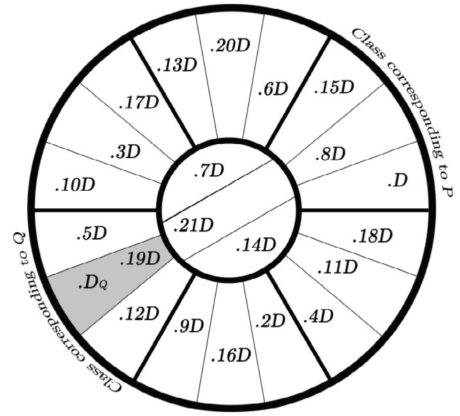


Figure 4.3: Illustrative example with $l = 7$, $k = 5$ and $n_0 = 2$

Now if we let $B \in J_{\mathfrak{m}}(\mathbb{F}_q)$ be the element corresponding to the $\mathfrak{m}$-equivalence class of D_Q, it follows that $B = (k + n_0 l)A$. Under our assumption that the DLP in $J_{\mathfrak{m}}(\mathbb{F}_q)$ is easy, we can therefore recover (with non-negligible probability) $\log_A B \overset{\text{def}}{=} k + n_0 l$. Finally, we obtain the really neat relation

$$\log_P Q = (\log_A B) \bmod l. \tag{4.2}$$

That of course contradicts our first assumption that the discrete logarithm problem in $J(\mathbb{F}_q)$ was computationnaly infeasible. The moral of the story is that we should expect the discrete logarithm problem in $J_{\mathfrak{m}}(\mathbb{F}_q)$ to be at least as hard as the one on $J(\mathbb{F}_q)$. So when trying to construct a $J_{\mathfrak{m}}(\mathbb{F}_q)$ suitable for cryptographic applications, we should therefore start with a curve C for which the DLP in $J(\mathbb{F}_q)$ is believed to be intractable. And to do so, nearly twenty years of research in this direction will be available for us to use.

4.2 Equivalence relation induced from a modulus

Now finally comes the time to explicitely define the $\mathfrak{m}$-equivalence relation which is the key ingredient in the construction of generalized Jacobians. As outlined in the previous section, our strategy will be to first work over an arbitrary algebraically closed field (having of course $\overline{\mathbb{F}}_q$ in mind) in order to get acquainted with the generalized Jacobian, and right before we jump into the applications, we will simply specify a finite subgroup (where all computations can be performed over $\mathbb{F}_q$) to work with.

So let K be an algebraically closed field and C be a smooth algebraic curve defined over K. Recall that we aim at *'refining'* linear equivalence in such a way that the new equivalence classes will be subdivisions of the originals. If we let $D = \sum_{P \in C} n_P(P)$, $D' = \sum_{P \in C} n'_P(P) \in \mathrm{Div}^0(C)$ be given such that D is linearly equivalent to D', then there is a nonzero rational function f in the function field $K(C)$ of C satisfying $\mathrm{div}(f) = D - D'$. That is,

$$\sum_{P \in C} \mathrm{ord}_P(f)(P) = \sum_{P \in C} (n_P - n'_P)(P),$$

which can also be expressed as

$$\mathrm{ord}_P(f) = n_P - n'_P \text{ for all } P \in C.$$

The whole idea behind these equivalence relations is to somehow *'measure'* how much D differs from D'. A possible additional criterion would be to consider a specific point $M \in C$ and check whether $n_M = n'_M$. If it is the case, then $\mathrm{ord}_M(f) = 0$ and so f is defined and nonzero at M. We are thus led to consider the value of $f(M)$. But since f is determined up to multiplication by a nonzero constant (c.f. Section 3.3.3 on page 84), we can then assume without loss of generality that we chose f such that $f(M) = 1$. That will now ensure that our function f is uniquely determined. So now we can consider a second point $N \in C$ distinct from M and wonder if $n_N = n'_N$. In the affirmative, compute $f(N)$ for the record. And we could continue just the same with more points if we please. So let $P_0 := M$, $P_1 := N$, P_2, ..., P_r be the chosen distinct points of C where we want to require that $n_{P_i} = n'_{P_i}$ $(0 \leq i \leq r)$. We could then define a tentative relation '$\dot{\sim}$' as follows:

$$D \dot{\sim} D' \quad \text{iff} \quad D \sim D' \text{ and } n_{P_i} = n'_{P_i} \text{ for } 0 \leq i \leq r.$$

This is clearly an equivalence relation on $\mathrm{Div}^0(C)$. Notice that it can also be rephrased as

$$D \dot{\sim} D' \quad \text{iff} \quad \exists f \in K(C)^* \text{ such that } \mathrm{div}(f) = D - D' \text{ and } \mathrm{ord}_{P_i}(f) = 0 \text{ for } 0 \leq i \leq r.$$

However, that relation merely takes care of ensuring that $f(P_i)$ is defined and nonzero. But once this verification is done, why not taking advantage of the value of $f(P_i)$? So suppose that we have computed the values $f(P_0)$, $f(P_1)$, ..., $f(P_r)$ and wish to compare them somehow. Since we want to end up with an equivalence relation, the safest bet is to work with equalities. We could then look for divisors satisfying

$$1 = f(P_0) = f(P_1) = ... = f(P_r) \tag{4.3}$$

(recall that f was chosen such that $1 = f(M)$ and that $P_0 := M$). This condition certainly is a much stronger requirement than before, as illustrated in Figure 4.4.

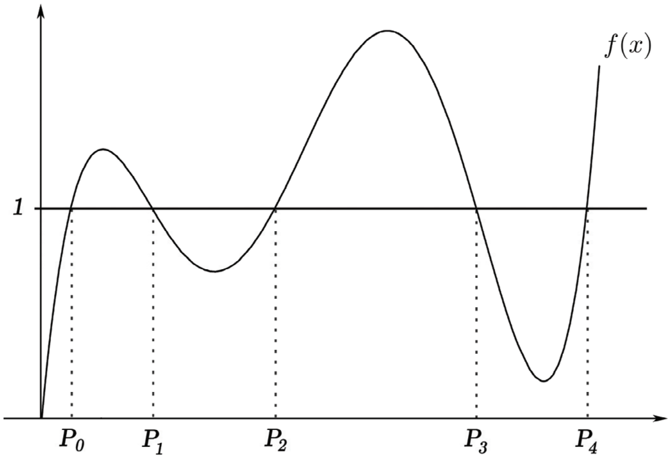


Figure 4.4: A stronger requirement on the function f

Notice that we can express condition (4.3) in a slightly different form which will be directly related to divisors of functions, as condition $f(P_i) = 1$ is equivalent to $\mathrm{ord}_{P_i}(1-f) \geq 1$. We can therefore write down our second candidate:

$D \ddot{\sim} D'$ iff $\exists f \in K(C)^*$ such that $\mathrm{div}(f) = D - D'$ and $\mathrm{ord}_{P_i}(1-f) \geq 1$ for each $P_i \in S$, where $S = \{P_0, P_1, \ldots, P_r\}$.

It is once again a simple matter to check that this indeed defines an equivalence relation. There is yet another modification that might be interesting. Indeed, the condition $\mathrm{ord}_{P_i}(1-f) \geq 1$ says that $1-f$ has a zero at P_i, but the order of this zero is not specified at all. Hence, for an integer $m_i \geq 1$, we could impose the stricter condition that $\mathrm{ord}_{P_i}(1-f) \geq m_i$ if we want. So given positive integers m_0, m_1, ..., m_r, we can consider the following relation:

$$D \overset{\dots}{\sim} D' \quad \text{iff} \quad \exists f \in K(C)^* \text{ such that } \mathrm{div}(f) = D - D' \text{ and } \mathrm{ord}_{P_i}(1-f) \geq m_i \text{ for each } P_i \in S. \tag{4.4}$$

This relation is *reflexive* because $D - D = \mathrm{div}(1)$ and $\mathrm{ord}_{P_i}(0) = \infty$ by convention (see Definition 3.67).

It is *symmetric* as well, for if $D \overset{\dots}{\sim} D'$ with $D - D' = \mathrm{div}(f)$, then $D' - D = \mathrm{div}(1/f)$ and

$$\mathrm{ord}_{P_i}\left(1 - \frac{1}{f}\right) = \mathrm{ord}_{P_i}\left(-\frac{1-f}{f}\right) = \mathrm{ord}_{P_i}(1-f) - \underbrace{\mathrm{ord}_{P_i}(f)}_{=0} \geq m_i.$$

Finally, it is *transitive* since if $D \overset{\dots}{\sim} D'$ and $D' \overset{\dots}{\sim} D''$ with $D - D' = \mathrm{div}(f)$ and $D' - D'' = \mathrm{div}(g)$, then $D - D'' = \mathrm{div}(fg)$ and we have that

$$\begin{aligned} \mathrm{ord}_{P_i}(1-fg) &= \mathrm{ord}_{P_i}((1-f) + (1-g) - (1-f)(1-g)) \\ &\geq \min\Big(\underbrace{\mathrm{ord}_{P_i}((1-f)+(1-g))}_{\geq m_i}, \underbrace{\mathrm{ord}_{P_i}((1-f)(1-g))}_{\geq 2m_i}\Big) \\ &\geq m_i. \end{aligned}$$

We have therefore convinced ourselves through this little exercise that (4.4) is an equivalence relation.

We now take the time to simplify the notations a little. Since we need to specify each point P_i together with an associated positive integers m_i, a compact way to do so would be to write it as the effective divisor

$$\mathfrak{m} = \sum_{i=0}^{r} m_i(P_i).$$

It is also a standard notation to write $f \equiv 1 \bmod \mathfrak{m}$ as a shorthand for the requirement $\mathrm{ord}_{P_i}(1-f) \geq m_i$ for each $P_i \in S$. For this reason, it is customary to call $\mathfrak{m}$ a *modulus* supported on $S_{\mathfrak{m}} = \{P_0, P_1, ..., P_r\}$.

We also want to point out that if a divisor D is such that $D \overset{\dots}{\sim} \mathbf{0}$, then $D = \mathrm{div}(f) = \sum_{P \in C} \mathrm{ord}_P(f)(P)$ where $\mathrm{ord}_{P_i}(f) = 0$ for $1 \leq i \leq r$. That is, $\mathrm{supp}(D)$ is disjoint from $S_{\mathfrak{m}}$. Consequently, it will be convenient to define our equivalence relation (only) on the set of divisors having support disjoint from $S_{\mathfrak{m}}$. We are finally ready to rewrite (4.4) with the new terminology and to formally define $\mathfrak{m}$-equivalence.

Definition 4.1 *Let $\mathfrak{m}$ be an effective divisor supported on $S_\mathfrak{m}$ and let D and D' be two divisors prime to $S_\mathfrak{m}$. We say that D and D' are* $\mathfrak{m}$-equivalent, *and write $D \sim_\mathfrak{m} D'$ if*

$$\exists f \in K(C)^* \text{ such that } \operatorname{div}(f) = D - D' \text{ and } f \equiv 1 \bmod \mathfrak{m}.$$

As promised in Section 4.1, this definition readily implies property (4.1), which said that *'If two divisors are $\mathfrak{m}$-equivalent, then they are linearly equivalent as well'.* Since this is such an important property for us, we now grant it the status it deserves.

Lemma 4.2 *Let D and D' be two divisors prime to $S_\mathfrak{m}$. If $D \sim_\mathfrak{m} D'$, then $D \sim D'$ as well.*

If we denote by $[D]$ (respectively $[D]_\mathfrak{m}$) the class of D under linear equivalence (respectively $\mathfrak{m}$-equivalence), then the above fact implies that $[D]_\mathfrak{m} \subseteq [D]$, as wanted. We were therefore right when we claimed that *'each $\mathfrak{m}$-equivalence class is a subdivision of an original divisor class'.*

4.3 Generalized Jacobian Varieties

We here keep the conventions and notations of the previous section and we begin by introducing a few more definitions in order to be able to easily work with $\mathfrak{m}$-equivalence. So let $\operatorname{Div}_\mathfrak{m}(C)$ be the subgroup of $\operatorname{Div}(C)$ formed by all divisors of C which are prime to $S_\mathfrak{m}$. Let also $\operatorname{Div}^0_\mathfrak{m}(C)$ be the subgroup of $\operatorname{Div}_\mathfrak{m}(C)$ composed of divisors of degree zero. Moreover, let $\operatorname{Princ}_\mathfrak{m}(C)$ be the subset of principal divisors which are $\mathfrak{m}$-equivalent to the zero divisor. In other words, $\operatorname{Princ}_\mathfrak{m}(C) = [\mathbf{0}]_\mathfrak{m} = \{\operatorname{div}(f) \,|\, f \in K(C)^* \text{ and } f \equiv 1 \bmod \mathfrak{m}\}$.

Since we want to show that the set of $\mathfrak{m}$-equivalence classes is indeed a group, the first step will be to show that $\operatorname{Princ}_\mathfrak{m}(C)$ is a subgroup of $\operatorname{Div}^0_\mathfrak{m}(C)$. This is a formality. First notice that $\mathbf{0} \in \operatorname{Princ}_\mathfrak{m}(C)$ by definition and that $\operatorname{Princ}_\mathfrak{m}(C) \subseteq \operatorname{Div}^0_\mathfrak{m}(C)$ since all principal divisors have degree zero (c.f. Proposition 3.72). Now let $D \sim_\mathfrak{m} \mathbf{0}$ be given. By symmetry, $\mathbf{0} \sim_\mathfrak{m} D$ as well so there is a $f \in K(C)^*$ such that $\mathbf{0} - D = \operatorname{div}(f)$ and $f \equiv 1 \bmod \mathfrak{m}$. Thus, $(-D) - \mathbf{0} = \operatorname{div}(f)$, which shows that $-D \sim_\mathfrak{m} \mathbf{0}$. Lastly, let D and D' be two divisors prime to $S_\mathfrak{m}$ such that $D \sim_\mathfrak{m} \mathbf{0}$ and $D' \sim_\mathfrak{m} \mathbf{0}$. By the above argument $-D' \sim_\mathfrak{m} \mathbf{0}$, and so $\mathbf{0} \sim_\mathfrak{m} -D'$ (by symmetry). Then, $D \sim_\mathfrak{m} \mathbf{0}$ and $\mathbf{0} \sim_\mathfrak{m} -D'$ implies $D \sim_\mathfrak{m} -D'$ (by transitivity). There is thus a $f \in K(C)^*$ such that $\operatorname{div}(f) = D + D'$ and $f \equiv 1 \bmod \mathfrak{m}$. It then follows that $D + D' \sim_\mathfrak{m} \mathbf{0}$, as wanted. We have thus completed our homework and verified that $\operatorname{Princ}_\mathfrak{m}(C)$ is indeed a subgroup of $\operatorname{Div}^0_\mathfrak{m}(C)$.

We will therefore consider the quotient group $\operatorname{Div}^0_\mathfrak{m}(C)/\operatorname{Princ}_\mathfrak{m}(C)$, which will be denoted by $\operatorname{Pic}^0_\mathfrak{m}(C)$. We are therefore in possession of an abelian group whose elements are the $\mathfrak{m}$-equivalence classes. We are now crossing our fingers and hoping that there exists an algebraic group isomorphic to $\operatorname{Pic}^0_\mathfrak{m}(C)$.

As pointed out in the introduction, this reasoning is in fact similar to the case of the (usual) Jacobian J, which was treated in *'Variétés abéliennes et courbes algébriques' (i.e. Abelian varieties and algebraic curves)* [Wei48] by André Weil[3]. Recall that the Jacobian of C is an *abelian variety* of dimension equal to the genus of C (c.f. Theorem 3.92). It is therefore a *complete* algebraic variety. However, the generalized Jacobians won't in general enjoy this property. Maxwell Rosenlicht [Ros54, p.515] summarizes the situation as follows:

> *"We proceed to construct a generalized Jacobian variety (...). The method is the same as that used by Weil to construct the ordinary Jacobian variety of C, but unfortunately the noncompleteness of our generalized Jacobians will considerably complicate the steps used in [Wei48], and that proof cannot be taken over verbatim to the present case."*

Regrettably, it would therefore be much too involving to reproduce his construction here. We will instead have to be satisfied with an outline of the technique used. But before we do so, we of course state the existence theorem whose complete proof can be found in the original article of Rosenlicht [Ros54] as well as in [Ser88, Chapter V, in particular Prop. 2 and Thm 1(b)].

Theorem 4.3 *Let K be an algebraically closed field and C be a smooth algebraic curve of genus g defined over K. Then for every modulus $\mathfrak{m}$, there exists a commutative algebraic group $J_{\mathfrak{m}}$ isomorphic to the group $\operatorname{Pic}^0_{\mathfrak{m}}(C)$.The dimension π of $J_{\mathfrak{m}}$ is given by*

$$\pi = \begin{cases} g & \text{if } \mathfrak{m} = \mathbf{0}, \\ g + \deg(\mathfrak{m}) - 1 & \text{otherwise.} \end{cases} \tag{4.5}$$

We can finally present the definition of a generalized Jacobian:

Definition 4.4 *The algebraic group $J_{\mathfrak{m}}$ is called the* generalized Jacobian *of the curve C with respect to the modulus $\mathfrak{m}$.*

As announced, we now outline the main steps of the proof[4] in a quick summary, which follows the terminology and conventions of *'Foundations of Algebraic Geometry'* of Weil [Wei46]. The main idea is to employ the method of *generic points* in order to first build a *birational group Y* defined over K, and then apply a result of Weil yielding the existence and uniqueness of a true algebraic group birationnaly isomorphic to Y (over K). The birational group Y is obtained by

[3] An amusing fact: Émile Picard was the co-advisor of André Weil during his studies at the Université de Paris in the 1920s.

[4] Please take note that the ideas behind this proof are not needed for the sequel.

endowing the π-fold symmetric product $C^{(\pi)}$ of C with a commutative rational composition law $\circledast : Y \times Y \to Y$ defined over K, where π is the arithmetic (or virtual) genus of the singular curve $C_{\mathfrak{m}}$ defined by $\mathfrak{m}$ [Ser88, Chapter IV, Section 4]. Finally, the theorem of Weil [Wei48, Wei55] ensures the existence of an algebraic group $J_{\mathfrak{m}}$ together with a birational map $\Phi : C^{(\pi)} \to J_{\mathfrak{m}}$ defined over K satisfying $\Phi(P) + \Phi(Q) = \Phi(P \circledast Q)$, where P and Q are independent generic points of $C^{(\pi)}$.

If we now go back to very basic properties of generalized Jacobians, notice that there are many $J_{\mathfrak{m}}$ associated to a fixed curve C, one for each choice of modulus $\mathfrak{m}$ in fact. This contrasts with the (usual) Jacobian which is uniquely determined from C. And of course, it might happen that two generalized Jacobians $J_{\mathfrak{m}}$ and $J_{\mathfrak{m}'}$ be isomorphic *as abelian groups* even if $\mathfrak{m} \neq \mathfrak{m}'$. The large quantity of generalized Jacobians we can choose from certainly is a potential advantage for cryptographic applications since generating a suitable curve seems *a priori* much harder than selecting (random) points for the modulus.[5]

4.3.1 Link Between Ordinary and Generalized Jacobians

We now want to establish the existence of a canonical surjective homomorphism from $J_{\mathfrak{m}}$ to J, which can then be used to compare various properties of these two groups. First recall that by Theorem 3.92, there is a natural group isomorphism φ between J and the group $\operatorname{Pic}^0(C)$ of divisors of degree zero modulo linear equivalence:

$$\varphi : \operatorname{Pic}^0(C) \xrightarrow{\sim} J \tag{4.6}$$

By Lemma 4.2, we also know that for a divisor D prime to $S_{\mathfrak{m}}$, we have $[D]_{\mathfrak{m}} \subseteq [D]$. Hence, there is a surjective homomorphism σ from $\operatorname{Pic}^0_{\mathfrak{m}}(C)$ to $\operatorname{Pic}^0(C)$ that sends $[D]_{\mathfrak{m}}$ to the divisor class $[D]$:

$$\begin{array}{rccc} \sigma : & \operatorname{Pic}^0_{\mathfrak{m}}(C) & \twoheadrightarrow & \operatorname{Pic}^0(C) \\ & [D]_{\mathfrak{m}} & \mapsto & [D] \end{array} \tag{4.7}$$

Futhermore, Theorem 4.3 implied the existence of a group isomorphism ψ between $\operatorname{Pic}^0_{\mathfrak{m}}(C)$ and the generalized Jacobian $J_{\mathfrak{m}}$:

$$\psi : \operatorname{Pic}^0_{\mathfrak{m}}(C) \xrightarrow{\sim} J_{\mathfrak{m}} \tag{4.8}$$

[5] Of course, this remark only concerns algebraic curves suitable for public-key cryptography, such as elliptic and hyperelliptic curves.

The following diagram can therefore be obtained by combining (4.6), (4.7), and (4.8):

$$\begin{tikzcd}
\operatorname{Pic}^0_{\mathfrak{m}}(C) \arrow[r, "\psi"] \arrow[d, "\sigma"'] & J_{\mathfrak{m}} \\
\operatorname{Pic}^0(C) \arrow[r, "\varphi"] & J
\end{tikzcd}$$

As a result, there is a surjective homomorphism $\tau := \varphi \circ \sigma \circ \psi^{-1}$ from $J_{\mathfrak{m}}$ to J:

$$\tau : J_{\mathfrak{m}} \twoheadrightarrow J.$$

If the map τ and its inverse can be efficiently computed, then it can be used for instance to *'transport'* the group law on $J_{\mathfrak{m}}$ to the one on J, as put forward in Section 4.1 Indeed, given P and Q in J, their sum can be computed as follows. Since τ is onto, first find any A and B in $J_{\mathfrak{m}}$ such that $\tau(A) = P$ and $\tau(B) = Q$. Then add A and B using the known group operation on $J_{\mathfrak{m}}$ to obtain the element C. Then, $\tau(C)$ is the sum of P and Q as

$$\tau(C) = \tau(A + B) = \tau(A) + \tau(B) = P + Q.$$

Notice that this is well-defined since for any choice A' and B' satisfying $\tau(A') = P$ and $\tau(B') = Q$ and such that $A' + B' = C'$, we will have

$$\tau(C') = \tau(A' + B') = \tau(A') + \tau(B') = \tau(A) + \tau(B) = \tau(A + B) = \tau(C).$$

An interesting object of study certainly is the kernel $L_{\mathfrak{m}}$ of the map τ since it it might give us information about the structure of $J_{\mathfrak{m}}$.

4.3.2 Fundamental Exact Sequence

First notice that since τ is a homomorphism, then $L_{\mathfrak{m}}$ is a subgroup of $J_{\mathfrak{m}}$. We can then consider the following short exact sequence (of abelian groups):

$$0 \longrightarrow L_{\mathfrak{m}} \xrightarrow{\text{inclusion}} J_{\mathfrak{m}} \xrightarrow{\tau} J \longrightarrow 0$$

Therefore,

The generalized Jacobian $J_{\mathfrak{m}}$ is an extension of the usual Jacobian J by $L_{\mathfrak{m}}$.

Evidently, the direct product $L_{\mathfrak{m}} \times J$ whose group law is given by $(k_1, P_1) + (k_2, P_2) = (k_1k_2, P_1 + P_2)$ can be seen as a *trivial extension* of J by $L_{\mathfrak{m}}$ as it satisfies the exact sequence:

$$0 \longrightarrow L_{\mathfrak{m}} \xrightarrow{\iota} L_{\mathfrak{m}} \times J \xrightarrow{\rho} J \longrightarrow 0,$$

where $\iota(k) = (k, 0)$ and $\rho(k, P) = P$. Notice that this direct product is not really interesting from a cryptographic point of view since

$$n(k,P) \overset{\text{def}}{=} \underbrace{(k,P) + (k,P) + ... + (k,P)}_{n \text{ times}} = (k^n, nP),$$

and therefore offers no cryptographic advantage over the cartesian product of $L_{\mathfrak{m}}$ with J. It would therefore be really convenient to know at this point under which circumstances can $J_{\mathfrak{m}}$ become a direct product. Luckily, as we are about to see, this almost never happens and there is moreover a really simple criterion to fulfill in order to avoid this degenerate case. The answer once again resides in a theorem of Rosenlicht, whose proof concludes the article *'Generalized Jacobian Varieties'* [Ros54, Thm 13]:

Theorem 4.5 (Rosenlicht) *Let C be a smooth algebraic curve defined over an algebraically closed field, J be the Jacobian of C and $J_{\mathfrak{m}}$ be the generalized Jacobian of C with respect to a modulus $\mathfrak{m}$. If the genus g of C and the dimension π of $J_{\mathfrak{m}}$ satisfy*

$$0 < g < \pi, \tag{4.9}$$

then there exists no regular cross section for the natural homomorphism $\tau : J_{\mathfrak{m}} \twoheadrightarrow J$.

Recall that a *regular cross section* for τ is an everywhere defined rational map $\tilde{\tau} : J \to J_{\mathfrak{m}}$ such that $\tau \circ \tilde{\tau}$ is the identity on J:

$$\begin{tikzcd} J_{\mathfrak{m}} \arrow[r, shift right, "\tau"'] & J. \arrow[l, shift right, "\tilde{\tau}"'] \end{tikzcd}$$

But if we consider the direct product $L_{\mathfrak{m}} \times J$, there is an obvious cross section $\tilde{\rho} : J \to L_{\mathfrak{m}} \times J$ given by $\tilde{\rho}(P) = (1, P)$ since

$$\tilde{\rho}(P + Q) = (1, P + Q) = (1, P) + (1, Q) = \tilde{\rho}(P) + \tilde{\rho}(Q).$$

Therefore, requirement (4.5) suffices to guarantee that $J_{\mathfrak{m}}$ is not the direct product $L_{\mathfrak{m}} \times J$. And since the dimension of $J_{\mathfrak{m}}$ is given by (4.5), it follows that $g < \pi$ is equivalent to $\deg(\mathfrak{m}) \geq 2$. These observations can now be stated formally:

Corollary 4.6 *Let C be a smooth algebraic curve of genus g defined over an algebraically closed field and $J_{\mathfrak{m}}$ be the generalized Jacobian of C with respect to a modulus $\mathfrak{m}$. If $g \geq 1$ and* $\deg(\mathfrak{m}) \geq 2$*, then $J_{\mathfrak{m}}$ is not a trivial direct product.*

This amazingly simple statement suggests to consider the case where the two lower bounds $g = 1$ and $\deg(\mathfrak{m}) = 2$ are simultaneously reached. For this reason, the generalized Jacobians of an elliptic curve with respect to a modulus $\mathfrak{m} = (M) + (N)$, where $M \neq N$, is the family of groups we chose to study and put forward for cryptographic applications in Chapter 5.

4.4 Group Extensions

Since $J_{\mathfrak{m}}$ is an extension of J by $L_{\mathfrak{m}}$, it is possible to say a little bit more concerning the representation of the elements of $J_{\mathfrak{m}}$ as well as its group operation. Indeed, from the theory of group extensions which can be found in [HS71, Chapter III] and [Wei69, Chapter 5], the following classical results are available[6].

Theorem 4.7 *Let $(G,+)$ be a group and $(A,\cdot)$ be a commutative group. Let also*

$$1 \longrightarrow A \xrightarrow{i} \overline{G} \xrightarrow{p} G \longrightarrow 0 \tag{4.10}$$

be a short exact sequence defining the group extension $\overline{G}$ of G by A. Denote by $\oplus$ the group operation on $\overline{G}$. Then,

1. *Let $s : G \to \overline{G}$ be a (set-theoretic) section for p (that is, $p \circ s$ is the identity on G but s doesn't have to be a group homomorphism). Then the map*

$$\begin{aligned} A \times G &\longrightarrow \overline{G} \\ (a,\sigma) &\longmapsto a \oplus s(\sigma) \end{aligned}$$

*is a bijection **of sets.** Hence, each element of $\overline{G}$ can be unequivocally represented as a pair (a,σ), where $a \in A$ and $\sigma \in G$.*

2. *There is a well-defined natural action of G on A given by*

$$\begin{aligned} A \times G &\longrightarrow A \\ (a,\sigma) &\longmapsto a^{\sigma} := x \oplus a \ominus x, \end{aligned}$$

where x is any element of $\overline{G}$ satisfying $p(x) = \sigma$ and $\ominus x$ denotes the inverse of x in $\overline{G}$.

3. *In fact, the group operation $\oplus : \overline{G} \times \overline{G} \to \overline{G}$ can be expressed in terms of this action:*

$$(a,\sigma) \oplus (b,\tau) = \left(a \cdot b^{\sigma} \cdot c(\sigma,\tau), \sigma + \tau\right), \tag{4.11}$$

where $c : G \times G \to A$ must satisfy the following condition (since the group operation $\oplus$ is associative):

$$c(\sigma,\tau) \cdot c(\sigma+\tau,\rho) = c(\tau,\rho)^{\sigma} \cdot c(\sigma,\tau+\rho). \tag{4.12}$$

A function c satisfying (4.12) is called a 2-cocycle on G with values in A, and the set of all such cocycles is denoted $Z^2(G,A)$.

[6] Remark that in order to be consistent with the litterature, we chose here to follow (for the most part) the usual notation for group extensions (for example, greek letters no longer represent functions, but rather elements of a group G). The main exception to the rule being that we will use the multiplicative (respectively additive) notation for the group A (respectively G) in order to be coherent with the concrete applications we have in mind.

4. Finally, c can be written in terms of s as

$$c(\sigma,\tau) = s(\sigma) \oplus s(\tau) \ominus s(\sigma+\tau).$$

Despite this detailed and cumbersome notation, equation (4.11) really stands out. Indeed, the sharp eye of the cryptographer will probably have noticed right from the start that

$$n(a,\sigma) := \underbrace{(a,\sigma) \oplus (a,\sigma) \oplus ... \oplus (a,\sigma)}_{n \text{ times}} = (*, n\sigma) \tag{4.13}$$

implies that the discrete logarithm problem on G and $\overline{G}$ are related. Before we can say more, we need to derive a few more (easy) properties of $\overline{G}$. First remark that taking $\tau = \rho = 0$ in (4.12) yields the pretty identity

$$c(0,0)^{\sigma} = c(\sigma,0). \tag{4.14}$$

It is also a routine exercise to verify that the identity element[7] of $\overline{G}$ is $0_{\bar{G}} := (c(0,0)^{-1}, 0)$ and that the inverse of an element $(a,\sigma) \in \overline{G}$ is given by:

$$\left(\left(a^{-\sigma} \cdot c(-\sigma,\sigma) \cdot c(0,0)\right)^{-1}, -\sigma\right).$$

We then have that $n(a,\sigma) = 0_{\overline{G}}$ implies $n\sigma = 0$, from which follows that

The order of σ divides the order of (a,σ),

assuming that (a,σ) has finite order.

Remark 4.8 *Take note that these basic properties will be used in Chapter 5 when we derive an explicit group law algorithm for a specific generalized Jacobian of an elliptic curve.*

4.5 The Algebraic Group $L_{\mathfrak{m}}$

We have seen so far that the generalized Jacobian $J_{\mathfrak{m}}$, with respect to $\mathfrak{m} = \sum_{P\in C} m_P(P)$ of support $S_{\mathfrak{m}}$, is an extension of the usual Jacobian J by $L_{\mathfrak{m}}$, the kernel of $\tau : J_{\mathfrak{m}} \twoheadrightarrow J$. It then followed that the elements of $J_{\mathfrak{m}}$ could be seen as pairs (k,P), where $k \in L_{\mathfrak{m}}$ and $P \in J$. Using this representation, the group law on $J_{\mathfrak{m}}$ could be expressed in terms of the group laws on $L_{\mathfrak{m}}$ and on J, and also involved a 2-cocycle on J with values in $L_{\mathfrak{m}}$. In addition, we already know efficient algorithms to compute in the Jacobian of a suitably chosen curves, such as an elliptic or an hyperelliptic curve. And at last, we now turn our attention to the mysterious group $L_{\mathfrak{m}}$.

[7] Notice that the identity of $\overline{G}$ is not necessarily $(1,0)$, as one might first suspect.

Recall that the map $\tau : J_{\mathfrak{m}} \twoheadrightarrow J$ was defined to be the composition $\varphi \circ \sigma \circ \psi^{-1}$, where both φ and ψ are isomorphisms:

$$J_{\mathfrak{m}} \xrightarrow{\psi^{-1}} \operatorname{Pic}{}^0_{\mathfrak{m}}(C) \xrightarrow{\sigma} \operatorname{Pic}{}^0(C) \xrightarrow{\varphi} J.$$

Consequently, $L_{\mathfrak{m}}$ is isomorphic to $\ker(\sigma)$, where σ simply sent the $\mathfrak{m}$-equivalence class $[D]_{\mathfrak{m}}$ of a divisor D to its divisor class $[D]$ under linear equivalence. Hence, $\sigma([D]_{\mathfrak{m}}) = [\mathbf{0}]$ if and only if D is a principal divisor prime to $S_{\mathfrak{m}}$. That is, there is a $f \in K(C)^*$ such that $D = \operatorname{div}(f)$ and $\operatorname{ord}_P(f) = 0$ for each $P \in S_{\mathfrak{m}}$. Notice that this latter condition means that f is a unit (i.e. is invertible) at every point of $S_{\mathfrak{m}}$. We therefore know that

$$[D]_{\mathfrak{m}} \in \ker(\sigma) \text{ iff } \exists f \in K(C)^* \text{ such that } D = \operatorname{div}(f) \text{ and } f \text{ is invertible at each } P \in S_{\mathfrak{m}}.$$

We would then like to have a representative for each $\mathfrak{m}$-equivalence class comprised of principal divisors. Notice that since $\operatorname{div}(f)$ determines f up to a (nonzero) constant factor, then we can just as well express a representative as a function.

Recall that $\operatorname{Pic}{}^0_{\mathfrak{m}}(C) = \operatorname{Div}{}^0_{\mathfrak{m}}(C)/\operatorname{Princ}_{\mathfrak{m}}(C)$, where $\operatorname{Princ}_{\mathfrak{m}}(C) = \{\operatorname{div}(f) | f \in K(C)^* \text{ and } f \equiv 1 \bmod \mathfrak{m}\}$. Therefore, two divisors will be $\mathfrak{m}$-equivalent if and only if they differ by an element of $\operatorname{Princ}_{\mathfrak{m}}(C)$. Let now f be any representative of the class $[\operatorname{div}(f)]_{\mathfrak{m}}$. So any given element of this class can be expressed as $\operatorname{div}(f \cdot h)$, for some $h \equiv 1 \bmod \mathfrak{m}$.

Fix a point $P \in S_{\mathfrak{m}}$ and let t be a uniformizer for C at P [Sil86, p.22] (that is, an element of $\overline{K}(C)$ satisfying $\operatorname{ord}_P(t) = 1$). Since h satisfies $\operatorname{ord}_P(h-1) \geq m_P$, then $h - 1$ has a zero of order *at least* m_P at P. Thus, h can be expressed as the formal series

$$h - 1 = a_{m_P} t^{m_P} + a_{m_P+1} t^{m_P+1} + a_{m_P+2} t^{m_P+2} + \dots .$$

Hence,

$$f \cdot h = f + f \cdot (a_{m_P} t^{m_P} + a_{m_P+1} t^{m_P+1} + a_{m_P+2} t^{m_P+2} + \dots),$$

where $\deg(f \cdot (h-1)) \geq m_P$ since $\operatorname{ord}_P(f) = 0$. Thus, we may assume without loss of generality that f has the form $f = a_0 + a_1 t + \dots + a_{m_P - 1} t^{m_P - 1}$, where $a_0 \neq 0$. It will however be more convenient to write f as

$$f = a_0 \cdot (1 + a_1' t + \dots + a_{m_P-1}' t^{m_P - 1}), \text{ where } a_0 \neq 0.$$

For example, if $m_P = 1$, each representative consist of a nonzero constant and we therefore recover a copy of the multiplicative group $\mathbb{G}_{\mathrm{m}}$. When $m_P = 2$,

$$[\operatorname{div}(a(1+bt))]_{\mathfrak{m}} + [\operatorname{div}(c(1+dt))]_{\mathfrak{m}} = [\operatorname{div}(ac(1+(b+d)t))]_{\mathfrak{m}},$$

and so a copy of both $\mathbb{G}_{\mathrm{m}}$ and $\mathbb{G}_{\mathrm{a}}$ are involved. In general, notice that if

$$f_1 = b_0 \cdot \left(1 + b_1 t + ... + b_{m_P - 1} t^{m_P - 1}\right) \text{ and } f_2 = c_0 \cdot \left(1 + c_1 t + ... + c_{m_P - 1} t^{m_P - 1}\right),$$

then the desired representative for $[\mathrm{div}(f_1 f_2)]_{\mathfrak{m}}$ can just as well be computed via the following matrix multiplication:

$$\begin{pmatrix} 1 & b_1 & b_2 & b_3 & \dots & b_{m_P-1} \\ 0 & 1 & b_1 & b_2 & \dots & b_{m_P-2} \\ 0 & 0 & 1 & b_1 & \dots & b_{m_P-3} \\ 0 & 0 & 0 & 1 & \dots & b_{m_P-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & c_1 & c_2 & c_3 & \dots & c_{m_P-1} \\ 0 & 1 & c_1 & c_2 & \dots & c_{m_P-2} \\ 0 & 0 & 1 & c_1 & \dots & c_{m_P-3} \\ 0 & 0 & 0 & 1 & \dots & c_{m_P-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Of course, the above semi formal discussion concerning $L_{\mathfrak{m}}$ is far from providing an actual proof of the following theorem of Rosenlicht [Ros54], but somehow at least captures the underlying ideas. The complete details can be found in [Ser88], Sections 13 to 17 of Chapter V.

Theorem 4.9 (Rosenlicht) *Let C be a smooth algebraic curve defined over an algebraically closed field, J be the Jacobian of C and $J_{\mathfrak{m}}$ be the generalized Jacobian of C with respect to a modulus $\mathfrak{m} = \sum_{P \in C} m_P(P)$ of support $S_{\mathfrak{m}}$. Let also $L_{\mathfrak{m}}$ be the kernel of the natural homomorphism τ from $J_{\mathfrak{m}}$ onto J. Then, $L_{\mathfrak{m}}$ is an algebraic group isomorphic to the product of a torus $T = (\mathbb{G}_m)^{\# S_{\mathfrak{m}} - 1}$ by a unipotent group V of the form*

$$V = \prod_{P \in S_{\mathfrak{m}}} V_{(m_P)},$$

where each $V_{(m_P)}$ is isomorphic to the group of matrices of the form:

$$\begin{pmatrix} 1 & a_1 & a_2 & a_3 & \dots & a_{m_P-1} \\ 0 & 1 & a_1 & a_2 & \dots & a_{m_P-2} \\ 0 & 0 & 1 & a_1 & \dots & a_{m_P-3} \\ 0 & 0 & 0 & 1 & \dots & a_{m_P-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

Remark 4.10 *Notice that since $L_{\mathfrak{m}}$, $J_{\mathfrak{m}}$ and J are all algebraic groups, then we can say that $J_{\mathfrak{m}}$ is in fact an extension as algebraic groups of J by $L_{\mathfrak{m}}$. Algebraic group extensions and their principal properties are discussed in Chapter VII of [Ser88].*

If we are in the situation where $\mathfrak{m} = (P_0) + (P_1) + ... + (P_r)$ with the P_i's distinct, then $L_{\mathfrak{m}}$ is isomorphic to a torus T of dimension r. Moreover, since the usual Jacobian of $\mathbb{P}^1$ is trivial (c.f. Section 3.91), it then follows that the generalized Jacobian of $\mathbb{P}^1$ with respect to $\mathfrak{m}$ will be isomorphic to T. As a result, algebraic tori of any dimension can be seen as generalized Jacobians. Algebraic tori over a finite field have interesting cryptographic properties, as demonstrated by Rubin and Silverberg [RS03, RS04a]. We will come back to this in Section 4.6.

The complete opposite situation would be to consider a module of the form $\mathfrak{m} = m(P)$. Then the group $L_{\mathfrak{m}}$ will be isomorphic to $V_{(m)}$. Observe that the discrete logarithm problem on $V_{(m)}$ *alone* is easy since

$$\begin{pmatrix} 1 & a_1 & a_2 & a_3 & \dots & a_{n-1} \\ 0 & 1 & a_1 & a_2 & \dots & a_{n-2} \\ 0 & 0 & 1 & a_1 & \dots & a_{n-3} \\ 0 & 0 & 0 & 1 & \dots & a_{n-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}^n = \begin{pmatrix} 1 & na_1 & * & * & \dots & * \\ 0 & 1 & na_1 & * & \dots & * \\ 0 & 0 & 1 & na_1 & \dots & * \\ 0 & 0 & 0 & 1 & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix},$$

and therefore $n = (na_1)a_1^{-1}$. However, according to (4.11), the group law on the generalized Jacobian will be given by

$$(M_1, P_1) + (M_2, P_2) = (M_1 M_2 \cdot c(P_1, P_2), P_1 + P_2)\,,$$

where $M_1,\ M_2 \in V_{(m)}$ and $P_1,\ P_2 \in J$. We therefore obtain that

$$n(M, P) = (M^n \cdot \mu, nP)\,,$$

where the value of μ equals $c(P, P) \cdot c(P, 2P) \cdot \ldots \cdot c(P, (n-1)P)$. Remark that μ can be seen as a mask hiding the value of the DL in the unipotent group. Of course in practice the value of μ can (and should!) be computed differently. But the point here is that μ depends on P and is independent of M. As a result, if it is computationnaly infeasible to compute μ given P and $n(M, P)$ (but *not* n), then the value of M^n will be just as hard to compute (the role of μ here is to *'mimic a one-time pad'*). Therefore, even if the discrete logarithm problem on $L_{\mathfrak{m}}$ is trivial, it does not necessarily implies that the corresponding problem on the generalized Jacobian will be easy. This simple example shows that a deeper analysis of the behavior of these *'masks'*, based on the specific 2-cocycle appearing in the group law algorithm, should be done for the specific generalized Jacobians that one wishes to use for cryptographic purposes.

4.5.1 A Concrete (and Easy) Example

Time for some hands-on practice. The goal of the following exercise is to work directly on divisors, from the definition of $\mathfrak{m}$-equivalence alone in order to recover the structure of the algebraic group $L_{\mathfrak{m}}$ introduced in Theorem 4.9. We will work in the simple case where the curve is the projective line $\mathbb{P}^1$ and the modulus is of the form $\mathfrak{m} = (L) + (M) + (N)$, where $L = (X_L : 1)$, $M = (X_M : 1)$, $N = (X_N : 1) \in \mathbb{P}^1$ are distinct points. Of course, it will be strictly forbidden to cheat and work backwards from the conclusion of the theorem: we should instead try to forget all we know so far about $L_{\mathfrak{m}}$ and let the *'mathemagic'* operate.

First recall that all degree zero divisors of $\mathbb{P}^1$ are principal (see Example 3.75). We are therefore considering $\mathfrak{m}$-equivalence on divisors of the form

$$D = \operatorname{div}(g), \text{ where } \operatorname{ord}_L(g) = \operatorname{ord}_M(g) = \operatorname{ord}_N(g) = 0.$$

So let D_1 and D_2 be two such divisors, with $D_1 = \operatorname{div}(f_1)$ and $D_2 = \operatorname{div}(f_2)$. From the very definition of $\mathfrak{m}$-equivalence, we have that

$$\begin{aligned}
D_1 \sim_{\mathfrak{m}} D_2 \quad & \text{iff} \quad \exists f \in K(\mathbb{P}^1)^* \text{ such that } \operatorname{div}(f) = D_1 - D_2 \text{ and } f \equiv 1 \bmod \mathfrak{m}, \\
& \text{iff} \quad \exists f \in K(\mathbb{P}^1)^* \text{ such that } \operatorname{div}(f) = D_1 - D_2 \text{ and} \\
& \qquad \operatorname{ord}_L(1-f) \geq 1,\ \operatorname{ord}_M(1-f) \geq 1,\ \operatorname{ord}_N(1-f) \geq 1, \\
& \text{iff} \quad \exists f \in K(\mathbb{P}^1)^* \text{ such that } \operatorname{div}(f) = \operatorname{div}\left(\frac{f_1}{f_2}\right) \text{ and } f(L) = f(M) = f(N) = 1, \\
& \text{iff} \quad \exists c \in K^* \text{ such that } \frac{f_1(L)}{f_2(L)} = \frac{f_1(M)}{f_2(M)} = \frac{f_1(N)}{f_2(N)} = \frac{1}{c}, \\
& \text{iff} \quad \frac{f_1(L)}{f_2(L)} = \frac{f_1(M)}{f_2(M)} = \frac{f_1(N)}{f_2(N)}, \\
& \text{iff} \quad \frac{f_1(L)}{f_1(M)} = \frac{f_2(L)}{f_2(M)} \text{ and } \frac{f_1(M)}{f_1(N)} = \frac{f_2(M)}{f_2(N)}.
\end{aligned}$$

It then follows that the map

$$\begin{aligned}
\psi : \quad \operatorname{Pic}^0_{\mathfrak{m}}(\mathbb{P}^1) & \longrightarrow \quad \mathbb{G}_{\mathrm{m}} \times \mathbb{G}_{\mathrm{m}} \\
[\operatorname{div}(f)]_{\mathfrak{m}} & \longmapsto \quad \left(\frac{f(L)}{f(M)}, \frac{f(M)}{f(N)}\right)
\end{aligned}$$

is well-defined and injective. As for surjectivity, let (a, b) be a given element of $\mathbb{G}_{\mathrm{m}} \times \mathbb{G}_{\mathrm{m}}$ for which we need to find a function $f \in K(\mathbb{P}^1)^*$ satisfying

$$\operatorname{ord}_L(f) = \operatorname{ord}_M(f) = \operatorname{ord}_N(f) = 0,\ \frac{f(L)}{f(M)} = a \text{ and } \frac{f(M)}{f(N)} = b.$$

To do so, we use the technique underlying the interpolation polynomial of Lagrange[8] in order

[8] To be accurate, we should point out that it was first discovered by Edward Waring in 1779, then rediscovered by Leonhard Euler in 1783 and finally by Joseph Louis Lagrange in 1795.

to set $f(X,Y)$ equal to

$$\frac{a\,(X-X_MY)\,(X-X_NY)}{(X_L-X_M)\,(X_L-X_N)\,Y^2}+\frac{(X-X_LY)\,(X-X_NY)}{(X_M-X_L)\,(X_M-X_N)\,Y^2}+\frac{(X-X_LY)\,(X-X_MY)}{b\,(X_N-X_L)\,(X_N-X_M)\,Y^2}.$$

Then, $f(L)=a$, $f(M)=1$ and $f(N)=b^{-1}$, which yields that $\psi\left([\operatorname{div}(f)]_{\mathfrak{m}}\right)=(a,b)$, as wanted. Thus, we have shown that $\psi:\operatorname{Pic}^0_{\mathfrak{m}}(\mathbb{P}^1)\to\mathbb{G}_{\mathrm{m}}\times\mathbb{G}_{\mathrm{m}}$ is a well-defined bijection of sets.

We are now ready to describe the group law in terms of this representation. Again, let $D_1=\operatorname{div}(f_1)$, $D_2=\operatorname{div}(f_2)$ be two divisors prime to $S_{\mathfrak{m}}$ and let

$$a_1=\frac{f_1(L)}{f_1(M)},\ b_1=\frac{f_1(M)}{f_1(N)}\text{ and }a_2=\frac{f_2(L)}{f_2(M)},\ b_2=\frac{f_2(M)}{f_2(N)}.$$

That is, D_1 and D_2 respectively correspond to (a_1,b_1) and to (a_2,b_2). We now want to know the element of $\mathbb{G}_{\mathrm{m}}\times\mathbb{G}_{\mathrm{m}}$ corresponding to D_1+D_2. But this is easy since

$$D_1+D_2=\operatorname{div}(f_1)+\operatorname{div}(f_2)=\operatorname{div}(f_1\cdot f_2),$$

and we just have to let $f_3:=f_1\cdot f_2$, then write down

$$\frac{f_3(L)}{f_3(M)}=\frac{f_1(L)\cdot f_2(L)}{f_1(M)\cdot f_2(M)}=a_1\cdot a_2\text{ and }\frac{f_3(M)}{f_3(N)}=\frac{f_1(M)\cdot f_2(M)}{f_1(N)\cdot f_2(N)}=b_1\cdot b_2$$

in order to conclude that D_1+D_2 is associated to $(a_1\cdot a_2,b_1\cdot b_2)$. We have thus recovered the torus of dimension 2 stated in Theorem 4.9.

4.6 Cryptosystems Falling in the Spectrum of Generalized Jacobians

We conclude this chapter by providing a perspective as to where cryptography based on generalized Jacobians actually *'fits'* within the numerous public-key protocols proposed to this date. This global picture will serve two purposes. First, it demonstrates that several of the most popular PKC based on discrete logarithms can be interpreted in the language of generalized Jacobians. This fundamental observation shows that seemingly unrelated structures can in fact be seen as realizations of the same mathematical object. Second, this unified approach further motivates the hunt for generalized Jacobians (where neither $L_{\mathfrak{m}}$ nor J are trivial) suitable for cryptographic applications[9].

Among the groups utilized in DL-based cryptography mentionned in Section 2.6, it turns out that the multiplicative group of a finite field, the elliptic curves, the Jacobian of hyperelliptic

[9] The hunt opens next chapter where our prey will be a generalized Jacobian of an elliptic curve.

curves and the algebraic tori can all be seen as generalized Jacobians. Moreover, two more cryptosystems whose underlying structures do not even form a group, namely LUC and XTR, are also closely related to generalized Jacobians.

Figure 4.5 provides a simplified view of the interrelation between the cryptosystems (on the bottom line) and their underlying structures. A line connecting two elements of the diagram means *'can be interpreted in the language of'*.

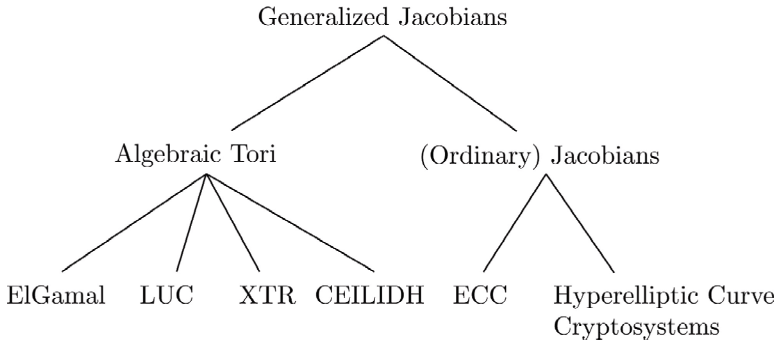


Figure 4.5: Relation between DL-based cryptosystems and generalized Jacobians

This schematic representation clearly highlights the two distinct sub-families of generalized Jacobians that have been used so far: the usual Jacobians and the algebraic tori. Curiously, the specific strengths of each family are somehow complementary. Indeed, the popularity of elliptic curves and Jacobians of low genus hyperelliptic curves is due in part to their resistance to subexponential attacks. On the other hand, algebraic tori (and their quotients) constitute a very neat way to represent elements in a compact form, significantly decreasing the amount of information that needs to be exchanged.

USUAL JACOBIANS. They are the generalized Jacobians corresponding to the case where the linear group $L_\mathfrak{m}$ is trivial. That is, if the modulus $\mathfrak{m} = \sum_{P \in C} m_P(P)$ with support $S_\mathfrak{m}$ was chosen to have degree zero or one. Indeed, if $\mathfrak{m} = \mathbf{0}$, then the condition $f \equiv 1 \bmod \mathfrak{m}$, i.e. $\mathrm{ord}_{P_i}(1-f) \geq m_i$ for each $P_i \in S_\mathfrak{m}$ is vacuously true and therefore, $\mathfrak{m}$-equivalence coincides with linear equivalence. As well, if $\mathfrak{m} = (M)$, then the requirement $f \equiv 1 \bmod \mathfrak{m}$ reduces to $\mathrm{ord}_M(1-f) \geq 1$, which is equivalent to $f(M) = 1$. Hence, $\mathfrak{m}$-equivalence in this case reads $D \sim_\mathfrak{m} D'$ iff $\exists f \in K(C)^*$ such that $\mathrm{div}(f) = D - D'$ and $f(M) = 1$. But since $\mathrm{div}(c \cdot f) = \mathrm{div}(f)$ for any nonzero constant c, the condition $f(M) = 1$ is superfluous. It then follows that when $\mathfrak{m} = (M)$, linear and $\mathfrak{m}$-equivalence also define the same divisor classes.

Recall that the use of Jacobians in cryptography via elliptic curves goes back to 1985

[Mil86c, Kob87] and that since then, it has prompted an impressive amount of research on the cryptographic uses of algebraic curves. From special hardware for hyperelliptic curves to side-channel attacks or pairings[10], it seems that the frenzy surrounding them has not faded in nearly twenty years. This contagious enthusiasm inevitably raised the possibility that other abelian varieties, or more generally algebraic groups, might be of interest for cryptographers (and needless to say, cryptanalysts).

ALGEBRAIC TORI. An algebraic torus T of dimension d is the generalized Jacobian of the projective line $\mathbb{P}^1$ with respect to a modulus $\mathfrak{m} = (P_0) + (P_1) + ... + (P_d)$, where the P_i's are distinct. Indeed, we have seen that every divisor of degree zero on $\mathbb{P}^1$ is principal, and consequently that its Jacobian J is trivial (c.f. Example 3.91). On the other hand, according to Theorem 4.9, we know that $L_\mathfrak{m}$ is isomorphic to the product of $(\mathbb{G}_\mathrm{m})^d$ by the unipotent group V. But since each $m_i = 1$ $(0 \leq i \leq d)$, it is easy to see that V is trivial in this case. Finally, we get that $J_\mathfrak{m}$ has to be isomorphic to T (since it is an extension of J by $L_\mathfrak{m}$).

The most obvious examples of applications of algebraic tori in cryptography are none other than the classical Diffie-Hellman key exchange, together with the ElGamal cryptosystem and signature (respectively covered in Sections 2.3.3, 2.4.3 and 2.5.2). In fact, as soon as the operations of a cryptographic scheme are performed in the multiplicative group of a finite field, we can say that they are based on the simplest torus, namely the multiplicative group $\mathbb{G}_\mathrm{m}$.

However, the first *explicit* use of algebraic tori in cryptography is fairly recent. Recall that the concept of *torus-based cryptography* has been formally introduced by Karl Rubin and Alice Silverberg at CRYPTO 2003 [RS03]. The quality of Silverberg's presentation at this conference was impressing, at all levels, and certainly contributed to give wings to these news ideas. Inspired by conjectures made about XTR by Bosma, Hutton and Verheul at ASIACRYPT 2002 [BHV02], Rubin and Silverberg were not only able to disprove these conjectural statements, but also reinterpreted XTR in terms of tori. In addition, they also showed how the cryptosystem LUC [LS93], based on Lucas functions, could also be reconsidered in the language of tori. That's not all. They set the general framework for torus-based cryptography and gave two explicit cryptosystems, one based on a 1-dimensional torus (corresponding to the case $n = 2$ described below) and another, CEILIDH, which uses a torus of dimension 2 (where $n = 6$).

More precisely, if we let T_n be the algebraic torus of dimension $\varphi(n)$, then the group $T_n(\mathbb{F}_q)$ is finite and can be identified with the cyclic subgroup of $\mathbb{F}_{q^n}^*$ of order $\Phi_n(q)$, where Φ_n is the n-th cyclotomic polynomial [RS03, lemma 7 (i)-(ii)]. In a nutshell, let's just say that $T_n(\mathbb{F}_q)$

[10] For instance, the web site '*Pairing-based Crypto Lounge*' of Paulo Barreto [Bar02] now lists over 200 articles related to the use of pairings in cryptography alone.

can be substituted in cryptographic protocols requiring that the DLP of the underlying group be presumably hard. The (straightforward) torus-based versions of Diffie-Hellman, ElGamal encryption and signature can be found in Section 6 of [RS03].

As for efficiency, the group operation on $T_n(\mathbb{F}_q)$ is simply the usual multiplication inherited from $\mathbb{F}_{q^n}^*$, so that poses no problem. But it raises the question as to what can possibly be the advantage of working in $T_n(\mathbb{F}_q)$ instead of in the whole group. So, as I often ask my students: *'where's the catch?'* This is precisely where it becomes interesting. The secret in fact lies in the way that elements of $T_n(\mathbb{F}_q)$ can be represented: they are just like the tiny umbrellas that can fit in a pocket, but once deployed offer a full size cover. Indeed, for suitably chosen values of n, we know how to represent the elements of $T_n(\mathbb{F}_q)$ in a *compact form*, using only $\varphi(n)$ elements of $\mathbb{F}_q$. For instance, if n is a prime power or a product of two prime powers, then we know that such a compact representation must exist (see the discussion following Voskresenskii's conjecture in [RS03, Section 4, Conjecture 9]). Explicit formulæ for converting from one representation to the other are given for T_2 and T_6 in Section 5 of [RS03]. So in practice, Alice and Bob each perform their computations directly in $T_n(\mathbb{F}_q) \subseteq \mathbb{F}_{q^n}^*$ and simply convert to the compact representation whenever they need to send data to the other party.

The main advantage of CEILIDH over LUC and XTR is that its underlying structure is a good old *group*. Hence, unlike the other two which only possess a natural exponentiation, CEILIDH has full multiplication *and* exponentiation. In fact, the elements exchanged in the LUC cryptosystem correspond to the ones of T_2/S_2, where S_k is the symmetric group on k letters. Even if the quotient variety T_2/S_2 is *not* an algebraic group, exponentiation is still well-defined on this set of equivalence classes, enabling for example to perform a key exchange *'à la Diffie-Hellman'*. Similarly, the system XTR of A. K. Lenstra and E. R. Verheul [LV00, LV01] is based on the variety T_6/S_3, and since exponentiation in T_6 preserves S_3-orbits, it follows that the exponentiation in the quotient is well-defined.

In the light of these observations, it is no longer mysterious as to why LUC and XTR also have the ability of compactly representing their elements. In a nutshell, LUC, XTR and CEILIDH have the discrete log security of $\mathbb{F}_{p^n}^*$, where $n = 2$ for LUC and $n = 6$ for XTR and CEILIDH, while it is possible to represent the elements using only $\varphi(n)$ elements of $\mathbb{F}_p$. In comparison with the classical Diffie-Hellman key exchange, we would have to work a priori with $\mathbb{F}_{p^n}^*$ directly in order to achieve a comparable security level, but then the elements transmitted between Alice and Bob would consist of n elements of $\mathbb{F}_p$. So interesting savings occur as soon as $\varphi(n)$ is rather small compared to n. If we consider the ratio of the number of bits of security to the number of bits transmitted, we therefore obtain a standard of measure of 1 for Diffie-Hellman, and a quotient of $2\log p\,/(\varphi(2)\log p)\; = 2$ for LUC and of $6\log p\,/(\varphi(6)\log p)\; = 3$ for XTR and CEILIDH.

That concludes our brief overview of torus-based cryptography. More details and recent advancements on the work of Rubin and Silverberg can be found in [RS04a], [RS04b], [RS04c], [DW04], [vDGP$^+$05], and [GV05].

To sum up, we are currently using two distinct types of generalized Jacobians in cryptography: the Jacobians (corresponding to trivial $L_\mathfrak{m}$) and the algebraic tori (for which J is now trivial). Hence, we know that *seperately*, both Jacobians and algebraic tori are great choices for DL-based cryptography. Standing right here, it seems now so obvious that the natural thing to do next is to consider a generalized Jacobian for which neither J nor $L_\mathfrak{m}$ is trivial. The goal we are after is to come up with sufficient evidences to confidently answer the following yes/no question:

*Can generalized Jacobians with nontrivial J **and** $L_\mathfrak{m}$ be used for cryptographic purposes?*

There are, as usual, two hidden requirements behind this question: the efficiency and the security aspects. And as in court, what we need is *one* good witness with competitive security and efficiency to win our case. The next step is to find a potentially good witness. Since we here venture in an unexplored territory, we are therefore free to choose a really simple case of study (and then hopefully simplify the analysis).

For the curves we wish to consider, the two natural candidates are elliptic curves and hyperelliptic curves. They are equally interesting candidates from our point of view, but unfortunately a cruel choice must be made here[11]. Given that this curve will be our spokesperson for these new ideas and given that ECC is (to this date) considered in the community as *'**the** alternative to RSA'*, we are therefore opting for elliptic curves. Lastly, we have to decide upon a modulus $\mathfrak{m}$ to use. Thanks to corollary 4.6, we know that once we have fixed a smooth elliptic curve E over a finite field $\mathbb{F}_q$, then it suffices to choose $\mathfrak{m}$ such that $\deg(\mathfrak{m}) \geq 2$ in order to guarantee that $J_\mathfrak{m}$ will not be a trivial direct product. In the simplest case, $\mathfrak{m} = (M) + (N)$ with distinct M and N in $E(\overline{\mathbb{F}}_q)$. Remark that we want to assume that $M \neq N$, since otherwise Theorem 4.9 tells us that $L_\mathfrak{m}$ is isomorphic to the additive group $\mathbb{G}_\mathrm{a}$, for which the DLP is really easy. Luckily, when $M \neq N$, that same theorem ensures that $L_\mathfrak{m}$ will be isomorphic to $\mathbb{G}_\mathrm{m}$. This is just perfect since the generalized Jacobians we get will then be a mixture of two well-studied cryptographic structures: elliptic curves and finite fields. So after all, the choice of a witness

[11] See Chapter 6 where further work is discussed.

was quite natural: E for popularity and $\mathfrak{m}$ for simplicity. And as previously advertised, this case study will be fully investigated in the next chapter.

Chapter 5

A Concrete Cryptosystem

"It is possible to write endlessly
on elliptic curves (this is not a threat)."

- Serge Lang

Cryptographers like finite fields because of their efficiency and care about elliptic curves for their security. Unfortunately, this dichotomy appears ineluctable: when comes the time to choose a group to implement a DL-based protocol, it seems that there is room for only one of them. So it sounds like we cannot have the best of both worlds... But before giving up too easily, let's recall a few facts for the record:

- *Elliptic curves are their own Jacobians*
- *Generalized Jacobians are extensions of a Jacobian by a linear group*
- *For suitably chosen moduli, this linear group coincides with* $\mathbb{G}_{\mathrm{m}}$.

So if we consider a generalized Jacobian $J_{\mathfrak{m}}$ which is a nontrivial extension of E by $\mathbb{G}_{\mathrm{m}}$, then we can naively picture $J_{\mathfrak{m}}(\mathbb{F}_q)$ as being an elliptic curve *'intertwined'* with a finite field, just like a ringwire puzzle[1] where two pieces of metal are interlaced. In comparison, a direct product would then correspond to a mere juxtaposition of the two parts.

Keeping this image in mind, we now have some serious work ahead of us before we can claim that this particular generalized Jacobian is an interesting candidate to consider for practical applications. Indeed, recall that the main requirements for a group G to be suitable for cryptography are that

[1] Which is sometimes also called a *'disentanglement puzzle'*.

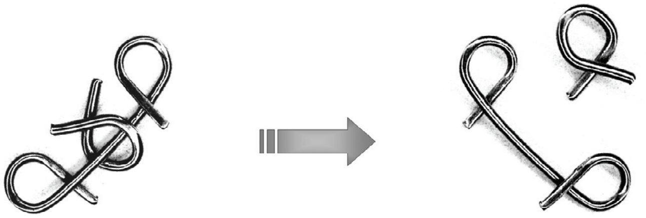


Figure 5.1: A ringwire puzzle: unsolved (left) and solved (right).

- *The elements of G can be easily represented in a compact form,*
- *The group operation can be performed efficiently,*
- *The discrete logarithm problem in G is believed to be intractable, and*
- *The group order can be efficiently computed.*

Ensuring that these requirements are fulfilled is the exciting program of this chapter. Once this is achieved, we could then right away use this generalized Jacobian as the underlying group of the (generalized) ElGamal cryptosystem, for instance. Thus all the work resides in showing that the above four properties hold. In the end, we will also have to keep in mind that the overall appreciation also has to take into account the relative performance obtained compared to other popular cryptosystems.

5.1 Initial Setup

This short section contains the global description of the generalized Jacobians that will be studied in this chapter, together with important reminders. It is also the time to make a few conventions in order to ease the exposition.

First recall that by Corollary 4.6, the simplest case where the generalized Jacobian is not a direct product arise when the curve we consider has genus one and the modulus has degree 2. So throughout this chapter, we will work with a smooth elliptic curve E defined over a finite field $K = \mathbb{F}_q$. For the purpose of constructing the generalized Jacobian, we will view E as being defined over $\overline{\mathbb{F}}_q$, so that the results of Chapter 4 directly apply here.

We now need to fix a modulus $\mathfrak{m} = (M) + (N)$, where M and N are points of $E(\overline{\mathbb{F}}_q)$. Remark that for the applications we have in mind, like the generalized ElGamal cryptosystem and signature, we need the group $J_\mathfrak{m}$ to be publicly known, so M and N are assumed to be public parameters. Also notice that in practice, we will be free to *select* M and N, so that for a given elliptic curve, there are in fact many possible moduli to choose from.

Now, since we ultimately want to *'intertwine'* E with the multiplicative group of a finite field[2] $\mathbb{F}_r$, the generalized Jacobian we consider should be an extension of E by $\mathbb{G}_{\mathrm{m}}$. Nothing easier since having $L_{\mathfrak{m}}$ isomorphic to $\mathbb{G}_{\mathrm{m}}$ is guaranteed by Theorem 4.9 as soon as $M \neq N$. One more thing: we will have to use the correspondence between $\mathrm{Pic}^{\,0}_{\mathfrak{m}}(E)$ and $J_{\mathfrak{m}}$ in order to *'transport'* the group law on divisors to $\mathbb{G}_{\mathrm{m}} \times E$, which means that we will certainly rely on the known group isomorphism

$$\begin{array}{rcl} E & \to & \mathrm{Pic}^{\,0}(E) \\ P & \mapsto & (P) - (\mathcal{O}) + \mathrm{Princ}(E) \end{array} \tag{5.1}$$

given in Proposition 3.90. Now, since $\mathfrak{m}$-equivalence is defined on divisors whose support is disjoint from $\{M, N\}$, we won't be able to use (5.1) directly, *unless* M, $N \neq \mathcal{O}$. So to make our lives a little easier, we will thereafter assume that condition M, $N \neq \mathcal{O}$ is also fulfilled. Hence, we can let $M = (X_M : Y_M : 1)$ and $N = (X_N : Y_N : 1)$. These are so far the only conditions we impose on $\mathfrak{m}$.

Lastly, let's establish two small conventions that will also contribute to simplify our lives. First, we know by Theorem 4.7 that there is a bijection of sets between $J_{\mathfrak{m}}$ and $\mathbb{G}_{\mathrm{m}} \times E$, so by an *'element of $J_{\mathfrak{m}}$'*, we will thereafter mean a pair (k, P), where $k \in \mathbb{G}_{\mathrm{m}}$ and $P \in E$. Also, once an explicit bijection between $\mathrm{Pic}^{\,0}_{\mathfrak{m}}(E)$ and $\mathbb{G}_{\mathrm{m}} \times E$ will be fixed, by *'the group law on $J_{\mathfrak{m}}$'*, it will be understood *'the group operation on $\mathbb{G}_{\mathrm{m}} \times E$ induced from the addition on $\mathrm{Pic}^{\,0}_{\mathfrak{m}}(E)$ through this particular bijection'*.

5.2 Explicit Bijection between $\mathrm{Pic}^{\,0}_{\mathfrak{m}}(E)$ and $\mathbb{G}_m \times E$

In the preceding section, we chose a tailor-made modulus that guaranteed the existence of a bijection of sets $\psi : \mathrm{Pic}^{\,0}_{\mathfrak{m}}(E) \to \mathbb{G}_{\mathrm{m}} \times E$. So we already know that the elements of our generalized Jacobian can be conveniently represented as pairs (k, P), where $k \in \mathbb{G}_{\mathrm{m}}$ and $P \in E$. The next step is to make this bijection explicit. Although the mere existence of ψ suffices to compactly represent the elements of $J_{\mathfrak{m}}$, understanding this correspondence in depth will prove to be useful in the next section when comes the time to work out explicit formulæ for the group operation on $\mathbb{G}_{\mathrm{m}} \times E$. Indeed, given (k_1, P_1) and $(k_2, P_2) \in J_{\mathfrak{m}}$, we will have to compute their sum $(k_3, P_3) \in J_{\mathfrak{m}}$. Hence if we know that (k_1, P_1) and (k_2, P_2) respectively correspond to the

[2] It will be possible to see how q and r are related once we have determined the group law in Section 5.3.

$\mathfrak{m}$-equivalence class of D_1 and D_2, then (k_3, P_3) must correspond to the $\mathfrak{m}$-equivalence class of $D_1 + D_2$. That is,

If $\psi\left([D_1]_{\mathfrak{m}}\right) = (k_1, P_1)$ *and* $\psi\left([D_2]_{\mathfrak{m}}\right) = (k_2, P_2)$, *then* $\psi\left([D_1 + D_2]_{\mathfrak{m}}\right) = (k_3, P_3)$.

Hence, exploring ψ can be seen as the first step towards the obtention of the group law algorithm on $\mathbb{G}_{\mathrm{m}} \times E$.

We are now ready to begin our investigation. As mentioned earlier, we already possess a group isomorphism between $\mathrm{Pic}^0(E)$ and E, so this will be our official starting point. Under this isomorphism, recall that the class of a divisor $D = \sum_{P \in E} n_P(P) \in \mathrm{Div}^0(E)$ is mapped to the sum $S = \sum_{P \in E} n_P P \in E$. By Abel's theorem (Theorem 3.84), there is then an $f \in \overline{K}(E)^*$ such that

$$D = (S) - (\mathcal{O}) + \mathrm{div}(f). \tag{5.2}$$

Notice that if D has disjoint support with $\mathfrak{m}$, then either $S \neq M$, N and $\mathrm{ord}_M(f) = \mathrm{ord}_N(f) = 0$, or else $S \in \{M, N\}$ and $\mathrm{ord}_S(f) = -1$. This latter case is undesirable here since we might need to evaluate f at both M and N, just as we did in the example on $\mathbb{P}^1$ of Section 4.5.1. Hence, if $S \neq M$, N, then we can keep equation (5.2) as is. Otherwise, Abel's theorem will once more come to the rescue: the idea is to use, in place of $(S) - (\mathcal{O})$, another simple divisor linearly equivalent to D which will now have disjoint support with $\mathfrak{m}$. Concretely, observe that if we *translate* S by a point $T \in E$, we obtain

$$D \sim (S) - (\mathcal{O}) \sim (S + T) - (T),$$

and thus if $T \notin \{\mathcal{O}, M, N, M - N, N - M\}$, then both $(M + T) - (T)$ and $(N + T) - (T)$ have disjoint support with $\mathfrak{m}$. So from now on, we will assume that such a *'translation point'* T is fixed and publicly known. We can now let

$$R = \begin{cases} \mathcal{O} & \text{if } S \notin \{M, N\}, \\ T & \text{otherwise,} \end{cases}$$

and so there is an $f \in \overline{K}(E)^*$ satisfying

$$D = (S + R) - (R) + \mathrm{div}(f), \tag{5.3}$$

where the property $\mathrm{ord}_M(f) = \mathrm{ord}_N(f) = 0$ is fulfilled as soon as D has disjoint support with $\mathfrak{m}$. Remark that this way of writing out a divisor highlights the point S of E corresponding to D, so it remains to determine how to *'read'* the corresponding element of $\mathbb{G}_{\mathrm{m}}$ from (5.3). This is what we undertake now.

Since any two divisors in an $\mathfrak{m}$-equivalence class are mapped to the same element of $\mathbb{G}_\mathrm{m} \times E$, our approach will be to unravel the definition of $\mathfrak{m}$-equivalence until we can clearly see how to associate an element of $\mathbb{G}_\mathrm{m} \times E$ to each class. So let $D_1 = (S_1 + R_1) - (R_1) + \mathrm{div}(f_1)$, $D_2 = (S_2 + R_2) - (R_2) + \mathrm{div}(f_2) \in \mathrm{Div}^0_\mathfrak{m}(E)$ be given such that

$$R_i = \begin{cases} \mathcal{O} & \text{if } S_i \notin \{M, N\}, \\ T & \text{otherwise,} \end{cases}$$

for $i = 1, 2$. We then have

$$\begin{aligned}
D_1 \sim_\mathfrak{m} D_2 \quad & \text{iff} \quad \exists f \in \overline{K}(E)^* \text{ such that } \mathrm{div}(f) = D_1 - D_2 \text{ and } f \equiv 1 \bmod \mathfrak{m}, \\
& \text{iff} \quad \exists f \in \overline{K}(E)^* \text{ such that } \mathrm{div}(f) = (S_1 + R_1) - (S_2 + R_2) + (R_2) - (R_1) \\
& \qquad + \mathrm{div}\left(\frac{f_1}{f_2}\right) \text{ and } \mathrm{ord}_M(1 - f) \geq 1,\ \mathrm{ord}_N(1 - f) \geq 1, \\
& \text{iff} \quad S_1 + R_1 - (S_2 + R_2) + R_2 - R_1 = \mathcal{O} \text{ and } \exists f \in \overline{K}(E)^* \text{ such that} \\
& \qquad \mathrm{div}(f) = \mathrm{div}\left(\frac{f_1}{f_2}\right) \text{ and } f(M) = f(N) = 1, \\
& \text{iff} \quad S_1 = S_2,\ R_1 = R_2 \text{ and } \exists c \in \overline{K}^* \text{ such that } \frac{f_1(M)}{f_2(M)} = \frac{f_1(N)}{f_2(N)} = \frac{1}{c}, \\
& \text{iff} \quad S_1 = S_2 \text{ and } \frac{f_1(M)}{f_2(M)} = \frac{f_1(N)}{f_2(N)}, \\
& \text{iff} \quad S_1 = S_2 \text{ and } \frac{f_1(M)}{f_1(N)} = \frac{f_2(M)}{f_2(N)}.
\end{aligned}$$

That means that in order to check whether two given divisors are $\mathfrak{m}$-equivalent, we simply have to test two equalities, one in E and one in $\mathbb{G}_\mathrm{m}$. The obvious candidate for ψ is thus the map

$$\begin{array}{rccc} \psi : & \mathrm{Pic}^0_\mathfrak{m}(E) & \longrightarrow & \mathbb{G}_\mathrm{m} \times E \\ & [D]_\mathfrak{m} & \longmapsto & (k, S), \end{array}$$

such that the $\mathfrak{m}$-equivalence class of $D = \sum_{P \in E} n_P(P) \in \mathrm{Div}^0_\mathfrak{m}(E)$ corresponds to $S = \sum_{P \in E} n_P P$ and $k = f(M)/f(N)$, where $f \in \overline{K}(E)^*$ is any function satisfying

$$\mathrm{div}(f) = \begin{cases} D - (S) + (\mathcal{O}) & \text{if } S \notin \{M, N\}, \\ D - (S + T) + (T) & \text{otherwise.} \end{cases}$$

Notice that the existence of f is guaranteed by Abel's theorem (c.f. Theorem 3.84) and that ψ is well-defined since we have just shown that for $D_1 = (S_1 + R_1) - (R_1) + \mathrm{div}(f_1)$, $D_2 = (S_2 + R_2) - (R_2) + \mathrm{div}(f_2)$, $k_1 = f_1(M)/f_1(N)$ and $k_2 = f_2(M)/f_2(N)$, we have:

$$[D_1]_\mathfrak{m} = [D_2]_\mathfrak{m} \textit{ implies that } k_1 = k_2 \textit{ and } S_1 = S_2.$$

Moreover, ψ is injective since we also already know that

$$(k_1, S_1) = (k_2, S_2) \textit{ implies that } [D_1]_{\mathfrak{m}} = [D_2]_{\mathfrak{m}}.$$

It therefore remains to show that ψ is surjective as well. So given $(k, S) \in \mathbb{G}_{\mathrm{m}} \times E$, we must find an $f \in \overline{K}(E)^*$ such that $f(M)/f(N) = k$. Using the idea behind the interpolation polynomial of Lagrange, or simply by inspection, we easily see that

$$f(X,Y,Z) = \begin{cases} \dfrac{k\,(X - X_N Z) + (X_M Z - X)}{(X_M - X_N)\,Z} & \text{if } X_M \neq X_N, \\ \dfrac{k\,(Y - Y_N Z) + (Y_M Z - Y)}{(Y_M - Y_N)\,Z} & \text{otherwise}, \end{cases}$$

fulfills the required conditions (notice that $X_M = X_N$ implies that $Y_M \neq Y_N$ since we assumed that $M \neq N$ and $Z_M = Z_N = 1$). Hence, the divisor

$$D = \begin{cases} (S) - (\mathcal{O}) + \operatorname{div}(f) & \text{if } S \notin \{M, N\}, \\ (S+T) - (T) + \operatorname{div}(f) & \text{otherwise}, \end{cases}$$

is mapped to (k, S), as wanted. *Et voilà:* we have therefore shown that ψ is the bijection we were looking for.

Proposition 5.1 *Let E be a smooth elliptic curve defined over $\mathbb{F}_q$, $T \in E\backslash\{\mathcal{O}, M, N, M-N, N-M\}$ and $\mathfrak{m} = (M) + (N)$ with M, $N \in E\backslash\{\mathcal{O}\}$, $M \neq N$ be given. Let also*

$$\begin{array}{rccc} \psi: & \operatorname{Pic}^0_{\mathfrak{m}}(E) & \longrightarrow & \mathbb{G}_m \times E \\ & [D]_{\mathfrak{m}} & \longmapsto & (k, S)\,, \end{array}$$

be such that the $\mathfrak{m}$-equivalence class of $D = \sum_{P \in E} n_P(P)$ corresponds to $S = \sum_{P\in E} n_P P \in E$ and $k = f(M)/f(N)$, where $f \in \overline{K}(E)^*$ is any function satisfying

$$\operatorname{div}(f) = \begin{cases} D - (S) + (\mathcal{O}) & \textit{if } S \notin \{M, N\}, \\ D - (S+T) + (T) & \textit{otherwise.} \end{cases}$$

Then, ψ is a well-defined bijection of sets.

Remark 5.2 *Notice that since the zero divisor can be written as*

$$\mathbf{0} = (\mathcal{O}) - (\mathcal{O}) + \operatorname{div}(c),$$

where c is any nonzero constant, then $\mathbf{0}$ corresponds to the pair $(1, \mathcal{O})$. That is, $(1, \mathcal{O})$ is the identity element of $J_{\mathfrak{m}}$.

5.3 The Group Law Algorithm

We here undertake the crucial task of inferring an algorithm to compute the group operation on $J_{\mathfrak{m}}$. So we are just about to establish the bridge between theory and practice: with the concrete equations at hand, even someone who never heard of generalized Jacobians or group cohomology before will just as well be able to understand the various properties of $J_{\mathfrak{m}}$.

Remember that by the theory of group extensions, we already know the basic structure of the addition on $J_{\mathfrak{m}}$. Actually, recall that by Theorem 4.7, we have for any k_1, $k_2 \in \mathbb{G}_{\mathrm{m}}$ and P_1, $P_2 \in E$,

$$(k_1, P_1) + (k_2, P_2) = (k_1 k_2 \cdot \mathbf{c}_{\mathfrak{m}}(P_1, P_2), P_1 + P_2)\,, \tag{5.4}$$

where $\mathbf{c}_{\mathfrak{m}} : E \times E \to \mathbb{G}_{\mathrm{m}}$ is a 2-cocycle depending on the modulus $\mathfrak{m}$. It thus suffices to make $\mathbf{c}_{\mathfrak{m}}$ explicit. As in the example of Section 4.5.1, we will roll up our sleeves and work directly with divisors. So given (k_1, P_1) and (k_2, P_2) in $J_{\mathfrak{m}}$, we wish to compute their sum (k_3, P_3).

There are two distinct cases to study, depending if the use of a *'translation point'* T is at all needed. Fortunately, there is an easy criterion to decide when it occurs. Indeed, suppose that the group we consider for cryptographic applications is the subgroup of $J_{\mathfrak{m}}$ generated by the element (k, P). By the addition rule (5.4), it immediately follows that

$$\textit{If } (j, Q) \in \langle (k, P) \rangle\,, \textit{ then } Q \in \langle P \rangle\,.$$

As a result, if neither M nor N is a multiple of P, then the group operation on $\langle (k, P) \rangle$ will *never* involve points of the form $(*, M)$ or $(*, N)$. Thus, there is no need to employ a translation point in this case. Of course, when either M or N lies in $\langle P \rangle$, then the corresponding addition formulæ will use translation points when appropriate in order to cover all possible cases. This motivates the following definition.

Definition 5.3 *Let E be an elliptic curve defined over $\mathbb{F}_q$ and $B \in E(\mathbb{F}_q)$ be a given basepoint. Let also M, $N \in E(\overline{\mathbb{F}_q})$ be given. Then the modulus $\mathfrak{m} = (M) + (N)$ is said to be B-unrelated if M, $N \notin \langle B \rangle$. Otherwise, it will be called B-related.*

5.3.1 Group Law for B-unrelated Moduli

As announced, the aim of this section is to *transport* the addition on $\mathrm{Pic}^0_{\mathfrak{m}}(E)$ to $\mathbb{G}_{\mathrm{m}} \times E$ in order to get explicit equations involving the group laws on $\mathbb{G}_{\mathrm{m}}$ and E. So given (k_1, P_1), (k_2, P_2) and (k_3, P_3) in $J_{\mathfrak{m}}$ such that

$$(k_1, P_1) + (k_2, P_2) = (k_3, P_3) \text{ and } P_1,\ P_2,\ P_3 \notin \{M, N\}\,,$$

our task is to express (k_3, P_3) in terms of (k_1, P_1) and (k_2, P_2). By the explicit bijection between $\mathrm{Pic}^0_{\mathfrak{m}}(E)$ and $\mathbb{G}_{\mathrm{m}} \times E$ (see Proposition 5.1), the elements (k_1, P_1) and (k_2, P_2) are respectively the image of the $\mathfrak{m}$-equivalence class of $D_1 = (P_1) - (\mathcal{O}) + \mathrm{div}(f_1)$ and $D_2 = (P_2) - (\mathcal{O}) + \mathrm{div}(f_2)$. Notice that since P_1, $P_2 \notin \{M, N\}$, then f_1 and f_2 are both defined and nonzero at M and N.

That being said, we can now endow $\mathbb{G}_{\mathrm{m}} \times E$ with the group operation inherited from $\mathrm{Pic}^{\,0}_{\,\mathfrak{m}}(E)$. So basically, all we need to know is to which element of $\mathbb{G}_{\mathrm{m}} \times E$ does $D_3 = D_1 + D_2$ correspond (and yes, this is the act where the hidden 2-cocycle finally makes its triumphal appearance). First, we have by definition that

$$D_3 = (P_1) + (P_2) - 2(\mathcal{O}) + \mathrm{div}(f_1 \cdot f_2), \tag{5.5}$$

so in order to get the element of $\mathbb{G}_{\mathrm{m}} \times E$ we are looking for, the way to go is to express the right hand side of (5.5) as $(P_3) - (\mathcal{O}) + \mathrm{div}(f_3)$. By Abel's theorem (c.f. Theorem 3.84), we know that

$$(P_1) + (P_2) - 2(\mathcal{O}) \sim (P_1 + P_2) - (\mathcal{O}),$$

and so there is a function $L_{P_1,P_2} \in \overline{K}(E)^*$ satisfying

$$(P_1) + (P_2) - 2(\mathcal{O}) = (P_1 + P_2) - (\mathcal{O}) + \mathrm{div}(L_{P_1,P_2}). \tag{5.6}$$

Combining (5.6) and (5.5) yields

$$D_3 = (P_1 + P_2) - (\mathcal{O}) + \mathrm{div}(f_1 \cdot f_2 \cdot L_{P_1,P_2}).$$

Phantastisch! That means that we can set $P_3 = P_1 + P_2$ and $f_3 = f_1 \cdot f_2 \cdot L_{P_1,P_2}$. Hence, D_3 corresponds to (k_3, P_3), where

$$k_3 = \frac{f_3(M)}{f_3(N)} = \frac{f_1(M) \cdot f_2(M) \cdot L_{P_1,P_2}(M)}{f_1(N) \cdot f_2(N) \cdot L_{P_1,P_2}(N)} = k_1 \cdot k_2 \cdot \frac{L_{P_1,P_2}(M)}{L_{P_1,P_2}(N)}.$$

That is,

$$(k_1, P_1) + (k_2, P_2) = \left(k_1 \cdot k_2 \cdot \frac{L_{P_1,P_2}(M)}{L_{P_1,P_2}(N)}, P_1 + P_2\right).$$

So we really are on the right track since our addition rule so far agrees with the prediction (5.4) obtained from group extensions. Hence the 2-cocycle $\mathbf{c}_{\mathfrak{m}} : E \times E \to \mathbb{G}_{\mathrm{m}}$ we were seeking is finally unveiled:

$$\mathbf{c}_{\mathfrak{m}}(P_1, P_2) = \frac{L_{P_1,P_2}(M)}{L_{P_1,P_2}(N)}. \tag{5.7}$$

The very last step is to make L_{P_1,P_2} explicit. We have to look for a function L_{P_1,P_2} satisfying (5.6), or equivalently,

$$\mathrm{div}(L_{P_1,P_2}) = (P_1) + (P_2) - (P_1 + P_2) - (\mathcal{O}). \tag{5.8}$$

That should sound familiar now (See Lemma 3.82). The natural approach is to consider the line ℓ_{P_1,P_2}, passing through P_1 and P_2, that will inevitably hit $-P_3 = -(P_1 + P_2)$ as well. Then,

$$\operatorname{div}\left(\frac{\ell_{P_1,P_2}}{Z}\right) = (P_1) + (P_2) + (-P_3) - 3(\mathcal{O}). \tag{5.9}$$

Not exactly what we want yet, so in order to introduce the term $-(P_1+P_2)$ and get rid of $(-P_3)$ at once, we might want to look at $\ell_{P_1+P_2,\mathcal{O}}$, which is of course the line passing through $P_1 + P_2$, $\mathcal{O}$, and a fortiori through $-P_3$. That is,

$$\operatorname{div}\left(\frac{\ell_{P_1+P_2,\mathcal{O}}}{Z}\right) = (P_1 + P_2) + (-P_3) - 2(\mathcal{O}). \tag{5.10}$$

Subtracting (5.10) from (5.9), we get

$$\operatorname{div}\left(\frac{\ell_{P_1,P_2}}{\ell_{P_1+P_2,\mathcal{O}}}\right) = (P_1) + (P_2) - (P_1 + P_2) - (\mathcal{O}). \tag{5.11}$$

Finally, equations (5.8) and (5.11) imply that L_{P_1,P_2} and $\ell_{P_1,P_2}/\ell_{P_1+P_2,\mathcal{O}}$ differ by a nonzero multiplicative constant:

$$\exists c \in \overline{K}^* \text{ satisfying } L_{P_1,P_2} = c \cdot \frac{\ell_{P_1,P_2}}{\ell_{P_1+P_2,\mathcal{O}}}. \tag{5.12}$$

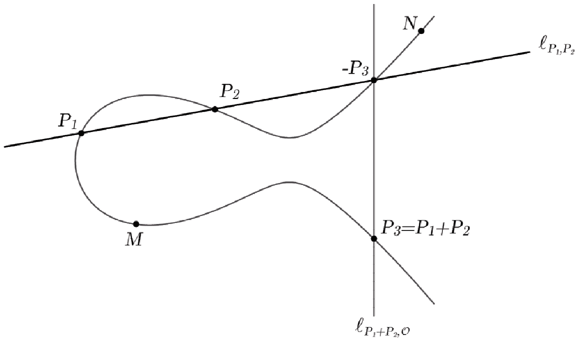


Figure 5.2: Unveiling the 2-cocycle $\mathbf{c}_\mathfrak{m}$

Let's point out that our initial conditions M, $N \neq \mathcal{O}$ and P_1, P_2, $P_3 = P_1 + P_2 \notin \{M, N\}$ are sufficient to ensure that $L_{P_1,P_2}(M)$ and $L_{P_1,P_2}(N)$ will both be defined and nonzero, since equation (5.8) tells us that the only zeros and poles of L_{P_1,P_2} occur at P_1, P_2, P_1+P_2 and $\mathcal{O}$. But say we want to compute $L_{P_1,P_2}(M)$ by evaluating $\ell_{P_1,P_2}(M)$ and $\ell_{P_1+P_2,\mathcal{O}}(M)$ separately. That will work just fine as long as $M \neq -P_3$. But when $M = -P_3$, we get $\ell_{P_1,P_2}(M) = \ell_{P_1+P_2,\mathcal{O}}(M) =$

0 and so evaluating $\ell_{P_1,P_2}(M)/\ell_{P_1+P_2,\mathcal{O}}(M)$ amounts to study the indeterminate form '0/0'. Since the goal of this section is to obtain a group law valid for any P_1, P_2, $P_3 \in \langle B\rangle$ under the assumption that M, $N \notin \langle B\rangle$, then it follows that $-P_3 \notin \{M,N\}$ anyway, so we do not have to worry about this case now. We can then add the extra requirement $-(P_1+P_2) \notin \{M,N\}$ to the points we consider and simply move on.

Therefore, by equations (5.7) and (5.12), it is now legitimate to write

$$\mathbf{c}_{\mathfrak{m}}(P_1,P_2) = \frac{L_{P_1,P_2}(M)}{L_{P_1,P_2}(N)} = \frac{c\cdot\ell_{P_1,P_2}(M)}{\ell_{P_1+P_2,\mathcal{O}}(M)}\cdot\frac{\ell_{P_1+P_2,\mathcal{O}}(N)}{c\cdot\ell_{P_1,P_2}(N)} = \frac{\ell_{P_1,P_2}(M)}{\ell_{P_1+P_2,\mathcal{O}}(M)}\cdot\frac{\ell_{P_1+P_2,\mathcal{O}}(N)}{\ell_{P_1,P_2}(N)}, \quad (5.13)$$

and our goal is achieved since the 2-cocycle $\mathbf{c}_{\mathfrak{m}}$ is now completely determined. To be on the safe side, we may want to double-check that expression (5.13) is well-defined since after all, we have some freedom on both the equations of the lines (they are determined up to a constant factor) and on the representatives for the homogeneous coordinates of M and N. That is, for $M = (X_M : Y_M : 1)$, $N = (X_N : Y_N : 1)$ and λ_1, λ_2, c_1, c_2 any nonzero constants, we have $M \sim (\lambda_1 X_M : \lambda_1 Y_M : \lambda_1)$, $N \sim (\lambda_2 X_N : \lambda_2 Y_N : \lambda_2)$ and $c_1\cdot\ell_{P_1,P_2}$, $c_2\cdot\ell_{P_1+P_2,\mathcal{O}}$ respectively defining the same line as ℓ_{P_1,P_2} and $\ell_{P_1+P_2,\mathcal{O}}$. Since ℓ_{P_1,P_2} and $\ell_{P_1+P_2,\mathcal{O}}$ are both homogeneous polynomials of degree one, it follows that

$$\frac{c_1\cdot\ell_{P_1,P_2}(\lambda_1X_M,\lambda_1Y_M,\lambda_1)}{c_2\cdot\ell_{P_1+P_2,\mathcal{O}}(\lambda_1X_M,\lambda_1Y_M,\lambda_1)}\cdot\frac{c_2\cdot\ell_{P_1+P_2,\mathcal{O}}(\lambda_2X_N,\lambda_2Y_N,\lambda_2)}{c_1\cdot\ell_{P_1,P_2}(\lambda_2X_N,\lambda_2Y_N,\lambda_2)} =$$
$$\frac{\lambda_1\cdot\ell_{P_1,P_2}(X_M,Y_M,1)}{\lambda_1\cdot\ell_{P_1+P_2,\mathcal{O}}(X_M,Y_M,1)}\cdot\frac{\lambda_2\cdot\ell_{P_1+P_2,\mathcal{O}}(X_N,Y_N,1)}{\lambda_2\cdot\ell_{P_1,P_2}(X_N,Y_N,1)} =$$
$$\frac{\ell_{P_1,P_2}(M)}{\ell_{P_1+P_2,\mathcal{O}}(M)}\cdot\frac{\ell_{P_1+P_2,\mathcal{O}}(N)}{\ell_{P_1,P_2}(N)},$$

which confirms that formula (5.13) was well-defined. Finally, we are ready to properly write down the group law we just obtained.

Proposition 5.4 *Let E be a smooth elliptic curve and let $\mathfrak{m} = (M) + (N)$ be given such that M and N are distinct nonzero points of E. If (k_1,P_1) and (k_2,P_2) are elements of $J_{\mathfrak{m}}$ fulfilling P_1, P_2, $\pm(P_1+P_2) \notin \{M,N\}$, then*

$$(k_1,P_1) + (k_2,P_2) = (k_1k_2\cdot\mathbf{c}_{\mathfrak{m}}(P_1,P_2), P_1+P_2), \quad (5.14)$$

where $\mathbf{c}_{\mathfrak{m}} : E\times E \to \mathbb{G}_m$ is the 2-cocycle given by

$$\mathbf{c}_{\mathfrak{m}}(P_1,P_2) = \frac{\ell_{P_1,P_2}(M)}{\ell_{P_1+P_2,\mathcal{O}}(M)}\cdot\frac{\ell_{P_1+P_2,\mathcal{O}}(N)}{\ell_{P_1,P_2}(N)},$$

and $\ell_{P,Q}$ denotes the equation of the straight line passing through P and Q (tangent at the curve if $P = Q$).

5.3.2 Group Law for B-related Moduli

Inspired by the method used to obtain a group operation for B-unrelated moduli, we here treat the general case of adding *arbitrary* points of $J_{\mathfrak{m}}$. So let (k_1, P_1), (k_2, P_2) and (k_3, P_3) be elements of $J_{\mathfrak{m}}$ satisfying

$$(k_3, P_3) = (k_1, P_1) + (k_2, P_2).$$

As suggested in Section 5.2, an easy way to proceed is to use what we called a *'translation point'* each time we encounter a divisor whose support contains M or N (see p.126 for details). So we first need to fix a point $T \notin \{\mathcal{O}, M, N, M-N, N-M\}$ of E and let, for $i = 1, 2$,

$$R_i = \begin{cases} T & \text{if } P_i \in \{M, N\}, \\ \mathcal{O} & \text{otherwise.} \end{cases}$$

That way, $(P_i) - (\mathcal{O}) \sim (P_i + R_i) - (R_i)$ and the support of the divisor on the right hand side satisfies $\{P_i + R_i, R_i\} \cap \{M, N\} = \emptyset$. For $i = 1, 2$, let

$$D_i = (P_i + R_i) - (R_i) + \operatorname{div}(f_i), \tag{5.15}$$

and notice that this implies that $\operatorname{ord}_M(f_i) = \operatorname{ord}_N(f_i) = 0$. Using the bijection $\psi : \operatorname{Pic}^0_{\mathfrak{m}}(E) \to \mathbb{G}_{\mathrm{m}} \times E$ of Proposition 5.1, the $\mathfrak{m}$-equivalence class of D_i $(i = 1, 2)$ can be specified by the pair (k_i, P_i), where

$$k_i = \frac{f_i(M)}{f_i(N)}.$$

With all this information at hand, we should now be able to find an expression for k_3 and P_3 in terms of k_1, k_2, P_1 and P_2 in the twinkling of an eye. First set $D_3 = D_1 + D_2$ and use equation (5.15) in order to rewrite D_3 as

$$D_3 = (P_1 + R_1) + (P_2 + R_2) - (R_1) - (R_2) + \operatorname{div}(f_1 \cdot f_2). \tag{5.16}$$

Just as before, we need to find a way to express this divisor as

$$D_3 = (P_3 + R_3) - (R_3) + \operatorname{div}(f_3),$$

where R_3 will of course be defined according to the value of P_3:

$$R_3 = \begin{cases} T & \text{if } P_3 \in \{M, N\}, \\ \mathcal{O} & \text{otherwise,} \end{cases}$$

and $f_3 \in \overline{K}(E)^*$ is a function yet to be determined. Once more, Abel's theorem (c.f. Theorem 3.84) will provide the intuition we need since

$$(P_1 + R_1) + (P_2 + R_2) - (R_1) - (R_2) \sim (P_1 + P_2 + R_3) - (R_3).$$

Hence, there is an $L \in \overline{K}(E)^*$ (involving possibly all of P_1, P_2, P_3, R_1, R_2 and R_3) such that

$$(P_1 + R_1) + (P_2 + R_2) - (R_1) - (R_2) = (P_1 + P_2 + R_3) - (R_3) + \operatorname{div}(L). \tag{5.17}$$

From (5.16) and (5.17), we obtain:

$$D_3 = (P_1 + P_2 + R_3) - (R_3) + \operatorname{div}(f_1 \cdot f_2 \cdot L),$$

and we can simply let $P_3 = P_1 + P_2$ and $f_3 = f_1 \cdot f_2 \cdot L$. So it means that D_3 is associated with (k_3, P_3), where

$$k_3 = \frac{f_3(M)}{f_3(N)} = \frac{f_1(M) \cdot f_2(M) \cdot L(M)}{f_1(N) \cdot f_2(N) \cdot L(N)} = k_1 \cdot k_2 \cdot \frac{L(M)}{L(N)}.$$

Playing with lines will once more prove to be a good tactic to deduce an explicit expression for L. So in order to see which ones we should consider, we first take the time to rewrite (5.17) as

$$\operatorname{div}(L) = (P_1 + R_1) - (R_1) + (P_2 + R_2) - (R_2) - (P_3 + R_3) + (R_3). \tag{5.18}$$

A quick inspection of this principal divisor suggests that the favorite candidates are the following six straight lines: ℓ_{P_1,R_1}, $\ell_{P_1+R_1,\mathcal{O}}$, ℓ_{P_2,R_2}, $\ell_{P_2+R_2,\mathcal{O}}$, ℓ_{P_3,R_3} and $\ell_{P_3+R_3,\mathcal{O}}$ (yes, this is what it takes). Plus, in order to exactly obtain (5.18), it might not be a bad idea to consider ℓ_{P_1,P_2} and $\ell_{P_1+P_2,\mathcal{O}}$ as well. We therefore get:

$$\begin{aligned}
\operatorname{div}\left(\frac{\ell_{P_3,R_3}}{\ell_{P_3+R_3,\mathcal{O}}}\right) &= (P_3) + (R_3) - (P_3 + R_3) - (\mathcal{O}),\\
-\operatorname{div}\left(\frac{\ell_{P_1,R_1}}{\ell_{P_1+R_1,\mathcal{O}}}\right) &= (P_1 + R_1) - (P_1) - (R_1) + (\mathcal{O}),\\
-\operatorname{div}\left(\frac{\ell_{P_2,R_2}}{\ell_{P_2+R_2,\mathcal{O}}}\right) &= (P_2 + R_2) - (P_2) - (R_2) + (\mathcal{O}),\\
\operatorname{div}\left(\frac{\ell_{P_1,P_2}}{\ell_{P_1+P_2,\mathcal{O}}}\right) &= (P_1) + (P_2) - (P_1 + P_2) - (\mathcal{O}).
\end{aligned}$$

Adding these four equations yields

$$\begin{gathered}
\operatorname{div}\left(\frac{\ell_{P_1,P_2}}{\ell_{P_1+P_2,\mathcal{O}}} \cdot \frac{\ell_{P_1+R_1,\mathcal{O}}}{\ell_{P_1,R_1}} \cdot \frac{\ell_{P_2+R_2,\mathcal{O}}}{\ell_{P_2,R_2}} \cdot \frac{\ell_{P_3,R_3}}{\ell_{P_3+R_3,\mathcal{O}}}\right) = \\
(P_1 + R_1) - (R_1) + (P_2 + R_2) - (R_2) - (P_3 + R_3) + (R_3).
\end{gathered} \tag{5.19}$$

From (5.18) and (5.19), we have consequently determined L up to a nonzero constant. But since this constant will cancel out when computing $L(M)/L(N)$, we can without loss of generality assume that

$$L = \frac{\ell_{P_1,P_2}}{\ell_{P_3,\mathcal{O}}} \cdot \frac{\ell_{P_1+R_1,\mathcal{O}}}{\ell_{P_1,R_1}} \cdot \frac{\ell_{P_2+R_2,\mathcal{O}}}{\ell_{P_2,R_2}} \cdot \frac{\ell_{P_3,R_3}}{\ell_{P_3+R_3,\mathcal{O}}}. \tag{5.20}$$

As we can see, the term $\ell_{P_1,P_2}/\ell_{P_1+P_2,\mathcal{O}}$ still appears, but is now followed by *'correction factors'* that will ensure, thanks to (5.18), that the *only* true zeros and poles of L arise at R_i and P_i+R_i ($i = 1, 2, 3$), which are all different from M and N by construction. Hence, the quotient $L(M)/L(N)$ will always be defined and nonzero.

Since the group operation we just obtained holds on all of $J_{\mathfrak{m}}$, we might wonder what happens when P_1, P_2, $\pm P_3 \notin \{M, N\}$. In this case, we have $R_1 = R_2 = R_3 = \mathcal{O}$ and (5.20) reduces to:

$$L = \frac{\ell_{P_1,P_2}}{\ell_{P_3,\mathcal{O}}} \cdot \frac{\ell_{P_1,\mathcal{O}}}{\ell_{P_1,\mathcal{O}}} \cdot \frac{\ell_{P_2,\mathcal{O}}}{\ell_{P_2,\mathcal{O}}} \cdot \frac{\ell_{P_3,\mathcal{O}}}{\ell_{P_3,\mathcal{O}}} = \frac{\ell_{P_1,P_2}}{\ell_{P_3,\mathcal{O}}},$$

which coincides with Proposition 5.4 for B-unrelated moduli. So evaluating a group operation on $J_{\mathfrak{m}}$ will be more expensive as soon as one of the P_i's equals M or N. The relevance of this difference will be addressed in Section 5.3.4. But first, it is time to summarize what we've got.

Proposition 5.5 *Let E be a smooth elliptic curve, $\mathfrak{m} = (M) + (N)$ be given such that M and N are distinct nonzero points of E and let $T \in E$ be any point such that $T \notin \{\mathcal{O}, M, N, M-N, N-M\}$. Given (k_1, P_1) and (k_2, P_2) in $J_{\mathfrak{m}}$, set $P_3 = P_1 + P_2$ and let, for $i = 1, 2, 3$,*

$$R_i = \begin{cases} T & \text{if } P_i \in \{M, N\}, \\ \mathcal{O} & \text{otherwise.} \end{cases}$$

Then,

$$(k_1, P_1) + (k_2, P_2) = \left(k_1 k_2 \cdot \frac{L(M)}{L(N)}, P_3\right),$$

where

$$L = \frac{\ell_{P_1,P_2}}{\ell_{P_3,\mathcal{O}}} \cdot \frac{\ell_{P_1+R_1,\mathcal{O}}}{\ell_{P_1,R_1}} \cdot \frac{\ell_{P_2+R_2,\mathcal{O}}}{\ell_{P_2,R_2}} \cdot \frac{\ell_{P_3,R_3}}{\ell_{P_3+R_3,\mathcal{O}}}.$$

As usual, $\ell_{P,Q}$ denotes the equation of the straight line passing through P and Q (tangent at the curve if $P = Q$).

5.3.3 Toy Example

Before going any further, we work out a tiny *paper and pencil* example in order to get a flavor of how the computations in $J_{\mathfrak{m}}$ will be performed in practice. It is also the right time to start looking for tricks to speed things up, since everyone knows that computing by hand gives us a strong motivation (and plenty of time!) to realize what shortcuts could be considered.

We will work with the generalized Jacobian of the elliptic curve

$$E : y^2 = x^3 + x + 4$$

over $\mathbb{F}_p = \mathbb{F}_7$ with respect to the modulus $\mathfrak{m} = (M) + (N)$, where $M = (x_M, y_M) = (0, 5)$ and $N = (x_N, y_N) = (5, 1)$. All computations are performed in the subgroup of $J_{\mathfrak{m}}$ generated by (k, P), where $k = 1$ and $P = (x_P, y_P) = (6, 3)$. Notice that p, E, M, N, P and k are all publicly known quantities that determine the group $\langle (k, P) \rangle$ we are working in, and provide the necessary information to perform the group operation inside $\langle (k, P) \rangle$.

Before we begin, we can quickly verify that E is nonsingular, as $a = 1$, $b = 4$ and $\Delta = -16 \cdot \left(4a^3 + 27b^2\right) = 3 \neq 0$. Next we want to compute multiples of $(1, P)$ in order to get an idea of what $\langle (1, P) \rangle$ looks like. In the present case, it is easier to work in affine coordinates since we do not have to worry about the cost of inversions (of course, homogeneous coordinates would have worked perfectly fine too).

We now proceed to compute $2(1, P) = (1, P) + (1, P)$. According to Theorem 3.55, we have $\ell_{P,P}(x, y) = y - \mathbf{m}x - \mathbf{b}$, where

$$\mathbf{m} = \frac{3x_P^2 + a}{2y_P} = \frac{3 \cdot 6^2 + 1}{2 \cdot 3} = 3 \text{ and } \mathbf{b} = \frac{-x_P^3 + ax_P + 2b}{2y_P} = \frac{-6^3 + 6 + 2 \cdot 4}{2 \cdot 3} = 6.$$

Thus, $2P = (x_{2P}, y_{2P}) = (4, 3)$ since

$$x_{2P} = \mathbf{m}^2 - 2x_P = 3^2 - 2 \cdot 6 = 4 \quad \text{and} \quad y_{2P} = -\mathbf{m}x_{2P} - \mathbf{b} = -3 \cdot 4 - 6 = 3.$$

It follows that $\ell_{2P,\mathcal{O}}(x, y) = x - x_{2P} = x - 4$, and we have

$$\frac{\ell_{P,P}(M)}{\ell_{2P,\mathcal{O}}(M)} = \frac{\ell_{P,P}(x_M, y_M)}{\ell_{2P,\mathcal{O}}(x_M, y_M)} = \frac{y_M - \mathbf{m}x_M - \mathbf{b}}{x_M - 4} = \frac{5 - 3 \cdot 0 - 6}{-4} = 2.$$

Similarly,

$$\frac{\ell_{2P,\mathcal{O}}(N)}{\ell_{P,P}(N)} = \frac{\ell_{2P,\mathcal{O}}(x_N, y_N)}{\ell_{P,P}(x_N, y_N)} = \frac{x_N - 4}{y_N - \mathbf{m}x_N - \mathbf{b}} = \frac{5 - 4}{1 - 3 \cdot 5 - 6} = 1.$$

Finally, by Proposition 5.4 , we get that

$$2(1, P) = \left(1 \cdot 1 \cdot \frac{\ell_{P,P}(M)}{\ell_{2P,\mathcal{O}}(M)} \cdot \frac{\ell_{2P,\mathcal{O}}(N)}{\ell_{P,P}(N)}, 2P\right) = (2, 2P) = (2, (4, 3)).$$

Almost too easy. We could then continue to compute the multiples of $(1, P)$ by hand, and perhaps make it an interesting alternative to counting sheep at night... Otherwise, a small Magma program readily produces the output shown in Table 5.1.

Since neither $M = (0, 5)$ nor $N = (5, 1)$ appear in this table, that means that $\mathfrak{m} = (M) + (N)$ is a P-unrelated modulus (and this is why we did not need to specify a translation point here). We also have that $(1, P)$ has order $30 = \#\mathbb{F}_7^* \times \operatorname{ord}(P)$ and

$$\langle (1, P) \rangle = \{ (i, Q) | \, i \in \mathbb{F}_7^* \text{ and } Q \in \langle P \rangle \}.$$

We will come back to this example a little latter, although it is overwhelmingly tempting to look at the DLP in $\langle (1, P) \rangle$ right away... (Unthinkable to wait until Section 5.5? Then cogitate on a good general strategy[3] to recover, say, 27 from the couple $(3, (4, 3))$).

[3] We of course have to *pretend* that we can't do an exhaustive search here.

$(1,P)=(1,(6,3))$	$11(1,P)=(2,(6,3))$	$21(1,P)=(4,(6,3))$
$2(1,P)=(2,(4,3))$	$12(1,P)=(4,(4,3))$	$22(1,P)=(1,(4,3))$
$3(1,P)=(4,(4,4))$	$13(1,P)=(1,(4,4))$	$23(1,P)=(2,(4,4))$
$4(1,P)=(4,(6,4))$	$14(1,P)=(1,(6,4))$	$24(1,P)=(2,(6,4))$
$5(1,P)=(3,\mathcal{O})$	$15(1,P)=(6,\mathcal{O})$	$25(1,P)=(5,\mathcal{O})$
$6(1,P)=(3,(6,3))$	$16(1,P)=(6,(6,3))$	$26(1,P)=(5,(6,3))$
$7(1,P)=(6,(4,3))$	$17(1,P)=(5,(4,3))$	$27(1,P)=(3,(4,3))$
$8(1,P)=(5,(4,4))$	$18(1,P)=(3,(4,4))$	$28(1,P)=(6,(4,4))$
$9(1,P)=(5,(6,4))$	$19(1,P)=(3,(6,4))$	$29(1,P)=(6,(6,4))$
$10(1,P)=(2,\mathcal{O})$	$20(1,P)=(4,\mathcal{O})$	$30(1,P)=(1,\mathcal{O})$

Table 5.1: Multiples of $(1,P)$ obtained with Magma

5.3.4 Properties of the Group Law

Now that we have derived nice explicit formulæ for the group operation, we take a (well-deserved!) pause for contemplating them. That's right. This small section aims at collecting tricks, remarks and corollaries on the group law in order to be fully prepared for the more serious efficiency and security aspects of the next sections.

B-RELATED OR UNRELATED? That is the question, indeed. Well, the obvious remark is that B-related moduli offer greater generality while the group law associated to B-unrelated ones is simpler. But if we were to pick one for applications, which one should we choose? First recall that by Proposition 5.4 and 5.5, the general group law (both for B-related and unrelated moduli) can be rewritten as

$$(k_1,P_1)+(k_2,P_2)=\left(k_1k_2\cdot\frac{L(M)}{L(N)},P_1+P_2\right),$$

where

$$L=\begin{cases}\dfrac{\ell_{P_1,P_2}}{\ell_{P_3,\mathcal{O}}} & \text{if } \{P_1,P_2,P_1+P_2,-P_1-P_2\}\cap\{M,N\}=\emptyset,\\ \dfrac{\ell_{P_1,P_2}}{\ell_{P_3,\mathcal{O}}}\cdot\dfrac{\ell_{P_1+R_1,\mathcal{O}}\cdot\ell_{P_2+R_2,\mathcal{O}}\cdot\ell_{P_3,R_3}}{\ell_{P_1,R_1}\cdot\ell_{P_2,R_2}\cdot\ell_{P_3+R_3,\mathcal{O}}} & \text{otherwise.}\end{cases}$$

The straightforward implementation of this group law will then satisfy the following two properties:

1. Computing $(k_1,P_1)+(k_2,P_2)$ when $\{P_1,P_2,P_1+P_2,-P_1-P_2\}\cap\{M,N\}\neq\emptyset$ will require more computing time than if $\{P_1,P_2,P_1+P_2,-P_1-P_2\}\cap\{M,N\}=\emptyset$.

2. The code will contain conditional *'if-then-else'* statements in order to compute L.

The obvious consequence of property 1 is that some computing time can be saved if we employ a B-unrelated modulus. This discrepancy in efficiency is however the lesser of two evils. As a matter of fact, B-related moduli seem to be more susceptible to *side-channel attacks*. Recall that the general (and greatly simplified!) principle behind these attacks is to measure side-channel information of a cryptographic device (like running times, power consumption or electromagnetic emanations) in order to retrieve secret data[4]. As a result, codes whose running time depend on secret input data have to be avoided as much as possible in practice. In addition, the collected side-channel information is sometimes even sufficient to retrieve which branch of a conditional statement was executed, therefore revealing information about the value of the condition tested. Therefore, the straightforward algorithm for computing a group operation for a B-related modulus violates two basic principles for minimizing side-channel leakage.

These remarks show that the efficiency and security characteristics of B-related and B-unrelated moduli differ, so they should be studied separately. The above evidences also suggest that B-unrelated moduli might have a higher potential for practical implementations, and since the ultimate objective of this chapter is to build a practical cryptosystem based on a simple generalized Jacobian in order to highlight the potential of these algebraic groups, B-unrelated moduli were chosen for our case study (inevitably relegating B-related moduli to further work). Thus from this point on, we will assume that the moduli we consider are all B-unrelated.

BASIC PROPERTIES. We here present a small collection of properties of the group law that will prove to be very useful in the sequel. In fact, they are either remarks previously made in the text (that we formally state for the record) or else are easily derived from Proposition 5.4 .

Corollary 5.6 *Let E be a smooth elliptic curve and let $\mathfrak{m} = (M) + (N)$ be given such that M and N are distinct nonzero points of E. Let also (k, P), (k_1, P_1), $(k_2, P_2) \in J_\mathfrak{m}$ be given such that P_1, P_2, $\pm(P_1 + P_2) \notin \{M, N\}$. Then,*

1. *$(1, \mathcal{O})$ is the identity element of $J_\mathfrak{m}$.*
2. *$\mathbf{c}_\mathfrak{m}(P_1, P_2) = \mathbf{c}_\mathfrak{m}(P_2, P_1)$ (This reflects the fact that $J_\mathfrak{m}$ is abelian).*
3. *If $M = (X_M : Y_M : 1)$ and $N = (X_N : Y_N : 1)$, then $\mathbf{c}_\mathfrak{m}(P, -P) = \ell_{P,\mathcal{O}}(M) / \ell_{P,\mathcal{O}}(N)$, and so the inverse of (k, P) is given by*

$$-(k, P) = \left(\frac{1}{k} \cdot \frac{\ell_{P,\mathcal{O}}(N)}{\ell_{P,\mathcal{O}}(M)}, -P \right)$$

[4] More information concerning these attacks can be found in the original articles [Koc96, KJJ99], as well as in Chapters IV and V of [BSS05].

4. *$\mathbf{c}_{\mathfrak{m}}(\mathcal{O}, P) = 1$ for all $P \in E \backslash \{M, N\}$. Hence,*

$$(k_1, \mathcal{O}) + (k_2, P) = (k_1 k_2, P)\,.$$

5. *Furthermore, $J_{\mathfrak{m}}$ contains a subgroup isomorphic to $\mathbb{G}_m$, as*

$$(k_1, \mathcal{O}) + (k_2, \mathcal{O}) = (k_1 k_2, \mathcal{O}) \text{ for all } k_1, k_2 \in \mathbb{G}_m\,.$$

6. *If E is defined over $\mathbb{F}_q$, $B \in E(\mathbb{F}_q)$ and M, $N \in E(\mathbb{F}_{q^r})$ are such that $\mathfrak{m}$ is B-unrelated, then $\mathbb{F}_{q^r}^* \times \langle B \rangle$, together with the addition law of Proposition 5.4, is a subgroup of $J_{\mathfrak{m}}$.*

The only statement that might require a further justification is property 6. Notice that it simply follows from properties 1 and 3, together with the observation that $\ell_{P_1,P_2}(M)$, $\ell_{P_1,P_2}(N) \in \mathbb{F}_{q^r}^*$ whenever P_1, $P_2 \in \langle B \rangle$. So at last, we have made completely explicit the finite group $\mathbb{F}_{q^r}^* \times \langle B \rangle$ that we will be using for cryptographic applications.

5.4 Efficiency

Now that we have the group law algorithm at hand, it is time to wonder about the practicality of generalized Jacobians: from the choice of a B-unrelated modulus to scalar multiplication, various efficiency aspects have to be addressed.

5.4.1 Additions in the Group

The first remark in order is that the group operation on $J_{\mathfrak{m}}$ has mainly two parts: performing an addition on E and evaluating the cocycle. These two steps involve the equation of the *same* straight lines, so intermediate computations for the addition on E should be reused in order to get the cocycle value. As a second remark, notice that the coordinates of $M = (X_M : Y_M : 1)$ and $N = (X_N : Y_N : 1)$ will be much used in the evaluation of the cocycle. Since we have the freedom to select the modulus of our choice, this might be an opportunity to speed up the computations. For instance, some (or even all) of X_M, X_N, Y_M and Y_N could be chosen such that the cost of a multiplication of a field element by these special coordinates becomes significantly faster than that of a general multiplication. For this reason, we will thereafter make a distinction between a general multiplication, which will be denoted by the symbol '$*$', and a multiplication by a constant, represented by '$\cdot$'.

For (k_1, P_1), $(k_2, P_2) \in \mathbb{F}_{q^r}^* \times \langle B \rangle$, we now wish to determine the cost of computing $(k_3, P_3) = (k_1, P_1) + (k_2, P_2)$. We will first work in affine coordinates, so let $M = (x_M, y_M)$ and $N =$

(x_N, y_N) be given. Recall that by property 4 of Corollary 5.6, $(k_3, P_3) = (k_1 * k_2, P_1 + P_2)$ as soon as P_1 or P_2 equals $\mathcal{O}$. Thus in this case the computation cost is merely a multiplication in $\mathbb{F}_{q^r}^*$ plus an addition on E, which will be abbreviated by $\mathbf{M} + \mathbf{E}$. Now if P_1, $P_2 \neq \mathcal{O}$ and $P_1 + P_2 = \mathcal{O}$, then

$$\begin{aligned}(k_3, P_3) &= \left(\frac{k_1 * k_2 * \ell_{P_1,P_2}(M) * \ell_{P_1+P_2,\mathcal{O}}(N)}{\ell_{P_1,P_2}(N) * \ell_{P_1+P_2,\mathcal{O}}(M)}, P_1 + P_2\right) \\ &= \left(\frac{k_1 * k_2 * \ell_{P_1,\mathcal{O}}(M) * 1}{\ell_{P_1,\mathcal{O}}(N) * 1}, P_1 + P_2\right) \\ &= \left(k_1 * k_2 * (x_M - x_1) * \frac{1}{(x_N - x_1)}, P_1 + P_2\right),\end{aligned}$$

where $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Thus the associated cost is three multiplications and an inversion in $\mathbb{F}_{q^r}^*$, together with an addition on E, or $3\mathbf{M} + \mathbf{I} + \mathbf{E}$ for short. Finally, we consider the case where P_1, P_2, $P_3 \neq \mathcal{O}$, and we let $P_3 = (x_3, y_3)$. According to Theorem 3.55 , the slope $\mathbf{m}$ of the line passing through P_1 and P_2 will be computed as an intermediate result while evaluating P_3 since

$$P_3 = \left(\mathbf{m}^2 + a_1\mathbf{m} - a_2 - x_1 - x_2, (x_1 - x_3)\,\mathbf{m} - a_1x_3 - y_1 - a_3\right).$$

Thus,

$$\begin{aligned}(k_3, P_3) &= \left(\frac{k_1 * k_2 * \ell_{P_1,P_2}(M) * \ell_{P_1+P_2,\mathcal{O}}(N)}{\ell_{P_1,P_2}(N) * \ell_{P_1+P_2,\mathcal{O}}(M)}, P_1 + P_2\right) \\ &= \left(\frac{k_1 * k_2 * (y_M - y_1 + (x_1 - x_M) * \mathbf{m}) * (x_N - x_3)}{(y_N - y_1 + (x_1 - x_N) * \mathbf{m}) * (x_M - x_3)}, P_1 + P_2\right) \\ &= \left(\frac{k_1 * k_2 * (y_M - y_1 + x_1 * \mathbf{m} - x_M \cdot \mathbf{m}) * (x_N - x_3)}{(y_N - y_1 + x_1 * \mathbf{m} - x_N \cdot \mathbf{m}) * (x_M - x_3)}, P_1 + P_2\right)\end{aligned}$$

which yields a cost of 5 general multiplications, two multiplications by a constant and one inversion in $\mathbb{F}_{q^r}^*$, plus an addition on E, or $5\mathbf{M} + 2\mathbf{C} + \mathbf{I} + \mathbf{E}$.

However, as soon as the cost of a field inversion is significantly higher than that of a multiplication, projective coordinates will be preferred. Recall that in (ordinary) projective coordinates, $(X : Y : Z)$ corresponds to the affine point $(X/Z, Y/Z)$ if $Z \neq 0$ and to $\mathcal{O}$ otherwise. So let $P_i = (X_i : Y_i : Z_i)$ $(i = 1, 2, 3)$ and $M = (X_M : Y_M : 1)$, $N = (X_N : Y_N : 1)$ be the projective coordinates of the points.

In the most common case where P_1, P_2, $P_3 \neq \mathcal{O}$, the corresponding formulæ for computing $P_1 + P_2$ will not evaluate $\mathbf{m}$ directly (since that would require an inversion), but rather computes quantities α and β such that $\mathbf{m} = \alpha/\beta$.

Remark 5.7 *While computing a scalar multiple of an element, individually storing both the numerators and denominators is profitable since it allows to perform a single inversion at the very end of the computation.*

So with the values of α and β already known, we can evaluate

$$\begin{aligned}\gamma &= ((Y_M \cdot Z_1 - Y_1) * \beta + (X_1 - X_M \cdot Z_1) * \alpha) * (X_N \cdot Z_3 - X_3) \text{ and} \\ \delta &= ((Y_N \cdot Z_1 - Y_1) * \beta + (X_1 - X_N \cdot Z_1) * \alpha) * (X_M \cdot Z_3 - X_3)\end{aligned}$$

such that $\mathbf{c}_{\mathfrak{m}}(P_1, P_2) = \gamma/\delta$. Hence if we keep track of the numerators and denominators separately, so that we are given a_1, b_1, a_2, $b_2 \in \mathbb{F}_{q^r}^*$ such that $k_1 = a_1/b_1$ and $k_2 = a_2/b_2$, then $k_3 = a_3/b_3$, where

$$a_3 = a_1 * a_2 * \gamma \text{ and } b_3 = b_1 * b_2 * \delta.$$

Notice that these equations hold both when $P_1 = P_2$ and $P_1 \neq P_2$. Hence, when P_1, P_2, $P_3 \neq \mathcal{O}$, the number of operations needed for computing a sum is given by $10\mathbf{M} + 6\mathbf{C} + \mathbf{E}$. Similarly, if P_1 or P_2 equals $\mathcal{O}$, then we simply set $\gamma = \delta = 1$, which yields a cost of $2\mathbf{M} + \mathbf{E}$ for computing (k_3, P_3). Lastly, if P_1, $P_2 \neq \mathcal{O}$ but $P_3 = \mathcal{O}$, then $\gamma = X_M \cdot Z_1 - X_1$ and $\delta = X_N \cdot Z_1 - X_1$, so the evaluation of (k_3, P_3) requires $4\mathbf{M} + 2\mathbf{C} + \mathbf{E}$. Of course, these costs should be seen as an *upper bound* rather than a precise account of the complexity.

Indeed, a careful analysis would first require to consider fields of characteristic two and of odd characteristic separately, and should *simultaneously* optimize the cost of an addition on E and the computation of k_3. A separate account should also be performed for the case $P_1 \neq P_2$ and for $P_1 = P_2$, since the equations for adding or doubling points on E differ. Then one would need to compare the results obtained for various coordinate systems, like the Jacobian (or weighted projective) coordinates, the Chudnovsky Jacobian coordinates as well as various mixed or redundant representations. Finally, a similar parallel inspection should be performed when protection against side-channel attacks is required.

Needless to say, this tedious analysis should be performed prior to any serious performance comparison with other cryptosystems. However, it would be premature to do so at this stage since the goal we are currently after is to establish the relevance of generalized Jacobians in cryptography.

5.4.2 Scalar Multiplications

In this section, we will assume that we fixed a smooth elliptic curve E over $\mathbb{F}_q$ together with a point $B \in E(\mathbb{F}_q)$ of prime order l to serve as our basepoint. Moreover, we will assume that a B-unrelated modulus $\mathfrak{m} = (M) + (N)$ was chosen such that M and N are distinct nonzero points of $E(\mathbb{F}_{q^r})$. By property 6 of Corollary 5.6, we know that $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is a finite subgroup of $J_\mathfrak{m}$. So given $(k, P) \in \mathbb{F}_{q^r}^* \times \langle B \rangle$ and a positive integer n, we are looking for an efficient way to compute the scalar multiple $n(k, P)$.

First notice that if $P = \mathcal{O}$, then $n(k, \mathcal{O}) = (k^n, \mathcal{O}) = (k^{n \bmod \mathrm{ord}(k)}, \mathcal{O})$ by property 4 of Corollary 5.6 . So computing this scalar multiple of $(k, \mathcal{O})$ simply requires to perform the discrete exponentiation $k^{n \bmod \mathrm{ord}(k)}$ in the finite field $\mathbb{F}_{q^r}$.

Now if $P \in \langle B \rangle \setminus \{\mathcal{O}\}$, then P has prime order l. In order to compute $n(k, P)$, the obvious remark is that a repeated application of the group law yields $n(k, P) = (*, nP)$. Thus if we set $n_0 = n \bmod l$, we get $n(k, P) = (*, n_0P)$. So instead of computing $n(k, P)$ directly, we could make use of the value of $n_0(k, P)$. Indeed, if we let $n_1 = \lfloor n/l \rfloor$, then $n = n_1 \cdot l + n_0$ and so $n(k, P) = n_1 l(k, P) + n_0(k, P)$. Therefore, if we let $l(k, P) = (\lambda, \mathcal{O})$ and $n_0(k, P) = (\nu_{n_0}, n_0P)$, we obtain

$$\begin{aligned} n(k, P) &= n_1 l(k, P) + n_0(k, P) \\ &= n_1(\lambda, \mathcal{O}) + (\nu_{n_0}, n_0P) \\ &= (\lambda^{n_1}, \mathcal{O}) + (\nu_{n_0}, n_0P) \\ &= (\nu_{n_0} \cdot \lambda^{n_1}, n_0P) \end{aligned} \tag{5.21}$$

by repeated applications of property 4 of Corollary 5.6. Hence evaluating $n(k, P)$ using this trick is essentially computing λ, λ^{n_1} and $n_0(k, P)$. So if several scalar multiples of the *same* element (k, P) need to be performed, then the value of λ may be precomputed in order to speed up the computations.

Lastly, notice that the really simple equality $n(k, P) = (\nu_{n_0} \cdot \lambda^{n_1}, n_0P)$ in fact relates three instances of the discrete logarithm problem in three different groups, namely a generalized Jacobian, an elliptic curve and a finite field. This expression therefore deserves to be studied in Section 5.5, where security matters will be addressed. For future reference, we now state it properly as a little lemma.

Lemma 5.8 *Let E be a smooth elliptic curve defined over $\mathbb{F}_q$, $B \in E(\mathbb{F}_q)$ of prime order l be given and $\mathfrak{m} = (M) + (N)$ be a B-unrelated modulus, where M and N are distinct nonzero points of $E(\mathbb{F}_{q^r})$. For $k \in \mathbb{F}_{q^r}^*$, $P \in \langle B \rangle \setminus \{\mathcal{O}\}$ and a positive integer n, let $l(k, P) = (\lambda, \mathcal{O})$ and $n_0(k, P) = (\nu_{n_0}, n_0P)$. Then,*

$$n(k, P) = (\nu_{n_0} \cdot \lambda^{n_1}, n_0P),$$

where $n_0 = n \bmod l$ and $n_1 = \lfloor n/l \rfloor$.

5.4.3 Choosing a Suitable Modulus

Since the beginning of this chapter, we have deduced various desirable properties of the public modulus $\mathfrak{m} = (M) + (N)$. Namely, we want $M = (X_M : Y_M : 1)$ and $N = (X_N : Y_N : 1)$ to be distinct nonzero points of $E(\mathbb{F}_{q^r})$ such that $\mathfrak{m} = (M) + (N)$ is a B-unrelated modulus.

Remark 5.9 *For efficiency, we have also previously raised the possibility of selecting some or all of the coordinates of M and N in such a way that multiplying a field element by those chosen constants is notably faster than computing a general multiplication. However, since it would take us too far afield to formalize the intuitive notion of 'notably faster multiplication', this supplementary requirement will not be taken into account in this section.*

The next step is to ensure that these requirements can be *efficiently* and *simultaneously* fulfilled. To do so, the prime power q and the positive integer r can first be fixed. For cryptographic applications, we usually consider elliptic curves defined over $\mathbb{F}_{2^s}$ or $\mathbb{F}_p$, where s is a positive integer and $p > 3$ is prime. Another possibility is to work over an optimal extension field, as described in [BP98].

If $\text{Char}(\mathbb{F}_q) = 2$, then non-supersingular[5] elliptic curves should be used in order to avoid the MOV attack [MOV93]. In that case, the elliptic curve can be taken to have a Weierstraß equation of the form

$$E_{a,b} : y^2 + xy = x^3 + ax^2 + b, \tag{5.22}$$

with $a, b \in \mathbb{F}_q$. The discriminant $\Delta = b$ must also be nonzero in order to guarantee that the curve be nonsingular.

If $\text{Char}(\mathbb{F}_q) = p \neq 2, 3$, the elliptic curve considered is given by

$$E_{a,b} : y^2 = x^3 + ax + b, \tag{5.23}$$

where $a, b \in \mathbb{F}_q$ and $\Delta = -16(4a^3 + 27b^2) \neq 0$.

Before we start to think about a good way of choosing a

B-unrelated modulus, some facts are worth mentioning.. First notice that random choices of M and N have the advantage of being less susceptible to future attacks targeted at moduli with special properties: it is thus a wise choice when long-term security is seeked. Moreover, when the parameters of a cryptosystem are generated by a third party, Tracy, then the possibility that they were specifically chosen such that Tracy possesses a trapdoor compromising the security of the system might become an issue. However, if the parameters are *verifiably pseudo-random*,

[5] Recall that an elliptic curve E over $\mathbb{F}_q$ is said to be supersingular if $\text{Char}(\mathbb{F}_q)$ divides $q + 1 - \#E(\mathbb{F}_q)$.

then it is very unlikely that Tracy knows such a trapdoor. A method for generating verifiably pseudo-random parameters for elliptic curves is described in the IEEE P1363 standard [IEE99, Sections A.12.4-A.12.7]. These two arguments show that even if pseudo-random moduli might not be the optimal choice for efficiency, they can provide security advantages over moduli with special properties.

So we now discuss how to efficiently generate an elliptic curve together with a pseudo-random B-unrelated modulus. Several methods are known to select an elliptic curve E over $\mathbb{F}_q$ with good cryptographic properties. The requirements on E as well as various techniques for choosing a suitable curve are discussed in Section 5.2 of the excellent survey [KMV00]. That Section also mentions heuristic arguments suggesting that pseudo-randomly choosing elliptic curves[6], until one fulfilling all criteria is found, is an efficient procedure to select a curve in characteristic 2 and p. It is moreover a simple matter to choose pseudo-random points on an elliptic curve, both in characteristic 2 and p. Such pseudo-codes can be found in Sections A.11.1 and A.11.2 of [IEE99]. The underlying idea is simply to successively generate pseudo-random values $x \in \mathbb{F}_{q^r}$, until there is a $y \in \mathbb{F}_{q^r}$ such that $(x, y) \in E(\mathbb{F}_{q^r})$. It is therefore a simple matter to efficiently generate pseudo-random M and N subject to the constraints M, $N \neq \mathcal{O}$ and $M \neq N$. The last step is to check whether $\mathfrak{m}$ is B-related or not.

A straightforward case arises when $r > 1$ and both M and N lie in $E(\mathbb{F}_{q^r})$, but are not in $E(\mathbb{F}_q)$. Indeed, for any $B \in E(\mathbb{F}_q)$, we have $\langle B \rangle \subseteq E(\mathbb{F}_q)$ and thus M, $N \notin \langle B \rangle$. We then conclude that $\mathfrak{m}$ is B-unrelated.

There is also an easy criterion to decide if the modulus is B-unrelated when at least one of M or N is a point of $E(\mathbb{F}_q)$. Recall that for cryptographic applications, it is recommended that $\#E(\mathbb{F}_q) = h \cdot l$, where l is a large prime and the cofactor h is small, while the order of $E(\mathbb{F}_q)$ can be determined using the Schoof-Elkies-Atkin (SEA) algorithm, which is outlined in [BSS99, Chapter VII]. In the standards for elliptic curve cryptography [IEE99, NIoST00, CR00], it is specified that h should equal $1, 2, 3$ or 4.

As usual, let $B \in E(\mathbb{F}_q)$ be a point of order l. If $h = 1$, then $\langle B \rangle = E(\mathbb{F}_q)$, which implies that the chosen modulus will be B-related as soon as one of M or N is in $E(\mathbb{F}_q)$. However, if $h > 1$, then at least half of the elements of $E(\mathbb{F}_q)$ are outside of $\langle B \rangle$. Thus when $h > 1$, pseudo-randomly choosing M and N until a B-unrelated modulus is encountered will be an efficient way to proceed, as long as we are able to quickly verify if M or N is a multiple of B.

Clearly, we have that

$$\textit{If } M \in \langle B \rangle \textit{, then } lM = \mathcal{O}.$$

[6] That is, selecting pseudo-random a and b in $\mathbb{F}_q$ that will determine the elliptic curve $E_{a,b}$.

For the elliptic curve we consider, it turns out that the converse also holds. Indeed, by the Structure Theorem for Finitely Generated Abelian Groups [Hun74, Theorem II.2.2], we know that $E(\mathbb{F}_q)$ is isomorphic to a direct sum of the form $\mathbb{Z}/l\mathbb{Z} \oplus G$, where $\mathbb{Z}/l\mathbb{Z}$ is the additive group of integers modulo l and G is a group of order h. Since l is prime, it follows that the only elements of order l in $E(\mathbb{F}_q)$ must lie in $\langle B\rangle$. We have therefore shown that if $M \in E(\mathbb{F}_q)$, then

$$lM = \mathcal{O} \textit{ if and only if } M \in \langle B\rangle .$$

We thus have an easy and efficient method to decide if a given modulus is B-unrelated. Alternatively, one could also use the Weil pairing in order to achieve the same goal. Indeed, back in 1986, Victor Miller noticed that the Weil pairing provides a solution to the subgroup membership problem on elliptic curves[7]. The following proposition, whose proof can be found in [Gal04, Section 8], provides an efficient method to decide if a point Q lies in the subgroup generated by P.

Proposition 5.10 *Let E be a smooth elliptic curve defined over a finite field $\mathbb{F}_q$, $m \geq 2$ be an integer prime to* $\mathrm{Char}(\mathbb{F}_q)$*, $P \in E(\overline{\mathbb{F}}_q)$ be a point of order m and $Q \in E[m]$ be given. Then,*

$$Q \in \langle P\rangle \textit{ if and only if } e_m(P, Q) = 1, \tag{5.24}$$

where $e_m : E[m] \times E[m] \rightarrow \mu_m$ is the classical Weil pairing, $\mu_m = \left\{\zeta \in \mathbb{F}^*_{q^k} \middle| \zeta^m = 1\right\}$ is the subgroup of m^{th} roots of unity and k is the smallest positive integer[8] satisfying $m \,\big|\, (q^k - 1)$.

Remark 5.11 *This proposition uses the original version of the Weil pairing[9] as defined in [Sil86, Section III.8], as opposed to the modified pairings exploited in several cryptographic applications, such as for the identity-based encryption scheme of Boneh and Franklin [BF01, BF03]. An easy way to remember which pairing to use here is to note that for property (5.24) to hold, we must have $e_m(P, P) = 1$. For this same reason, the above proposition might not hold for the Tate-Lichtenbaum pairing either, since it is not necessarily alternating.*

In his now famous unpublished manuscript[10] *"Short Programs for Functions on Curves"*, Miller presents a fast probabilistic polynomial-time algorithm to compute the Weil pairing. In the recent special issue *"Pairings and their Use in Cryptology"* of the Journal of Cryptology, Miller also signed an article concerned with the efficient calculation of this pairing [Mil04].

[7] This observation is in fact one of the ideas underlying his algorithm for determining the group structure of $E(\mathbb{F}_q)$, which is described in [Mil86a, Algorithm 3].

[8] This integer k is sometimes called the *embedding degree*, *MOV degree*, or *security multiplier*.

[9] This pairing was introduced by André Weil in 1940 and was used in his first proof of the Riemann Hypothesis for curves over finite fields [Wei40].

[10] Which is now available online (see [Mil86a, Algorithm 2]).

To sum up, we now know how to efficiently generate a B-unrelated modulus, simply by choosing random affine points $M \neq N$ on the curve[11] until both M and N are outside $\langle B \rangle$. In order to test if a given point lies in the subgroup generated by B, two methods were also mentioned.. From a global perspective, this means that we now have all the tools at hand to select the generalized Jacobians we are considering for the cryptographic applications of this chapter.

5.4.4 Group Order and Generators

For a given elliptic curve E over $\mathbb{F}_q$, a basepoint $B \in E(\mathbb{F}_q)$ of prime order l, and a B-unrelated modulus $\mathfrak{m} = (M) + (N)$ such that M and N are distinct nonzero points of $E(\mathbb{F}_{q^r})$, we now wish to make some observations concerning the order of the elements in $\mathbb{F}_{q^r}^* \times \langle B \rangle$, which will then be used to study the structure of this group. In what follows, the prime l is assumed to be known[12].

ORDER OF THE ELEMENTS. Given any element $(k, P) \in \mathbb{F}_{q^r}^* \times \langle B \rangle$ for which $P \neq \mathcal{O}$, we first want to efficiently compute its order m. By definition, m is the least positive integer such that $m(k, P) = (1, \mathcal{O})$. Since $m(k, P) = (*, mP)$, we have that $mP = \mathcal{O}$, and thus that m is a multiple of $l = \operatorname{ord}(P)$. There is then a positive integer n such that $m = n \cdot l$. Hence,

$$(1, \mathcal{O}) = m(k, P) = n \cdot l(k, P) = n(\lambda, \mathcal{O}) = (\lambda^n, \mathcal{O}),$$

where $\lambda \in \mathbb{F}_{q^r}^*$ satisfies $l(k, P) = (\lambda, \mathcal{O})$. It thus follows that $\lambda^n = 1$, for which the least solution is $n = \operatorname{ord}(\lambda)$. As a result,

$$\textit{The order of } (k, P) \textit{ equals } \operatorname{ord}(\lambda) \cdot l.$$

So in particular,

$$(k, P) \textit{ generates } \mathbb{F}_{q^r}^* \times \langle B \rangle \textit{ if and only if } \lambda \textit{ generates } \mathbb{F}_{q^r}^*. \tag{5.25}$$

Since l is already known, it only remains to determine $\operatorname{ord}(\lambda)$ if we wish to compute $\operatorname{ord}((k, P))$. First notice that computing λ can be done by evaluating the scalar multiple $l(k, P)$, since both l and (k, P) are known. Moreover, recall that determining the order of an element in a finite group G can be readily achieved when the factorization of $\#G$ is known. The corresponding deterministic algorithm can be found in [MvOV96, Algorithm 4.79]. As a result, the order of an element of $\mathbb{F}_{q^r}^* \times \langle B \rangle$ can be efficiently computed as soon as the factorization of $q^r - 1$ is known.

[11] Remembering that the underlying elliptic curve should have a cofactor greater than one when $r = 1$.

[12] Recall that $\#E(\mathbb{F}_q)$ can be efficiently computed and was chosen such that its factorization is of the form $l \cdot h$, where h is a small integer.

STRUCTURE OF $\mathbb{F}_{q^r}^* \times \langle B \rangle$. Next we turn our attention to the structure of $\mathbb{F}_{q^r}^* \times \langle B \rangle$. We first point out that this group has order $(q^r - 1) \cdot l$, and will therefore *never* have prime order. Consequently, $\mathbb{F}_{q^r}^* \times \langle B \rangle$ might not be a cyclic group. Fortunately, it is possible to investigate a little further by taking a closer look at (5.25). Indeed, let g be a generator of $\mathbb{F}_{q^r}^*$ and let α be the element of $\mathbb{F}_{q^r}^*$ such that $l(1, P) = (\alpha, \mathcal{O})$. In order to know if $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is cyclic, we can start by explicitly write down the values of $l(1,P), l(g,P), l(g^2,P), \ldots, l(g^{q^r-2}, P)$ in terms of α. We have

$$l\left(g^i, P\right) = l\left(\left(g^i, \mathcal{O}\right) + (1, P)\right) = l\left(g^i, \mathcal{O}\right) + l\left(1, P\right) = \left(g^{il}, \mathcal{O}\right) + (\alpha, \mathcal{O}) = (\alpha g^{il}, \mathcal{O}),$$

for any integer i such that $0 \leq i < q^r - 1$. We would therefore like to know if there is a generator of $\mathbb{F}_{q^r}^*$ among $\alpha, \alpha g^l, \alpha g^{2l}, \ldots, \alpha g^{(q^r-2)l}$. Now notice that as soon as $l \nmid (q^r - 1)$, we have that

$$\alpha g^{il} = \alpha g^{jl} \text{ iff } il \equiv jl \,(\operatorname{mod} q^r - 1) \text{ iff } i \equiv j \,(\operatorname{mod} q^r - 1) \text{ iff } i = j,$$

where $0 \leq i, j < q^r - 1$. Therefore, the $q^r - 1$ elements $\alpha, \alpha g^l, \alpha g^{2l}, \ldots, \alpha g^{(q^r-2)l}$ of $\mathbb{F}_{q^r}^*$ are all distinct, which means that $\mathbb{F}_{q^r}^* = \left\{\alpha, \alpha g^l, \alpha g^{2l}, \ldots, \alpha g^{(q^r-2)l}\right\}$, and thus that this set must contain a generator of $\mathbb{F}_{q^r}^*$. Finally, we use (5.25) to conclude that $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is cyclic whenever $l \nmid (q^r - 1)$.

If $l \mid (q^r - 1)$, then the above counting argument no longer holds. Indeed, the order of α now comes into play. By hypothesis, we know that $q^r - 1 = ld$ for some positive integer d. Thus the order of α divides $l \cdot d$. If we are in the situation where $\alpha^d = 1$, then $\left(\alpha g^{il}\right)^d = 1$ for $0 \leq i < q^r - 1$, which means that there is no generator of $\mathbb{F}_{q^r}^*$ among $\alpha, \alpha g^l, \alpha g^{2l}, \ldots, \alpha g^{(q^r-2)l}$. This simple observation shows that the behavior of $\mathbb{F}_{q^r}^* \times \langle B \rangle$ may be different when $l \mid (q^r - 1)$, and thus that a further study of this case would be of interest.

Remark 5.12 *It would also be possible to work in a proper cyclic subgroup of $\mathbb{F}_{q^r}^* \times \langle B \rangle$. However, an advantage of considering all of $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is that the plaintext imbedding of a message is then readily achieved. Efficient methods for imbedding plaintexts in an elliptic curve can be found in [Kob87, Section 3].*

FINDING A GENERATOR. Lastly, given a cyclic group $\mathbb{F}_{q^r}^* \times \langle B \rangle$ for which the factorization of the group order $(q^r - 1) \cdot l$ is known, we describe how a generator of this group can be efficiently selected. First recall that for any cyclic group $G = \langle g \rangle$ of order n and integer $1 \leq i \leq n$,

$$g^i \text{ is a generator of } G \text{ if and only if } \gcd(n, i) = 1. \tag{5.26}$$

There are therefore exactly $\phi(n)$ generators of G, where ϕ is Euler's totient function (see Section 2.4.2). So if we choose a random element of G, the probability that it is a primitive element is

$\phi(n)/n$. But $\phi(n) > n/(6\ln\ln n)$ as soon as $n \geq 5$ [RS62]. Therefore, the probability that a randomly chosen element of G be a generator is at least $1/(6\ln\ln n)$. As a result, successively choosing random elements of G until a generator is found is an efficient (expected polynomial-time) method to select a primitive element, as soon as there is an efficient deterministic procedure to decide if a given element is of maximal order. Thus if the factorization of $q^r - 1$ is known, we conclude that randomly choosing elements of $\mathbb{F}_{q^r}^* \times \langle B \rangle$ until a generator is found is an efficient (expected polynomial-time) procedure to obtain a primitive element of $\mathbb{F}_{q^r}^* \times \langle B \rangle$.

Remark 5.13 *Notice that if (k, P) is a generator of $\mathbb{F}_{q^r}^* \times \langle B \rangle$, then it does not imply that k is a generator of $\mathbb{F}_{q^r}^*$. Indeed, recall that in our toy example of Section 5.3.3, we saw that the element $(1, B)$ was a primitive element of $\mathbb{F}_7^* \times \langle B \rangle$.*

5.5 The Discrete Logarithm Problem

Among the four essential ingredients needed for a group to be suitable for DL-based cryptography outlined at the beginning of this chapter, we have so far covered three of them. Namely, we now know how to compactly represent the elements of our generalized Jacobian, how to efficiently perform the group operation and how to compute the group order. Thus, the very last step is to study the discrete logarithm problem in $\mathbb{F}_{q^r}^* \times \langle B \rangle$.

Throughout this section, E will as usual denote a smooth elliptic curve defined over $\mathbb{F}_q$, $B \in E(\mathbb{F}_q)$ a point of prime order l, $\mathfrak{m} = (M) + (N)$ a B-unrelated modulus with M and N distinct nonzero points of $E(\mathbb{F}_{q^r})$. Finally, we will assume that these parameters have been chosen such that $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is a cyclic subgroup of $J_\mathfrak{m}$ generated by (k, P).

5.5.1 A Natural Solution

The first exercise that should be done in order to get a flavor of how to attack the discrete logarithm problem in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is to try to write down the most natural way we could see to solve this problem. If we do so, we will then have an upper bound on the complexity of the problem that will (hopefully) raise several relevant questions concerning the overall difficulty of the problem.

Thus, given an element $(j, Q) \in \mathbb{F}_{q^r}^* \times \langle B \rangle$, we wish to determine the least positive integer n such that $n(k, P) = (j, Q)$. Notice that such a n exists since $\langle (k, P) \rangle = \mathbb{F}_{q^r}^* \times \langle B \rangle$ by hypothesis. While considering efficiency aspects in the previous section, recall that we came up with an interesting way of computing scalar multiples of (k, P). Indeed, by Lemma 5.8 , we have that

$$(j, Q) = n(k, P) = (\nu_{n_0} \cdot \lambda^{n_1}, n_0 P),$$

where $n_0 = n \bmod l$, $n_1 = \lfloor n/l \rfloor$, $l(k, P) = (\lambda, \mathcal{O})$ and $n_0(k, P) = (\nu_{n_0}, n_0 P)$. Thus, given the values of

$$j = \nu_{n_0} \cdot \lambda^{n_1} \text{ and } Q = n_0 P,$$

recovering n is the goal of this game. First notice that knowing n is equivalent to know both n_0 and n_1 (since l is public and $n = n_1 \cdot l + n_0$). Also observe that Q is independent of n_1 while j depends on both n_0 and n_1.

The obvious strategy is then to start by solving an instance of the discrete logarithm problem on E in order to recover n_0 from $Q = n_0 P$. Once n_0 is known, the value of ν_{n_0} can be easily computed, as $n_0(k, P) = (\nu_{n_0}, n_0 P)$. Next derive the value of λ^{n_1} by computing $\nu_{n_0}^{-1} \cdot j$ (notice that $\nu_{n_0} \neq 0$ since by construction, $\nu_{n_0} \in \mathbb{F}_{q^r}^*$). Then recover n_1 by computing the discrete logarithm of λ^{n_1} to the base λ. Finally, let $n = n_1 \cdot l + n_0$. We have therefore shown:

Lemma 5.14 *Let E be a smooth elliptic curve over $\mathbb{F}_q$, $B \in E(\mathbb{F}_q)$ be a point of prime order, $\mathfrak{m} = (M) + (N)$ be a B-unrelated modulus, where M and N are distinct nonzero points of $E(\mathbb{F}_{q^r})$ such that $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is a cyclic subgroup of $J_\mathfrak{m}$. Then, the discrete logarithm problem in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is no harder than* ***sequentially*** *solving a discrete logarithm in E followed by one in $\mathbb{F}_{q^r}^*$.*

In a nutshell, the computing sequence that was performed in order to extract n can be visualized as follows:

$$\boxed{n_0 P \;\underset{\text{in } E}{\overset{\text{DLP}}{\Rrightarrow}}\; n_0 \;\to\; \nu_{n_0} \;\to\; \lambda^{n_1} \;\underset{\text{in } \mathbb{F}_{q^r}^*}{\overset{\text{DLP}}{\Rrightarrow}}\; n_1 \;\to\; n}$$

Figure 5.3: Natural solution to a DLP on the generalized Jacobian

In this figure, the triple arrow '$\Rrightarrow$' emphasize that this step requires to perform a discrete logarithm, while the simple arrow '$\to$' means that this computation can be efficiently performed. There are therefore two bottlenecks in this solution: one for each DLP to be solved. Unfortunately, the most obvious solution to a problem needs not coincide with the optimal strategy, so we have to wonder:

Is it possible to do any better?

The remainder of this chapter will attempt to answer this intuitive question. Since providing a clear answer to a vague question is to no avail, the first step is to draw up a list of (still informal) subquestions of interest:

- *If we know how to solve the discrete logarithm problem in* $\mathbb{F}_{q^r}^* \times \langle B \rangle$*, do we necessarily know how to solve it in* E*?*
- *If we know how to solve the discrete logarithm problem in* $\mathbb{F}_{q^r}^* \times \langle B \rangle$*, do we necessarily know how to solve it in* $\mathbb{F}_{q^r}^*$*?*
- *Is it possible to solve a discrete logarithm problem in* $\mathbb{F}_{q^r}^* \times \langle B \rangle$ *by solving one in* E *and one in* $\mathbb{F}_{q^r}^*$ *in* ***parallel****?*
- *Can some clever precomputations be made in order to speed up the extraction of a discrete logarithm in* $\mathbb{F}_{q^r}^* \times \langle B \rangle$*?*

This list is far from being exhaustive and of course, several auxiliary questions will arise along the way. Before we take a closer look at them, we conclude this section with an analogy that has proved to be useful in Chapter 2 for providing a mental image of the inner workings of a cryptographic technique.

Indeed, the process of sequentially performing two discrete logarithms in two different structures, that arise naturally with this generalized Jacobian, has a simple conceptual interpretation in terms of padlocks and safes. With the above notation, suppose that Eve wishes to recover Bob's private key n from his public key (j, Q), with the help of the publicly available data (k, P) and λ. As observed above, the knowledge of n is equivalent to the knowledge of both n_0 and n_1, where

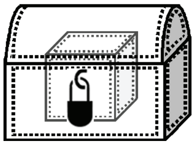
$$n_0 = \log_P Q \text{ and } n_1 = \log_\lambda \left(\nu_{n_0}^{-1} \cdot j\right).$$

Notice that the first discrete logarithm is in the elliptic curve E while the second is in the finite field $\mathbb{F}_{q^r}^*$. Thus, Bob in fact possesses two private keys: n_0 for the elliptic curve and n_1 for the finite field. Recall that with the simple model of a public-key cryptosystem described in Section 2.4.1, the combination used to open the safe played the role of the private key. Thus one possibility here would be to consider two safes, S_0 and S_1, with respective secret combinations n_0 and n_1.

If the two safes were side by side, then Eve and her evil friend Ed could *simultaneously* try to open both safes at the same time, thus recovering n_0 and n_1 is parallel. But as depicted in Figure 5.3, Eve's straightforward strategy is to first recover n_0 from n_0P, and then use this value to discover her second challenge λ^{n_1}, form which she gets n_1. That suggests that the safe S_1 should instead be placed *inside* of S_0.

This interpretation correctly suggests that the obvious strategy is to first open the outer safe, S_0, which protects the inner safe, S_1. Also notice that this physical model does not rule out the possibility that there might exist a smarter way to proceed (for example, if the lock of the inner

Simple Model for a Public-key Cryptosystem with Two Safes

Alice		***Bob***
Put message m in safe S_1 and lock it Put S_1 within the safe S_0 Lock S_0 and send it to Bob	 $\longrightarrow$	Open S_0 to recover the closed safe S_1 Unlock S_1 and retrieve m

safe is controlled by an electromagnet while the lock of the outer safe is purely mechanical, then cutting the current could unlock the inner safe before the outer safe is opened). But enough Hollywood scenarios, as we now need to seriously study this discrete logarithm.

5.5.2 Reductions among Discrete Logarithm Problems

In the previous section, we have seen that a natural approach to solve a discrete logarithm in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is to extract a discrete logarithm in E followed by one in $\mathbb{F}_{q^r}^*$. We would now like to say more about the interrelation between these three problems.

Loosely speaking, the goal we are after in this section is to show that any given algorithm that solves DLPs in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ may be used as a subroutine to solve DLPs in E as well as in $\mathbb{F}_{q^r}^*$. That means that if anyone ever discovers an efficient way to solve DLPs in $\mathbb{F}_{q^r}^* \times \langle B \rangle$, he or she could use it to efficiently solve instances of the DLP in E and $\mathbb{F}_{q^r}^*$, rendering obsolete all cryptographic protocols based on the discrete logarithm problem in these groups.

Before we start, it will be best to go over two important properties of discrete logarithms that will be used to prove the results of this section. For this purpose, let G be any (multiplicatively written) cyclic group of order n generated by an element g.

We begin with the *random self-reducible* property of discrete logarithms, which is based on the equality

$$g^a \cdot g^r = g^{a+r}. \tag{5.27}$$

We say that an algorithm $\mathcal{A}$ has a *non-negligible probability of solving the DLP in G* (to the base g) if for an input h uniformly chosen at random in G, there is a non-negligible probability[13] that $\mathcal{A}$ outputs $\log_g h$. But in practice, it is often desirable to learn the discrete logarithm of a *specific* element s of the group. It is however possible that the probability that $\mathcal{A}$ yields $a = \log_g s$ on input s equals zero[14]. The strategy is then to *'disguise'* s using equation (5.27). Indeed, if we

[13] That is, there is a polynomial p such that this probability is greater than $1/p(\log n)$.

[14] For instance, the algorithm could solve all instances for which the discrete logarithm is even, but fail otherwise.

uniformly pick an integer r in $\{0, 1, \ldots, n-1\}$, then $s \cdot g^r = g^a \cdot g^r = g^{a+r}$ (Take note that if r is uniformly selected, then so is $a+r$). So on input $s \cdot g^r$, there is now a non-negligible probability that $\mathcal{A}$ yields the value of $a + r$. If so, then a can be recovered since r is known. Thus, $\mathcal{A}$ implies the existence of a randomized algorithm $\mathcal{A}'$ such that for *any* input $s \in G$, there is a non-negligible probability that $\mathcal{A}'$ outputs $\log_g s$.

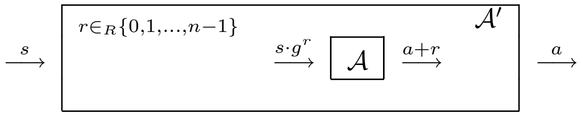


Figure 5.5: Constructing $\mathcal{A}'$ from $\mathcal{A}$

The second property concerns the choice of the generator of the group. Namely, if g_1 and g_2 are distinct generators of G, then any algorithm $\mathcal{A}_1$ that has a non-negligible probability of solving discrete logarithms in G to the base g_1 can readily be turned into an algorithm $\mathcal{A}_2$ having non-negligible probability of solving discrete logarithms in G to the base g_2. Indeed, let $h = g_2^a$ be an instance of the DLP in G to be solved. By the random self-reducible property of discrete logarithms, we can assume without loss of generality that for any $s \in G$, $\mathcal{A}_1$ has a non-negligible probability of producing $\log_{g_1} s$. So first invoke $\mathcal{A}_1$ on input g_2 in order to get, with non-negligible probability, an integer b such that $g_2 = g_1^b$ and $0 < b < n$. Since g_1 and g_2 are both generators, it follows that $\gcd(n, b) = 1$ (see Fact (5.26 on page 147)), and so b is an invertible element of $(\mathbb{Z}/n\mathbb{Z})^*$. Then compute an integer c such that $bc \equiv 1 \pmod{n}$ and $0 < c < n$ using, for instance, the extended Euclidean algorithm [MvOV96, Algorithm 2.107]. Then,

$$g_2^c = \left(g_1^b\right)^c = g_1^{bc} = g_1.$$

Next, we can obtain with non-negligible probability an integer d such that $h = g_1^d$ and $0 \leq d < n$ by invoking $\mathcal{A}_1$ on input h. Finally,

$$h = g_1^d = (g_2^c)^d = g_2^{cd},$$

and so $a = cd \bmod n$, which completes the argument.

The link between the DLP in the generalized Jacobian and in the elliptic curve appearing to be simple, we might want to analyze it first. We therefore want to show:

Lemma 5.15 *Let E be a smooth elliptic curve over $\mathbb{F}_q$, $B \in E(\mathbb{F}_q)$ be a point of prime order l, $\mathfrak{m} = (M) + (N)$ be a B-unrelated modulus, where M and N are distinct nonzero points of*

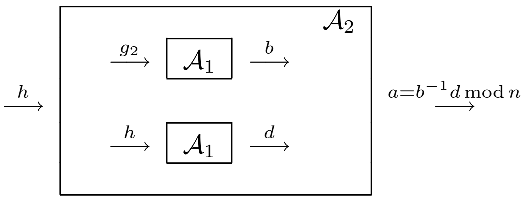


Figure 5.6: Constructing $\mathcal{A}_2$ from $\mathcal{A}_1$

$E(\mathbb{F}_{q^r})$ such that $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is a cyclic subgroup of $J_\mathfrak{m}$. Then, the discrete logarithm problem in $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is at least as hard as the discrete logarithm problem in $\langle B \rangle \subseteq E(\mathbb{F}_q)$.

Proof. Let $\mathcal{A}_{J_\mathfrak{m}}$ be an algorithm that has a non-negligible probability of solving discrete logarithms in $\mathbb{F}^*_{q^r} \times \langle B \rangle$ to the base (k, P), where (k, P) is a generator of $\mathbb{F}^*_{q^r} \times \langle B \rangle$. We wish to show that there is an algorithm $\mathcal{A}_E$ having a non-negligible probability of solving discrete logarithms in $\langle B \rangle$ to the base P. So let $Q = n_0 P$ be an instance of the discrete logarithm problem in $\langle B \rangle$, where $0 \leq n_0 < l$. By the random self-reducible property of discrete logarithms, we can assume without loss of generality that given any element of $\mathbb{F}^*_{q^r} \times \langle B \rangle$, its discrete logarithm (to the base (k, P)) has a non-negligible probability of being obtained with $\mathcal{A}_{J_\mathfrak{m}}$. Now, for a randomly chosen element $j \in \mathbb{F}^*_{q^r}$, invoke $\mathcal{A}_{J_\mathfrak{m}}$ on input (j, Q). With non-negligible probability, a non-negative integer n such that $n(k, P) = (j, Q)$ will be obtained, yielding $n_0 = n \bmod l$. □

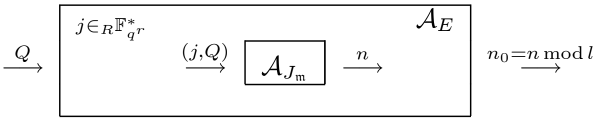


Figure 5.7: Converting an instance of the DLP in $\langle B \rangle$ into one in $\mathbb{F}^*_{q^r} \times \langle B \rangle$

Next we want to show a similar reduction between the discrete logarithm problem in $\mathbb{F}^*_{q^r} \times \langle B \rangle$ and in $\mathbb{F}^*_{q^r}$. Notice how this proof differs from the previous one, even though the same underlying technique is used in both proofs.

Lemma 5.16 *Let E be a smooth elliptic curve over $\mathbb{F}_q$, $B \in E(\mathbb{F}_q)$ be a point of prime order l, and $\mathfrak{m} = (M) + (N)$ be a B-unrelated modulus, where M and N are distinct nonzero points of $E(\mathbb{F}_{q^r})$ such that $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is a cyclic subgroup of $J_\mathfrak{m}$. Then, the discrete logarithm problem in $\mathbb{F}^*_{q^r} \times \langle B \rangle$ is at least as hard as the discrete logarithm problem in $\mathbb{F}^*_{q^r}$.*

Proof. Let $\mathcal{A}_{J_\mathfrak{m}}$ be an algorithm that has a non-negligible probability of solving discrete logarithms in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ to the base (k, P), where (k, P) is a generator of $\mathbb{F}_{q^r}^* \times \langle B \rangle$. We want to show the existence of an algorithm $\mathcal{A}_{\mathbb{F}_{q^r}^*}$ having a non-negligible probability of solving discrete logarithms in $\mathbb{F}_{q^r}^*$ to the base g, where g is a generator of $\mathbb{F}_{q^r}^*$. Thus let $h = g^n$ be an instance of the discrete logarithm problem in $\mathbb{F}_{q^r}^*$, with $0 \leq n < q^r - 1$. As usual, we can assume without loss of generality that given any element of $\mathbb{F}_{q^r}^* \times \langle B \rangle$, its discrete logarithm (to the base (k, P)) has a non-negligible probability of being obtained with $\mathcal{A}_{J_\mathfrak{m}}$ (by the random self-reducible property). Invoking $\mathcal{A}_{J_\mathfrak{m}}$ twice, on inputs $(g, \mathcal{O})$ and $(h, \mathcal{O})$, will yield with non-negligible probability integers a and b satisfying $(g, \mathcal{O}) = a\,(k, P)$, $(h, \mathcal{O}) = b\,(k, P)$ and $0 \leq a, b < (q^r - 1)\,l$. Notice that l must divide both a and b, so there are integers c and d such that $a = c \cdot l$, $b = d \cdot l$ and $0 \leq c, d < (q^r - 1)$.

If we now let $l\,(k, P) = (\lambda, \mathcal{O})$, then λ has to be a generator of $\mathbb{F}_{q^r}^*$ by (5.25). We then have $g = \lambda^c$ since

$$(g, \mathcal{O}) = a\,(k, P) = c \cdot l\,(k, P) = c(\lambda, \mathcal{O}) = (\lambda^c, \mathcal{O}).$$

Moreover, both g and λ generates $\mathbb{F}_{q^r}^*$, from which follows that $\gcd(c, q^r - 1) = 1$ (by (5.26)). Lastly, by property 4 of Corollary 5.6, we have

$$(\lambda^d, \mathcal{O}) = d\,(\lambda, \mathcal{O}) = d \cdot l\,(k, P) = b\,(k, P) = (h, \mathcal{O}) = (g^n, \mathcal{O}) = n(g, \mathcal{O}) = n(\lambda^c, \mathcal{O}) = (\lambda^{cn}, \mathcal{O}),$$

and we finally get $n = c^{-1} d \bmod (q^r - 1)$. □

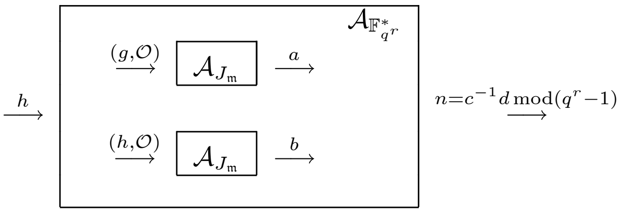

$\mathcal{A}_{\mathbb{F}_{q^r}^*}$
$\xrightarrow{h}$
$\xrightarrow{(g,\mathcal{O})}$
$\mathcal{A}_{J_\mathfrak{m}}$
$\xrightarrow{a}$
$\xrightarrow{(h,\mathcal{O})}$
$\mathcal{A}_{J_\mathfrak{m}}$
$\xrightarrow{b}$
$\xrightarrow{n=c^{-1}d \bmod (q^r-1)}$

Figure 5.8: Converting an instance of the DLP in $\mathbb{F}_{q^r}^*$ into two instances in $\mathbb{F}_{q^r}^* \times \langle B \rangle$

From a practical point of view, the two lemmas of this section imply that even though generalized Jacobians are newcomers in cryptography, we already know that solving their DLP cannot be easier than solving discrete logarithms in two of the most studied groups used in DL-based cryptography today.

5.5.3 Precomputations and Parallelization

Now that we have strong evidence that the discrete logarithm problem in the generalized Jacobians we consider is a computationally difficult problem, we further investigate the natural solution proposed in Section 5.5.1. Recall that Lemma 5.14 showed that an instance of the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ can be solved by sequentially extracting a discrete logarithm in E followed by one in $\mathbb{F}_{q^r}^*$. So the next step is to try to determine under which circumstances the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ could be performed any faster.

As usual, let $(j, Q) = n(k, P)$ be an instance of the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ to be solved, where $0 \leq n < (q^r - 1)\, l$. By Lemma 5.8, we know that the scalar multiple $n(k, P)$ can be computed as

$$n(k, P) = (\nu_{n_0} \cdot \lambda^{n_1}, n_0 P),$$

where we keep the notation $n = n_1 \cdot l + n_0$, $0 \leq n_0 < l$, $0 \leq n_1 < q^r - 1$ as well as $l(k, P) = (\lambda, \mathcal{O})$ and $n_0(k, P) = (\nu_{n_0}, n_0 P)$. Notice that the sequential solution of Section 5.5.1 performs computations involving $\nu_{n_0} \cdot \lambda^{n_1}$ *only* once ν_{n_0} is known, which can be pictured as follows.

$\mathbb{F}_{q^r}^*$	E
	$n_0 P$
	$\downarrow$
	n_0
	$\downarrow$
$\nu_{n_0} \cdot \lambda^{n_1}$	ν_{n_0}
$\downarrow$	
λ^{n_1}	
$\downarrow$	
n_1	

Figure 5.9: A sequential solution to the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$

We could instead attempt to extract a discrete logarithm in $\mathbb{F}_{q^r}^*$ in *parallel* with the one in the elliptic curve. On one hand, using the identity $(j, Q) = n(k, P) = (\nu_{n_0} \cdot \lambda^{n_1}, n_0 P)$, one can start to solve $Q = n_0 P$ for n_0 by extracting a discrete logarithm in E.

In the meantime, we can also start to extract a discrete logarithm in the finite field as follows. This time, let

$$n_2 = n \bmod (q^r - 1)\,.$$

Then compute $l\,(j, Q)$ which will equal, say, $(j', \mathcal{O})$. We now have:

$$(j', \mathcal{O}) = l\,(j, Q) = l \cdot n\,(k, P) = n \cdot l\,(k, P) = n\,(\lambda, \mathcal{O}) = (\lambda^n, \mathcal{O}) = (\lambda^{n_2}, \mathcal{O})\,.$$

Since j' and λ are known, we can then solve the following DLP in $\mathbb{F}_{q^r}^*$ in order to get n_2:

$$j' = \lambda^{n_2}.$$

Remark that this can be done in *parallel* with the computation of n_0.

Finally, try to combine n_0 and n_2 using the Chinese remainder theorem in order to recover n. However, we must have $\gcd(l, q^r - 1) = 1$ in order to fully recover n with this method. Notice the similarity with the Pohlig-Hellman method.

Let's now see what the situation is when $l \mid q^r - 1$. Thus, the order of our generalized Jacobian $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is of the form $d \cdot l^\alpha$, where $\alpha \geq 2$ and $l \nmid d$. For cryptographic purposes, recall that we can think of l as being a 160-bit prime and q^r to have roughly 1024 bits.

Now just as before, let $(j, Q) = n(k, P)$ be the instance of the discrete logarithm problem that we wish to solve. In order to be able to use the Chinese remainder theorem to recover n in this case, we will need to know

$$\begin{cases} n_\alpha := n \bmod l^\alpha \\ n_d := n \bmod d \end{cases}.$$

This can be achieved in several steps as follows.

1. Let's begin with the easy part. That is, the computation of n_d. Start by computing $l^\alpha(j, Q)$ which will equal, say, $(j', \mathcal{O})$. Then we get

$$\begin{aligned} (j', \mathcal{O}) &= l^\alpha(j, Q) = l^\alpha n(k, P) = n l^{\alpha-1} \cdot l(k, P) \\ &= n l^{\alpha-1}(\lambda, \mathcal{O}) = \left(\left(\lambda^{l^{\alpha-1}}\right)^n, \mathcal{O}\right) = \left(\left(\lambda^{l^{\alpha-1}}\right)^{n_d}, \mathcal{O}\right), \end{aligned}$$

   which means that

$$j' = \left(\lambda^{l^{\alpha-1}}\right)^{n_d},$$

   where j' and $\lambda^{l^{\alpha-1}}$ are known. It thus suffices to solve a DLP in $\mathbb{F}_{q^r}^*$ in order to recover n_d.

2. Next we want to determine n_α. To do so, first let $n_0 := n \bmod l$ $(= n_\alpha \bmod l)$. To get n_0, we proceed the obvious way:

$$(j, Q) = n(k, P) = (*, nP) = (*, n_0 P),$$

   and we thus have $Q = n_0 P$, which requires to solve a DLP in the elliptic curve. Notice that we have now retrieved *all* the information about n that Q contained. That is, we should expect that *all* other discrete logs that we have to solve from this point on will be in the finite field $\mathbb{F}_{q^r}^*$. Very well.

3. Then we will determine
$$n_1 := \frac{n - n_0}{l} \bmod l.$$
This is where it gets interesting. Indeed, rewrite n as $n_0 + n_1 l + ml^2$ for some (unknown) integer m and compute $dl^{\alpha-2}(j, Q)$ to get, say, $\left(j'', dl^{\alpha-2}Q\right)$. Now remark that
$$\begin{aligned}\left(j'', dl^{\alpha-2}Q\right) = dl^{\alpha-2}(j, Q) = dl^{\alpha-2} \cdot n\,(k, P) &= dl^{\alpha-2}\left(n_0 + n_1 l + ml^2\right)(k, P)\\ &= dl^{\alpha-2} n_0\,(k, P) + n_1 dl^{\alpha-2} \cdot l\,(k, P) + m \cdot dl^{\alpha}\,(k, P)\\ &= dl^{\alpha-2}\left(\nu_{n_0}, n_0 P\right) + n_1 dl^{\alpha-2}(\lambda, \mathcal{O}) + m\,(1, \mathcal{O})\\ &= \left(\left(\nu_{n_0}\right)^{dl^{\alpha-2}} \cdot \mu, dl^{\alpha-2}Q\right) + \left(\left(\lambda^{dl^{\alpha-2}}\right)^{n_1}, \mathcal{O}\right) + (1, \mathcal{O})\\ &= \left(\left(\nu_{n_0}\right)^{dl^{\alpha-2}} \cdot \mu \cdot \left(\lambda^{dl^{\alpha-2}}\right)^{n_1}, dl^{\alpha-2}Q\right),\end{aligned}$$
where μ is simply the product of the 2-cocycles from repeated applications of the group law. Notice that μ can be computed directly from Q and $dl^{\alpha-2}$. It therefore follows that
$$\frac{j''}{\mu \cdot \left(\nu_{n_0}\right)^{dl^{\alpha-2}}} = \left(\lambda^{dl^{\alpha-2}}\right)^{n_1},$$
where the only unknown is n_1. Thus, n_1 can be obtained by solving a DLP in $\mathbb{F}_{q^r}^*$.

4. If $\alpha = 2$, then we are done since $n_\alpha = n_0 + n_1 l$. Otherwise, proceed to compute n_2 such that
$$n_2 := \frac{n - n_0 - n_1 l}{l^2} \bmod l,$$
and then repeat this process for n_3, n_4, ..., $n_{\alpha-1}$. Finally get $n_\alpha = n_0 + n_1 l + n_2 l^2 + ... + n_{\alpha-1} l^{\alpha-1}$.

5. At last, combine n_d and n_α using the Chinese remainder theorem in order to get n.

The remarkable property of this method is that

The value of ν_{n_0} *is used to compute* n_1.

As a result, this *still* suggests that the value of n_0, obtained by solving a DLP in E, should be known prior to the computation of n_1. In other words, to compute n_α, the discrete logarithm on the elliptic curve should be performed *first*, and then be followed by discrete logarithms in $\mathbb{F}_{q^r}^*$. Reiterate that this sequence of operations is similar to the Pohlig-Hellman method. Now to the best of our knowledge, there is no version of this method that allows to retrieve n_1 without computing n_0 first. Thus the best method we know in this case involves to sequentially extract a discrete logarithm in E *followed* by at least one in $\mathbb{F}_{q^r}^*$.

Those familiar with pairing-based cryptography will have noticed that the problem of choosing a smooth elliptic curve E over $\mathbb{F}_q$ such that $\#E(\mathbb{F}_q) = l \cdot h$ (with h small) and $l \mid q^r - 1$ is in fact *identical* to the problem of generating suitable curves for pairing-based applications. As opposed to the usual ECC, generating curves at random until a suitable one is found no longer is an efficient method in this case [BK98]. Fortunately, the frenzy surrounding pairing-based crypto stimulated the search for efficient curve generation algorithms. We now know several techniques allowing to efficiently choose suitable curves for various values of r of cryptographic interest. See [MNT01, BS04, BW03, BN05, DEM05] for details.

The informal argument that "the inner workings of the Pohlig-Hellman method suggests that we *must* solve a DLP in E, followed by (at least) one in $\mathbb{F}_{q^r}^*$" is of course far from being a satisfactory answer. Indeed, there may be other techniques allowing to solve everything in parallel...

We thus now explore other avenues that could lead to a general parallel solution. We start by considering the computing sequence presented in Figure 5.10.

$\mathbb{F}_{q^r}^*$		E
$\nu_{n_0} \cdot \lambda^{n_1}$		$n_0 P$
↓		↓
$(\log_\lambda \nu_{n_0} + n_1) \bmod (q^r - 1)$		n_0
		↓
	ν_{n_0}	ν_{n_0}
	↓	
	$\log_\lambda \nu_{n_0}$	
	↓	
	n_1	

Figure 5.10: An alternate solution to the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$

At first sight, this method appears to be even worse than the solution of Figure 5.9 since it first performs a discrete logarithm in E and one in $\mathbb{F}_{q^r}^*$ in parallel, followed by a second discrete logarithm in $\mathbb{F}_{q^r}^*$. However, suppose that the values of $\log_\lambda \nu_{n_0}$ have been precomputed for all possible values of ν_{n_0}. That is, we possess a table $\mathbf{T}$ listing the possible ν_{n_0} along with their respective discrete logarithms to the base λ. Then as soon as the value of ν_{n_0} is known, a simple table look-up would yield $\log_\lambda \nu_{n_0}$, and thus n_1. Granted that these precomputations have been performed, we would then only need to solve a DLP in E and in $\mathbb{F}_{q^r}^*$ in parallel. The precomputation time will then determine if this strategy can be realistically considered.

During the precomputation phase, a discrete logarithm in $\mathbb{F}_{q^r}^*$ has to be computed for each of the possible value of ν_{n_0}. Thus if we let Γ be the set of all possible ν_{n_0}, that is

$$\Gamma = \{\nu_0, \nu_1, \nu_2, \ldots, \nu_{l-1}\} \subseteq \mathbb{F}_{q^r}^*,$$

then $1 \leq \#\Gamma \leq l$ represents the number of entries in $\mathbf{T}$. For instance, if Γ is of exponential size, then the precomputation time will be exponential as well, and this strategy would then be impractical. Thus the next step is to study the quantity $\#\Gamma$, but before we do so, we note a few observations conserning Γ.

Remark 5.17 *Even if $\mathbb{F}_{q^r}^* \times \langle B \rangle$ is cyclic, it is still possible to have $\nu_i = \nu_j$ with $i \neq j$. Indeed, in our toy example of Section 5.3.3, we had $\nu_3 = \nu_4 = 4$.*

Remark 5.18 *The set Γ, as well as its size, does not only depend on the choice of the group $\mathbb{F}_{q^r}^* \times \langle B \rangle$, but also on the specific generator chosen. In the example of Section 5.3.3, we considered the generalized Jacobian of $E : y^2 = x^3 + x + 4$ over $\mathbb{F}_7$ with respect to $\mathfrak{m} = ((0,5)) + ((5,1))$, and worked in the subgroup $\mathbb{F}_7^* \times \langle B \rangle$ of order 30 generated by $(k, B) = (1, (6,3))$. For that generator, we already possess all the information at hand to form the corresponding set $\Gamma = \{1, 2, 4\}$. Another possible generator for this subgroup is $(k', B') = 23(k, B) = (2, (4,4))$ since $\gcd(30, 23) = 1$ (see fact 5.26). Using the precomputed list of multiples of (k, B) provided in Table 5.1, it is a simple matter to get that*

$$\begin{aligned}
0(k', B') &= (1, \mathcal{O}), \\
1(k', B') &= 23(k, B) = (2, (4,4)), \\
2(k', B') &= 16(k, B) = (6, (6,3)), \\
3(k', B') &= 9(k, B) = (5, (6,4)), \\
4(k', B') &= 2(k, B) = (2, (4,3)).
\end{aligned}$$

Thus, $\Gamma' = \{1, 2, 5, 6\}$ is the set associated to (k', B'). Hence, we have that $\Gamma \subsetneq \Gamma'$, $\Gamma \supsetneq \Gamma'$ and $\#\Gamma \neq \#\Gamma'$, even if (k, B) and (k', B') generate the same group.

Remark 5.19 *Lastly, notice that Γ is simply a set, and is not necessarily a subgroup of $\mathbb{F}_{q^r}^*$ (and so $\#\Gamma$ does not have to divide $q^r - 1$). Indeed, we just saw that $\Gamma' = \{1, 2, 5, 6\}$ and thus Γ' cannot be a subgroup of $\mathbb{F}_7^*$ since $\#\Gamma' \nmid \#\mathbb{F}_7^*$.*

The Classical Occupancy Problem

In order to get a first idea on the size of Γ, a good place to start is to look at some empirical data. Using the computer algebra system MAGMA, six thousand generalized Jacobians with $r = 1$, q prime such that $2^{14} < q < 2^{20}$ and underlying elliptic curves with cofactor $h = 2$ were pseudo-randomly generated. For the time being, we leave the details of the implementation aside since we simply want to get an idea of the relative size of Γ at this point[15]. For each

[15] More details concerning the implementation can be found on page 165.

generalized Jacobian, the percentage $100 \cdot \#\Gamma/l$, representing the proportion of the number of distinct ν_{n_0} to the maximal size[16]

Note that we indeed have $l \leq q-1$ since by Hasse's Theorem (see Theorem 3.63), $l = \#E(\mathbb{F}_q)/h \leq \big(q+1+2\sqrt{q}\big)/2 \leq q - 1.00000\ 00000$ of Γ, was tabulated. The histogram of Figure 5.11 was obtained.

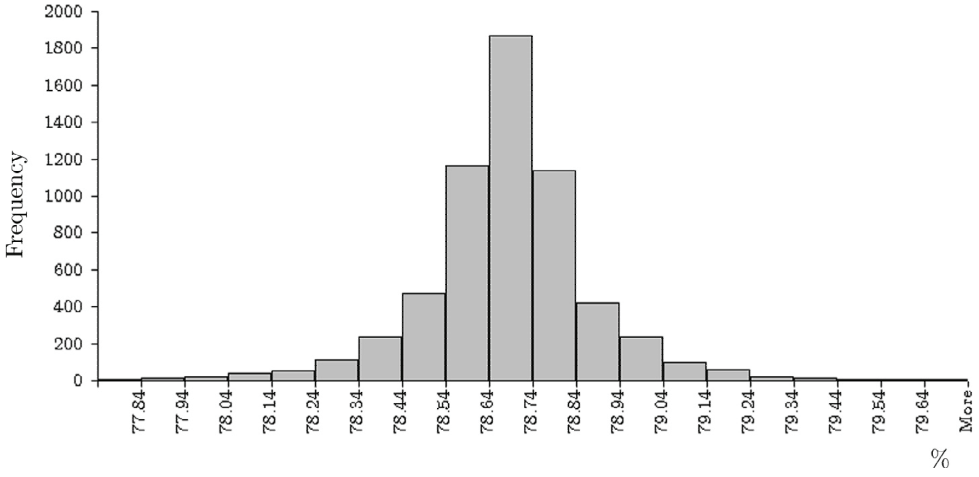


Figure 5.11: A first look at the relative size of Γ

The obvious observation is that the mean value on this graphic is about $78,7\%$. So roughly speaking, the size of Γ for a typical generalized Jacobian in this sample is at least $3l/4$: this would mean that the precomputation step requires the extraction of $3l/4$ discrete logarithms in $\mathbb{F}_{q^r}^*$, which is clearly out of reach. This first impression being really positive, we now need to take a closer look at the behavior of $100 \cdot \#\Gamma/l$.

Such a nice bell shape most probably means that more can be said about the expected value of the quantity $100 \cdot \#\Gamma/l$. In order to see how we could model this problem, we go back to the very definition of the ν_{n_0}'s. We know that for every $n_0 \in \{0, 1, \ldots, l-1\}$, the value of $\nu_{n_0} \in \mathbb{F}_{q^r}^*$ is obtained by evaluating $n_0(k, P) = (\nu_{n_0}, n_0 P)$. So each possible n_0 is *assigned to* one of $q^r - 1$ possible values. Thus, each n_0 could be represented by a tennis ball that is thrown into one of the possible $q^r - 1$ boxes. Once each of the l balls have been placed into their respective box, we simply need to count the number of *nonempty* boxes to get $\#\Gamma$.

Balls and urns are to probability theory what padlocks and safes are to cryptography. In addition of being great didactic tools, these physical models allow us to keep in mind intuitive properties while helping us to make the connection between seemingly unrelated problems. Thus

[16]

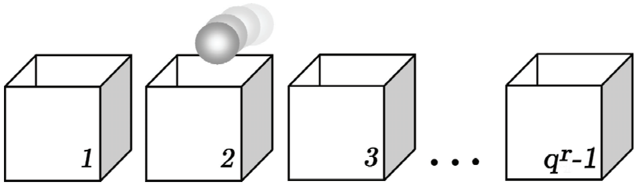


Figure 5.12: The Classical Occupancy Problem

reinterpreting our really specific problem in terms of balls and boxes might help us to have a more global perspective on the behavior of the generalized Jacobians we consider.

Perhaps the most natural experiment we can think of in terms of balls and boxes would be to *randomly* throw l balls into $q^r - 1$ boxes. That is, the l balls are randomly and independently distributed among the $q^r - 1$ equally probable boxes. At the end of this process, let Δ be the set of nonempty boxes that were obtained.

We have then produced, with the pseudo-random number generator of Magma, a sample of six thousand pairs of Γ and Δ in order to compare the behavior of genuine generalized Jacobians with random assignment of balls into boxes. To generate each pair, a pseudo-random generalized Jacobian (with $r = 1$, q prime such that $2^{14} < q < 2^{20}$ and underlying elliptic curve of cofactor $h = 2$) was first generated, and for the corresponding values of l and $q - 1$, the quantities $\#\Gamma$ and $\#\Delta$ were tabulated. The results are shown in the graphic below.

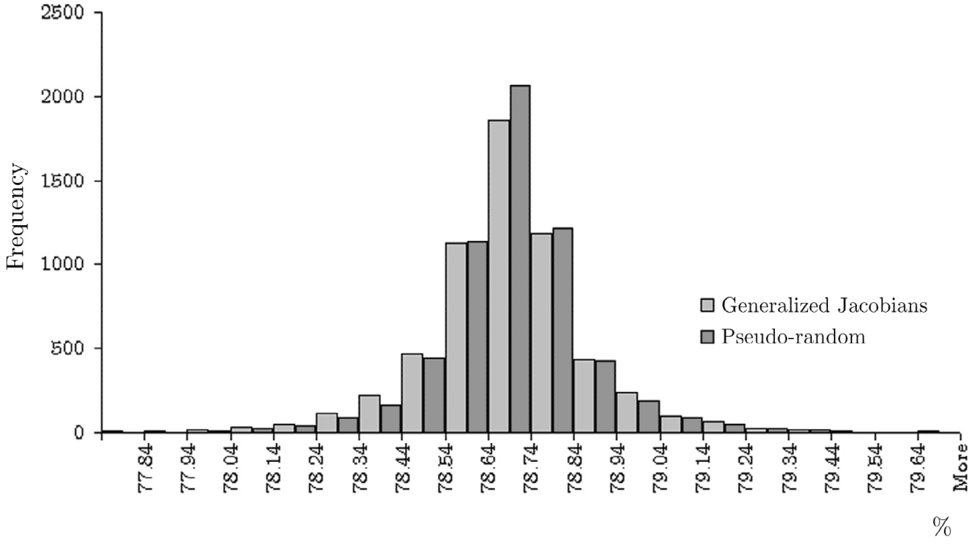


Figure 5.13: Comparing the relative size of Γ and Δ

Looking at this histogram, it appears that there is a strong correlation between these two sets of data. Indeed, the mean of both samples equals $78,69\%$, while the sample variance obtained is $0,038$ for data obtained from generalized Jacobians, and $0,029$ for pseudo-random numbers. It would therefore be useful to know more about the simplified experiment where each ball is *randomly* thrown.

Afterall, the idea of placing balls into boxes is quite natural, so it has certainly been considered before. We are thus looking for information on a discrete distribution for which our own description is in terms of tennis balls and shoe boxes[17]... In the first volume of *An Introduction to Probability Theory and its Applications* [Fel68], William Feller does provide the terminology we are looking for. Indeed, the *Classical Occupancy Problem* refers to the experiment where $\mathcal{B} > 0$ balls are distributed among $\mathcal{C} > 1$ cells such that each of the $\mathcal{C}^{\mathcal{B}}$ possible outcome has probability $(1/\mathcal{C})^{\mathcal{B}}$.

Interestingly enough, this urn model arises in a wide variety of applications, such as the theory of photographic emulsions, irradiation in biology, cosmic ray experiments and even gene distributions [Fel68, Section I.2]. As a consequence, various results concerning this problem are available in the literature.. Two other general references for the Classical Occupancy Problem are the following books by Johnson and Kotz [JK69, JK77].

For the application we have in mind, we are mainly concerned with the number $\mathbf{X}$ of *occupied* (i.e. nonempty) cells. First, the probability that *exactly* t cells ($1 \le t \le \min(\mathcal{B}, \mathcal{C})$) are taken is given by

$$\Pr(\mathbf{X} = t) = \frac{\binom{\mathcal{C}}{t}}{\mathcal{C}^{\mathcal{B}}} \sum_{i=1}^{t} (-1)^{t+i} \binom{t}{i} i^{\mathcal{B}},$$

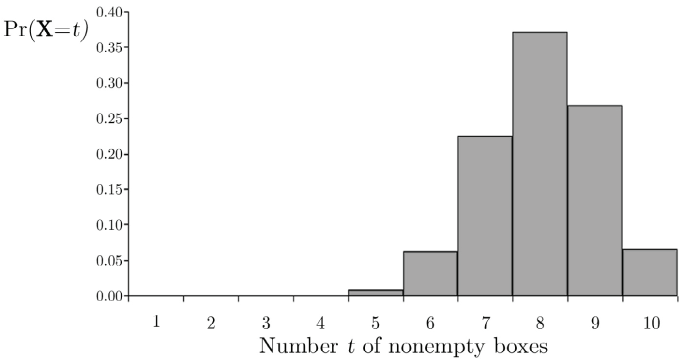
which can be obtained using Boole's formula [JK69, Sec. I.4], or simply by inspection.

Closed expressions for the expected value and variance can also be found in [JK69, Section 10.5]:

$$\begin{aligned} \mathrm{E}[\mathbf{X}] &= \mathcal{C}\left(1 - \left(1 - \frac{1}{\mathcal{C}}\right)^{\mathcal{B}}\right), \text{ and} \qquad (5.28) \\ \mathrm{Var}[\mathbf{X}] &= \mathcal{C}\left(1 - \frac{1}{\mathcal{C}}\right)^{\mathcal{B}} + \mathcal{C}(\mathcal{C}-1)\left(1 - \frac{2}{\mathcal{C}}\right)^{\mathcal{B}} - \mathcal{C}^2\left(1 - \frac{1}{\mathcal{C}}\right)^{2\mathcal{B}}. \end{aligned}$$

Example 5.20 *If we quickly look at a small example, say $\mathcal{B} = 10$ and $\mathcal{C} = 20$, the expected number of nonempty cells is $8,03$ while the variance equals $1,08$. visually, the probability function is as depicted in Figure 5.14.*

[17] There will always be situations where Google will be helpless, and that was one of them. Luckily, humans have a lot more imagination, so when asked the question *"You know the experiment of throwing balls into boxes...*



t	$\Pr(\mathbf{X}=t)$
1	$1,95 \times 10^{-12}$
2	$1,90 \times 10^{-8}$
3	$6,23 \times 10^{-6}$
4	$3,87 \times 10^{-4}$
5	$0,00773$
6	$0,0622$
7	$0,224$
8	$0,372$
9	$0,268$
10	$0,0655$

Figure 5.14: Probability function for the occupancy distribution with $\mathcal{B} = 10$ and $\mathcal{C} = 20$.

We are here interested in the relative size $\mathbf{Y} = 100 \cdot \mathbf{X}/\mathcal{B}$ of the set Δ. In particular, we would like to say more about the expected value and variance of $\mathbf{Y}$ subject to the constraints $\mathcal{B} = l = \#E(\mathbb{F}_q)/h$ and $\mathcal{C} = q^r - 1$ when q is relatively large[18].

To do so, we will treat the case $r = 1$ and $r > 1$ separately since they will turn out to have quite different behaviors. In fact, it is already possible to intuitively see why we should make this distinction. If $r > 1$, then the number of boxes is at least $q^2 - 1$ while the number of balls can never exceed $q+1+2\sqrt{q}$ by Hasse's theorem (c.f. Theorem 3.63). Since the number of boxes is rather large compared to the amount of balls, we thus suspect that the number of nonempty boxes should be really close to $\mathcal{B}$. On the other hand, if $r = 1$, then the number of balls is at least $\left(q + 1 - 2\sqrt{q}\right)/h$ while the number of boxes equals $q - 1$, and thus these two quantities are now of the same order of magnitude. In that case, the experimental data for $h = 2$ shown in Figure 5.13 suggest that for a given h, the expected quantity of nonempty boxes should be a certain fraction of $\mathcal{B}$ yet to be determined. We are now ready to turn these intuitive observations into factual statements.

We begin with the case $r = 1$, and hence we have $\mathcal{B} = l = \#E(\mathbb{F}_q)/h$ and $\mathcal{C} = q - 1$. Also recall that we consider cofactors $h \geq 2$ when $r = 1$ since all moduli would otherwise be B-related. So as previously pointed out, this implies that $\mathcal{B} \leq \mathcal{C}$. We first turn our attention to the expected value of $\mathbf{Y}$. Since $100/\mathcal{B}$ is a constant, it follows that $\mathrm{E}[\mathbf{Y}] = 100 \cdot \mathrm{E}[\mathbf{X}]/\mathcal{B}$. Using the lower and upper bound for $\mathcal{B}$ provided by Hasse's theorem,

$$\frac{q+1-2\sqrt{q}}{h} \leq \mathcal{B} \leq \frac{q+1+2\sqrt{q}}{h},$$

Does it have a name?", chances are someone will eventually remember where to look for the answer. I wish to thank Jose Correa of the McGill Statistical Consulting Service for suggesting to look at Feller's book [Fel68].

[18] That is, q has a minimum of 160 bits (and so is at least 1.5×10^{48}).

we therefore get that

$$\frac{100h(q-1)}{q+1+2\sqrt{q}}\left(1-\left(1-\frac{1}{q-1}\right)^{\frac{q+1-2\sqrt{q}}{h}}\right) \leq \mathrm{E}[\mathbf{Y}] \leq \frac{100h(q-1)}{q+1-2\sqrt{q}}\left(1-\left(1-\frac{1}{q-1}\right)^{\frac{q+1+2\sqrt{q}}{h}}\right).$$

Evaluating the limit of this lower and upper bound as q tends to infinity, we obtain that they both converge to the same quantity:

$$\lim_{q\to\infty}\frac{100h(q-1)}{q+1\pm 2\sqrt{q}}\left(1-\left(1-\frac{1}{q-1}\right)^{\frac{q+1\mp 2\sqrt{q}}{h}}\right) = 100h\left(1-\frac{1}{\sqrt[h]{e}}\right).$$

Thus by the squeeze (or sandwich) theorem [BS82, Thm 3.2.7] of elementary real analysis, it follows that

$$\lim_{q\to\infty}\mathrm{E}[\mathbf{Y}] = 100h\left(1-\frac{1}{\sqrt[h]{e}}\right).$$

The convergence of $\mathrm{E}[\mathbf{Y}]$ is illustrated in Figure 5.15 for $r=1$, $h=2$ and three particular (real) values of l.

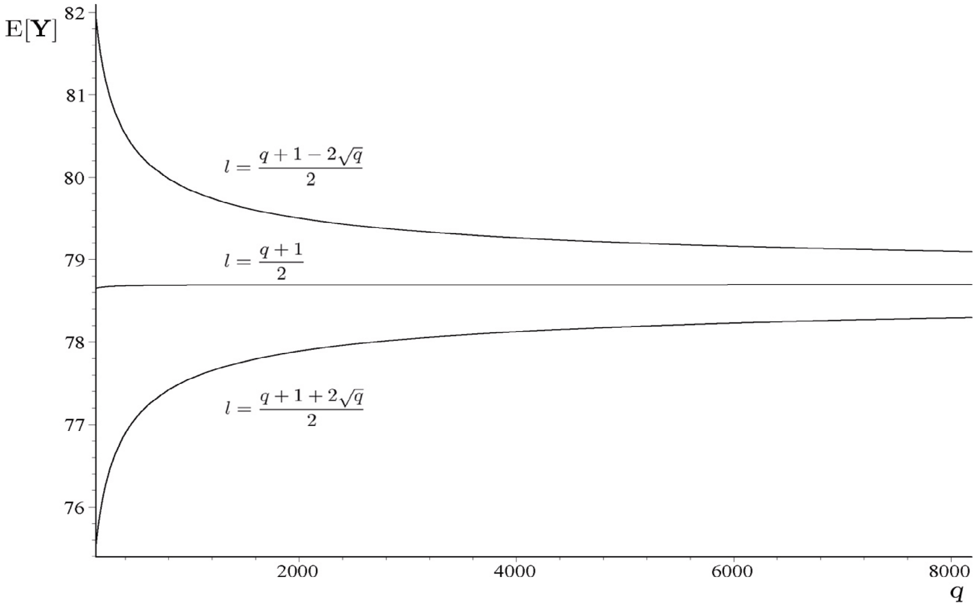


Figure 5.15: On the convergence of $\mathrm{E}[\mathbf{Y}]$ for three particular values of l

We therefore have that

$$\lim_{q\to\infty}\mathrm{E}[\mathbf{Y}] = \begin{cases} 78,6939\% & \text{if } h=2,\\ 85,0406\% & \text{if } h=3,\\ 88,4797\% & \text{if } h=4.\end{cases}$$

That is, for q large enough, we should expect for $h = 2$, 3 and 4 to have respectively $0,79 \cdot l$, $0,85 \cdot l$ and $0,88 \cdot l$ elements in Δ. The next step is then to compare these results with samples obtained from true generalized Jacobians.

But before we do so, and for the sake of completeness, we now describe the procedure used to generate all the samples of this section. Given positive integers r, h, LB and UB, each generalized Jacobian was pseudo-randomly chosen as follows. First, a random[19] prime p such that $2^{LB} < p < 2^{UB}$ is first fixed. Then, random a, $b \in \mathbb{F}_p$ are generated until the curve $E : y^2 = x^3 + ax + b$ is nonsingular and $\#E(\mathbb{F}_p)/h$ is a prime integer[20]. A random point $B \in E(\mathbb{F}_p)$ is then selected until it has order $\#E(\mathbb{F}_p)/h$, and the basepoint is set to (k, B) for a randomly chosen $k \in \mathbb{F}_{p^r}^*$. Notice that we did not require that the basepoint generates all of $\mathbb{F}_{p^r}^* \times \langle B \rangle$ in order to remain as general as possible. Next, $M \in E(\mathbb{F}_{p^r})$ is randomly chosen until $M \notin \langle B \rangle$, which is followed by a random choice of $N \in E(\mathbb{F}_{p^r})$ that fulfills $N \neq M$ and $N \notin \langle B \rangle$. Finally, we set $\mathfrak{m} = (M) + (N)$. Then starts the determination of Γ: from $\nu_0 = 1$, the elements $\nu_1, \nu_2, \ldots, \nu_{l-1}$ are recursively computed using the relation $\nu_{i+1} = k \cdot \nu_i \cdot \mathbf{c}_{\mathfrak{m}}(B, iB)$ $(0 \leq i < l - 1)$, which easily follows from the group law algorithm given in Proposition 5.4 :

$$\begin{aligned} (\nu_{i+1}, (i+1)B) &= (i+1)\,(k, B) = (k, B) + i\,(k, B) \\ &= (k, B) + (\nu_i, iB) = (k \cdot \nu_i \cdot \mathbf{c}_{\mathfrak{m}}(B, iB), (i+1)B)\,. \end{aligned}$$

In parallel, each time a ν_i is computed, a random element of $\mathbb{F}_{p^r}^*$ is also generated and included in the separate set Δ, which therefore allows to compare this generalized Jacobian with the Classical Occupancy Problem where $\mathcal{B} = l$ and $\mathcal{C} = p^r - 1$.

The results of our simulations for $r = 1$ are shown in Table 5.2, where each entry corresponds to the mean of a sample of size two thousand.

		$2^{14} < q < 2^{16}$	$2^{16} < q < 2^{18}$	$2^{18} < q < 2^{20}$
$h = 2$	Generalized Jacobians	$78,6859$	$78,6939$	$78,6920$
	Pseudo-Random	$78,6964$	$78,6936$	$78,6936$
$h = 3$	Generalized Jacobians	$85,0352$	$85,0405$	$85,0391$
	Pseudo-Random	$85,0346$	$85,0462$	$85,0451$
$h = 4$	Generalized Jacobians	$88,4724$	$88,4779$	$88,4818$
	Pseudo-Random	$88,4629$	$88,4811$	$88,4795$

Table 5.2: Sample means of the relative size of Γ and Δ for $r = 1$ and sample size two thousand

[19] Whenever we refer to a '*random choice*' of a parameter, it is understood that we used the buit-in MAGMA function that returns a pseudo-random element from the chosen set. For instance, `Random(E)` returns a pseudo-random point on the elliptic curve `E`.

[20] Notice that we did not use the complex multiplication method (CM method) to generate the curves.

In the light of these empirical results, we now have a better idea of the accuracy of the theoretical predictions inspired by the Classical Occupancy Problem. As the ideal companion to Table 5.2 is Figure 5.16, which displays three histograms, one for each cofactor, obtained for generalized Jacobians with $2^{18} < q < 2^{20}$.

A quick glance at the graphics of Figure 5.16 reveals that the majority of the results fall within $\pm 0,5\%$ of the mean, and thus that the variance is relatively small for these samples. So we now return to the Classical Occupancy Problem in order to study the behavior of Var[$\mathbf{Y}$]. Recall that $\mathbf{Y}$ =$100 \cdot \mathbf{X}/\mathcal{B} = 100 \cdot \mathbf{X}/l$ and that a closed expression for the variance of $\mathbf{X}$ was given by equation (5.28). Thus, $\mathrm{Var}[\mathbf{Y}] = (100/l)^2\, \mathrm{Var}[\mathbf{X}]$ and so

$$\begin{aligned}\mathrm{Var}[\mathbf{Y}] \quad = \quad & \frac{(100h)^2(q-1)}{\left(\#E(\mathbb{F}_q)\right)^2}\left(\left(1-\frac{1}{q-1}\right)^{\frac{\#E(\mathbb{F}_q)}{h}} + (q-2)\left(1-\frac{2}{q-1}\right)^{\frac{\#E(\mathbb{F}_q)}{h}}\right.\\ & \left. -(q-1)\left(1-\frac{1}{q-1}\right)^{\frac{2\#E(\mathbb{F}_q)}{h}}\right).\end{aligned}$$

By Hasse's theorem (Theorem 3.63), we know that Var[$\mathbf{Y}$] is bounded below by

$$\begin{aligned}(100h)^2 & \left(\frac{(q-1)}{\left(q+1+2\sqrt{q}\right)^2}\left(1-\frac{1}{q-1}\right)^{\frac{q+1+2\sqrt{q}}{h}} + \frac{(q-1)(q-2)}{\left(q+1+2\sqrt{q}\right)^2}\left(1-\frac{2}{q-1}\right)^{\frac{q+1+2\sqrt{q}}{h}}\right.\\ & \left. -\frac{(q-1)^2}{\left(q+1-2\sqrt{q}\right)^2}\left(1-\frac{1}{q-1}\right)^{\frac{2\left(q+1-2\sqrt{q}\right)}{h}}\right),\end{aligned}$$

which converges to zero as q tends to infinity. Similarly, an upper bound for Var[$\mathbf{Y}$] is given by

$$\begin{aligned}(100h)^2 & \left(\frac{(q-1)}{\left(q+1-2\sqrt{q}\right)^2}\left(1-\frac{1}{q-1}\right)^{\frac{q+1-2\sqrt{q}}{h}} + \frac{(q-1)(q-2)}{\left(q+1-2\sqrt{q}\right)^2}\left(1-\frac{2}{q-1}\right)^{\frac{q+1-2\sqrt{q}}{h}}\right.\\ & \left. -\frac{(q-1)^2}{\left(q+1+2\sqrt{q}\right)^2}\left(1-\frac{1}{q-1}\right)^{\frac{2\left(q+1+2\sqrt{q}\right)}{h}}\right),\end{aligned}$$

which also tends to zero when q goes to infinity. It thus follows that

$$\lim_{q\to\infty} \mathrm{Var}[\mathbf{Y}] = 0.$$

Actually, it is possible to graphically see how the variance decreases as q augments for genuine generalized Jacobians. Figure 5.17 shows the results of our simulations for $r = 1$, $h = 2$, and three intervals for q, where each sample is of size two thousand.

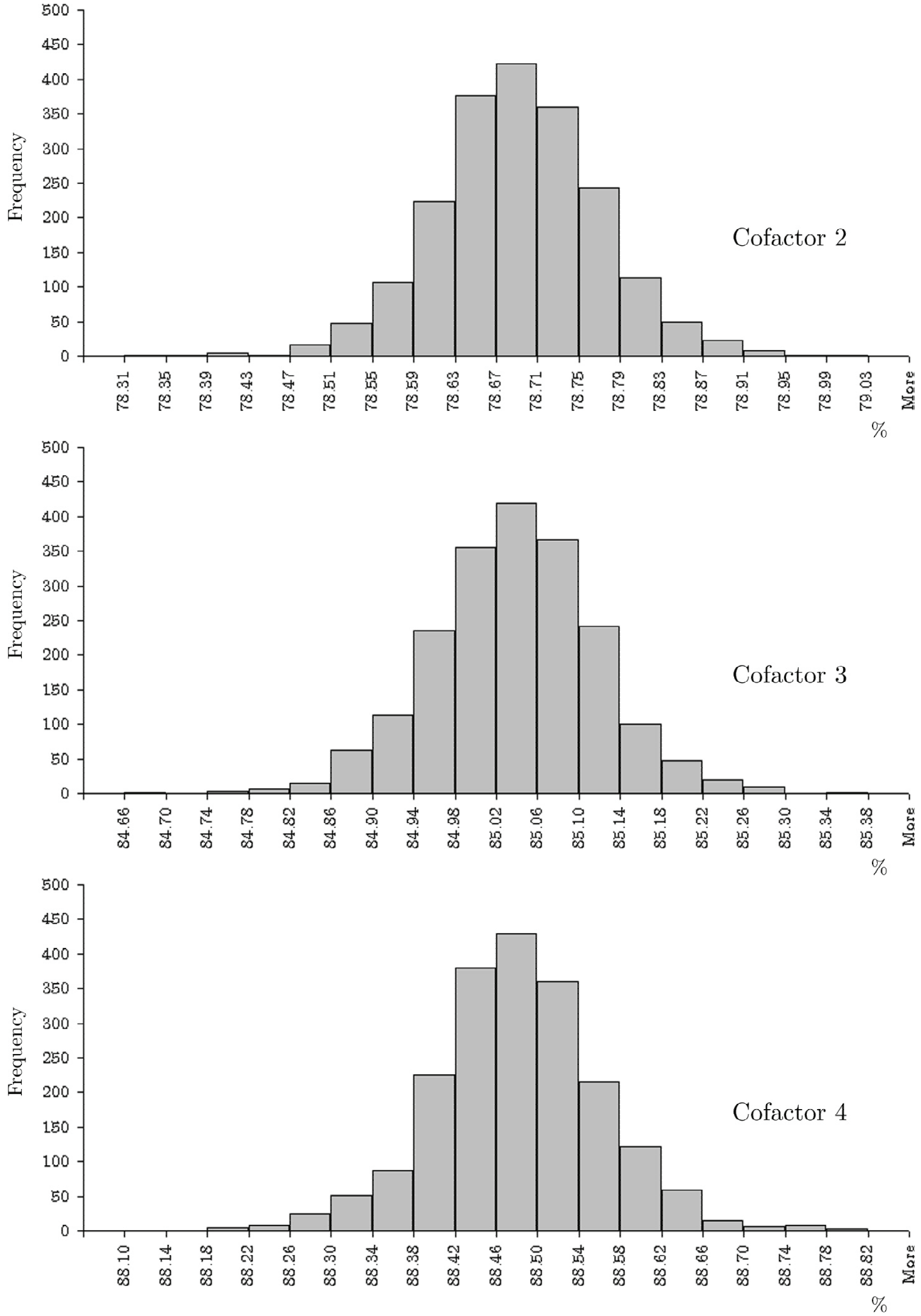


Figure 5.16: Relative size of Γ for $r = 1$, $2^{18} < q < 2^{20}$ and sample size two thousand

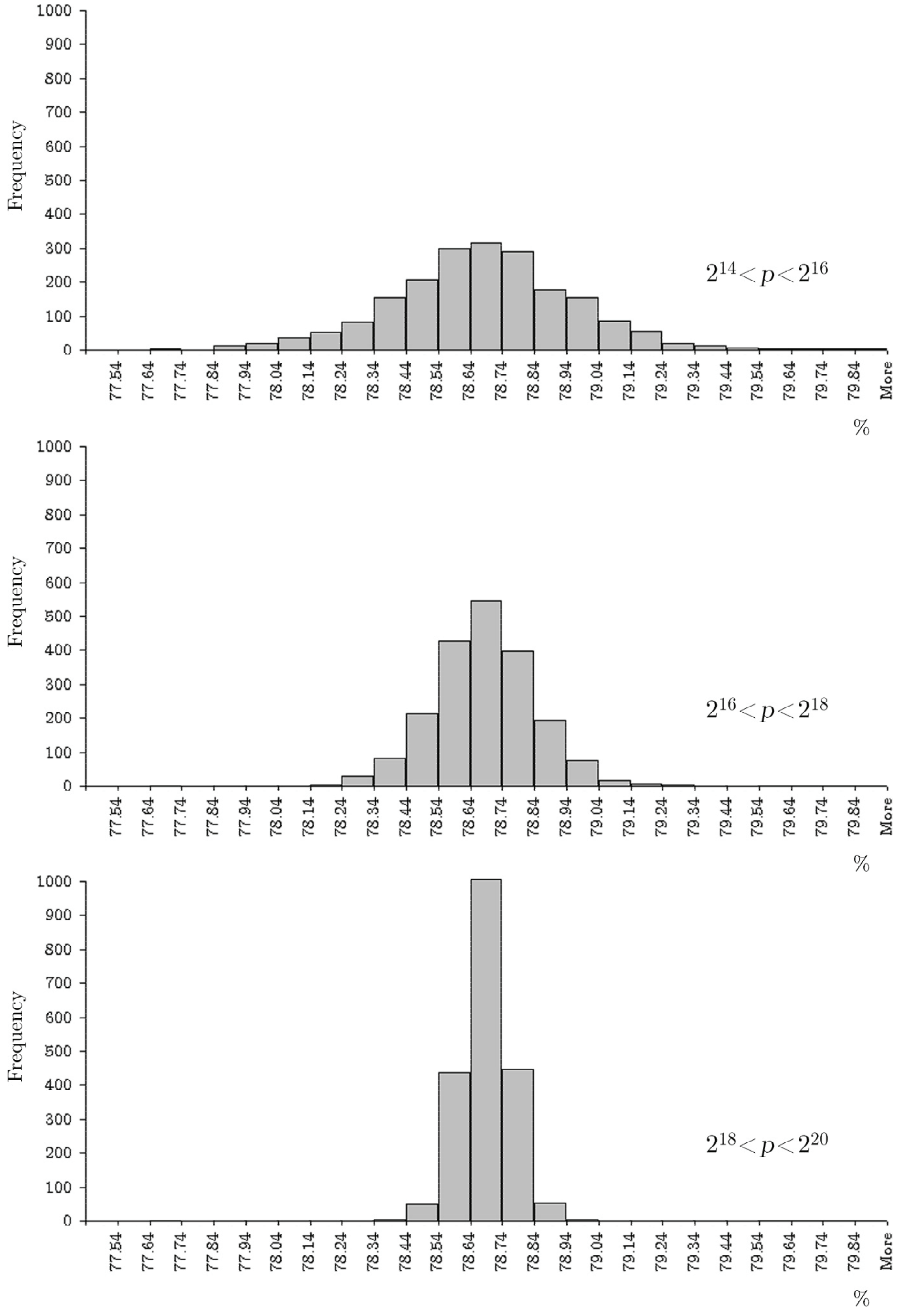


Figure 5.17: Relative size of Γ for $r = 1$, $h = 2$ and sample size two thousand

We of course also include Table 5.3 showing the variances[21] obtained for each sample, both for generalized Jacobians and for the pseudo-random case that simulates the Classical Occupancy Problem.

		$2^{14} < q < 2^{16}$	$2^{16} < q < 2^{18}$	$2^{18} < q < 2^{20}$
$h = 2$	Generalized Jacobians	0,08480	0,02452	0,00629
	Pseudo-Random	0,06379	0,01788	0,00437
$h = 3$	Generalized Jacobians	0,09912	0,02389	0,00635
	Pseudo-Random	0,08686	0,02089	0,00546
$h = 4$	Generalized Jacobians	0,10464	0,02464	0,00669
	Pseudo-Random	0,10176	0,02471	0,00594

Table 5.3: Sample variances of the relative size of Γ and Δ for $r = 1$ and sample size two thousand

At this point, we know that our experimental results for generalized Jacobians agree fairly well with the Classical Occupancy Problem, both for the mean and variance. However, we only have a really vague idea of *'how fast'* we should expect the variance to decrease towards zero. For illustrative purposes, the graph of $\mathrm{Var}[\mathbf{Y}]$, as a function of q, for the case $r = 1$, $h = 2$ and $\#E(\mathbb{F}_q) = q + 1 - 2\sqrt{q}$, $q + 1$ and $q + 1 + 2\sqrt{q}$, is presented in Figure 5.18.

In order to study the rate of convergence of $\mathrm{Var}[\mathbf{Y}]$, we will next consider the quantity $q \cdot \mathrm{Var}[\mathbf{Y}]$. Using standard methods of real analysis, one can show that $\lim\limits_{q\to\infty} q \cdot \mathrm{Var}[\mathbf{Y}]$ exists and is equal to a constant. More precisely, we have that

$$L_h = \lim_{q\to\infty} q \cdot \mathrm{Var}[\mathbf{Y}] = \frac{10^4 h \left(he^{1/h} - h - 1\right)}{e^{2/h}} = \begin{cases} 2188,46 & \text{if } h = 2, \\ 2877,76 & \text{if } h = 3, \\ 3301,99 & \text{if } h = 4. \end{cases}$$

Indeed, when evaluating this limit, the only part that is a little trickier is

$$\lim_{q\to\infty} q \left(\left(1 - \frac{2}{q-1}\right)^{\#E(\mathbb{F}_q)/h} - \left(1 - \frac{1}{q-1}\right)^{2\#E(\mathbb{F}_q)/h} \right) = \frac{-1}{he^{2/h}}.$$

So when q is of cryptographic size, we expect that for $h = 2$, 3 or 4,

$$\mathrm{Var}[\mathbf{Y}] \approx \frac{L_h}{q} < \frac{2^{12}}{2^{159}} = 2^{-147}.$$

These results can be interpreted as follows. By Chebyshev's inequality [Fel68, Section IX.6], we know that for any real number $\epsilon > 0$,

$$\Pr\left(|\mathbf{Y} - \mathrm{E}\left[\mathbf{Y}\right]| \geq \epsilon\right) \leq \frac{\mathrm{Var}[\mathbf{Y}]}{\epsilon^2}.$$

[21] Recall that the *sample variance* of a finite set $\{x_1, \ldots, x_n\}$ of real numbers is given by $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$, where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

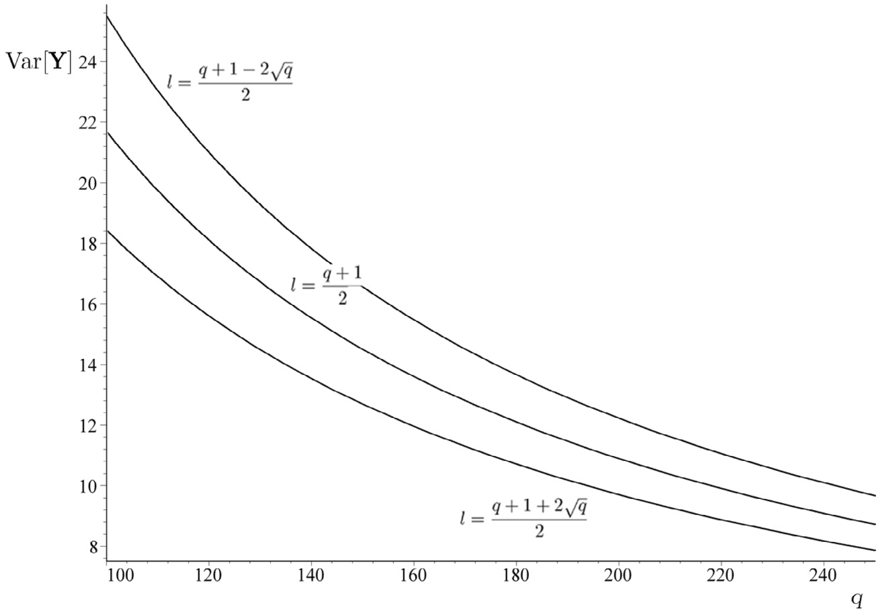


Figure 5.18: On the convergence of Var[**Y**] for three particular values of l

So for instance, if we allow a deviation of only 1% from the mean, we should have that

$$\Pr\left(|\mathbf{Y} - \mathrm{E}\left[\mathbf{Y}\right]| \geq 1\right) \leq \mathrm{Var}[\mathbf{Y}] \approx 2^{-147}$$

when q has (at least) 160 bits. Therefore, even if the upper bound given by Chebyshev's inequality might not be really tight[22], it is still extremely unlikely that Δ contains less than, say, $3l/4$ elements.

Next we consider the case $r > 1$, for which we have $\mathcal{B} = l = \#E(\mathbb{F}_q)/h$ balls and $\mathcal{C} = q^r - 1$ cells. We thus have the same number of balls as before, except that the number of boxes has now increased, which has made us previously remark that the expected number of nonempty boxes should be even larger in this case. Indeed, from equation (5.28), we get that

$$\mathrm{E}[\mathbf{Y}] = \frac{100h\left(q^r - 1\right)}{\#E(\mathbb{F}_q)}\left(1 - \left(1 - \frac{1}{q^r - 1}\right)^{\frac{\#E(\mathbb{F}_q)}{h}}\right),$$

and thus that $\mathrm{E}[\mathbf{Y}]$ is bounded below by

$$100h \cdot \frac{q}{q+1+2\sqrt{q}} \cdot \left(1 - \frac{1}{q^r}\right) \cdot q^{r-1}\left(1 - \left(1 - \frac{1}{q^r - 1}\right)^{\frac{q+1-2\sqrt{q}}{h}}\right), \tag{5.29}$$

[22] As it applies to *any* random variable for which the mean and variance exist and are finite.

and above by

$$100h \cdot \frac{q}{q+1-2\sqrt{q}} \cdot \left(1-\frac{1}{q^r}\right) \cdot q^{r-1}\left(1-\left(1-\frac{1}{q^r-1}\right)^{\frac{q+1+2\sqrt{q}}{h}}\right). \tag{5.30}$$

When evaluating the limits of (5.29) and (5.30) as $q \to \infty$, the only term that requires a little work is to show that

$$\lim_{q\to\infty} q^{r-1}\left(1-\left(1-\frac{1}{q^r-1}\right)^{\frac{q+1\pm 2\sqrt{q}}{h}}\right) = \frac{1}{h},$$

which can be done using a binomial expansion, for instance. As a result, both the lower and upper bound of $\mathrm{E}[\mathbf{Y}]$ converge to 100, and we therefore conclude that

$$\lim_{q\to\infty} \mathrm{E}[\mathbf{Y}] = 100. \tag{5.31}$$

So as soon as r is at least two and q is large, the relative size of Δ should be near 100%. In other words, the number of nonempty boxes should be close to $\mathcal{B}$, which agrees with our intuitive deductions previously made.

We could not hope for a better result about the Classical Occupancy Problem. Now is the time to put generalized Jacobians to the test: twelve samples, each containing two thousand generalized Jacobians with $r = 2$, have been generated[23] to cover cofactors up to four and three ranges for the prime q. Before we take a close look at the experimental means we obtained, a quick glance at Figure 5.19 reveals that our histograms for generalized Jacobians with $r = 2$ no longer have a symmetric *'bell shape'* (as in the case $r = 1$). Instead, all the weight is now concentrated to the right, near the maximum value of 100% that can be observed. Notice that this behavior agrees with equation (5.31) obtained for the Classical Occupancy Problem. Now, the details of the comparative results obtained for generalized Jacobians and for simulations of the Classical Occupancy Problem are shown in Table 5.4, where each entry is the mean of a sample of size two thousand with $r = 2$.

These amazingly good results imply, among other things, that *all* data observed in these samples must be strictly greater that 97%. Indeed, if even a single observation equals 97%, then the sample mean would be at most

$$\frac{97\% + 1999 \cdot 100\%}{2000} = 99,9985\%.$$

This thus indicates that we should expect the variance to be small in this case as well. In fact, if we once more go back to the Classical Occupancy Problem, we get the following surprisingly

[23] Using the same method as for $r = 1$ (see p.165).

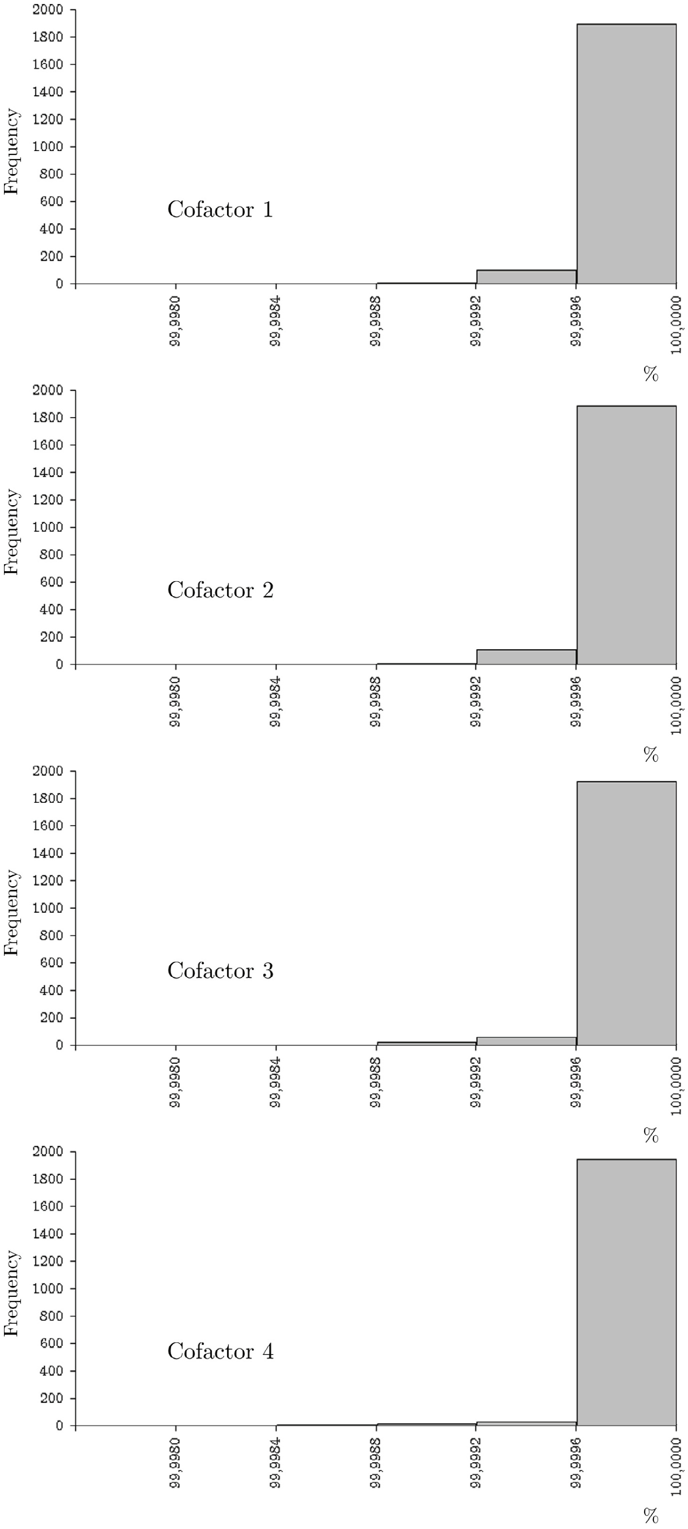


Figure 5.19: Relative size of Γ for $r = 2$, $2^{18} < q < 2^{20}$ and sample size two thousand

		$2^{14} < q < 2^{16}$	$2^{16} < q < 2^{18}$	$2^{18} < q < 2^{20}$
$h = 1$	Generalized Jacobians	$99,9986$	$99,9996$	$99,9999$
	Pseudo-Random	$99,9986$	$99,9996$	$99,9999$
$h = 2$	Generalized Jacobians	$99,9994$	$99,9998$	$100,0000$
	Pseudo-Random	$99,9993$	$99,9998$	$100,0000$
$h = 3$	Generalized Jacobians	$99,9994$	$99,9999$	$100,0000$
	Pseudo-Random	$99,9995$	$99,9999$	$100,0000$
$h = 4$	Generalized Jacobians	$99,9997$	$99,9999$	$100,0000$
	Pseudo-Random	$99,9996$	$99,9999$	$100,0000$

Table 5.4: Sample means of the relative size of Γ and Δ for $r = 2$ and sample size two thousand

simple result:

$$\lim_{q\to\infty} q^k \cdot \mathrm{Var}[\mathbf{Y}] = \begin{cases} 0 & \text{if } k < r, \\ 5000 & \text{if } k = r, \end{cases}$$

where $k \geq 0$ is an integer. Once more, this result can be shown using standard arguments of real analysis (mostly involving binomial expansions, geometric series and repeated applications of the squeeze theorem). So when q is large, we expect that

$$\mathrm{Var}[\mathbf{Y}] \approx \frac{5000}{q^r}.$$

Our experimental results once more concur with this theoretical prediction. Table 5.5 summarizes the sample variances obtained for various cofactors and ranges for q.

		$2^{14} < q < 2^{16}$	$2^{16} < q < 2^{18}$	$2^{18} < q < 2^{20}$
$h = 1$	Generalized Jacobians	$5,1396 \times 10^{-6}$	$3,2279 \times 10^{-7}$	$2,1332 \times 10^{-8}$
	Pseudo-Random	$4,7238 \times 10^{-6}$	$3,4185 \times 10^{-7}$	$2,1949 \times 10^{-8}$
$h = 2$	Generalized Jacobians	$3,7712 \times 10^{-6}$	$2,4373 \times 10^{-7}$	$1,9473 \times 10^{-8}$
	Pseudo-Random	$4,3389 \times 10^{-6}$	$3,0330 \times 10^{-7}$	$1,6301 \times 10^{-8}$
$h = 3$	Generalized Jacobians	$5,9096 \times 10^{-6}$	$2,5674 \times 10^{-7}$	$1,8891 \times 10^{-8}$
	Pseudo-Random	$4,4577 \times 10^{-6}$	$3,3428 \times 10^{-7}$	$1,8364 \times 10^{-8}$
$h = 4$	Generalized Jacobians	$3,9478 \times 10^{-6}$	$3,1000 \times 10^{-7}$	$2,0218 \times 10^{-8}$
	Pseudo-Random	$5,8625 \times 10^{-6}$	$2,5969 \times 10^{-7}$	$1,9297 \times 10^{-8}$

Table 5.5: Sample variances of the relative size of Γ and Δ for $r = 2$ and sample size two thousand

It is also possible to visualize how the variance diminishes as q increases with the help of the frequency histograms of these samples. The results of the simulations for generalized Jacobians associated with $r = 2$ and $h = 2$ are shown in Figure 5.20.

To wrap-up this (rather long) section, it might not be a bad idea to recapitulate and put in perspective the various results and observations we made. First recall that our main objective

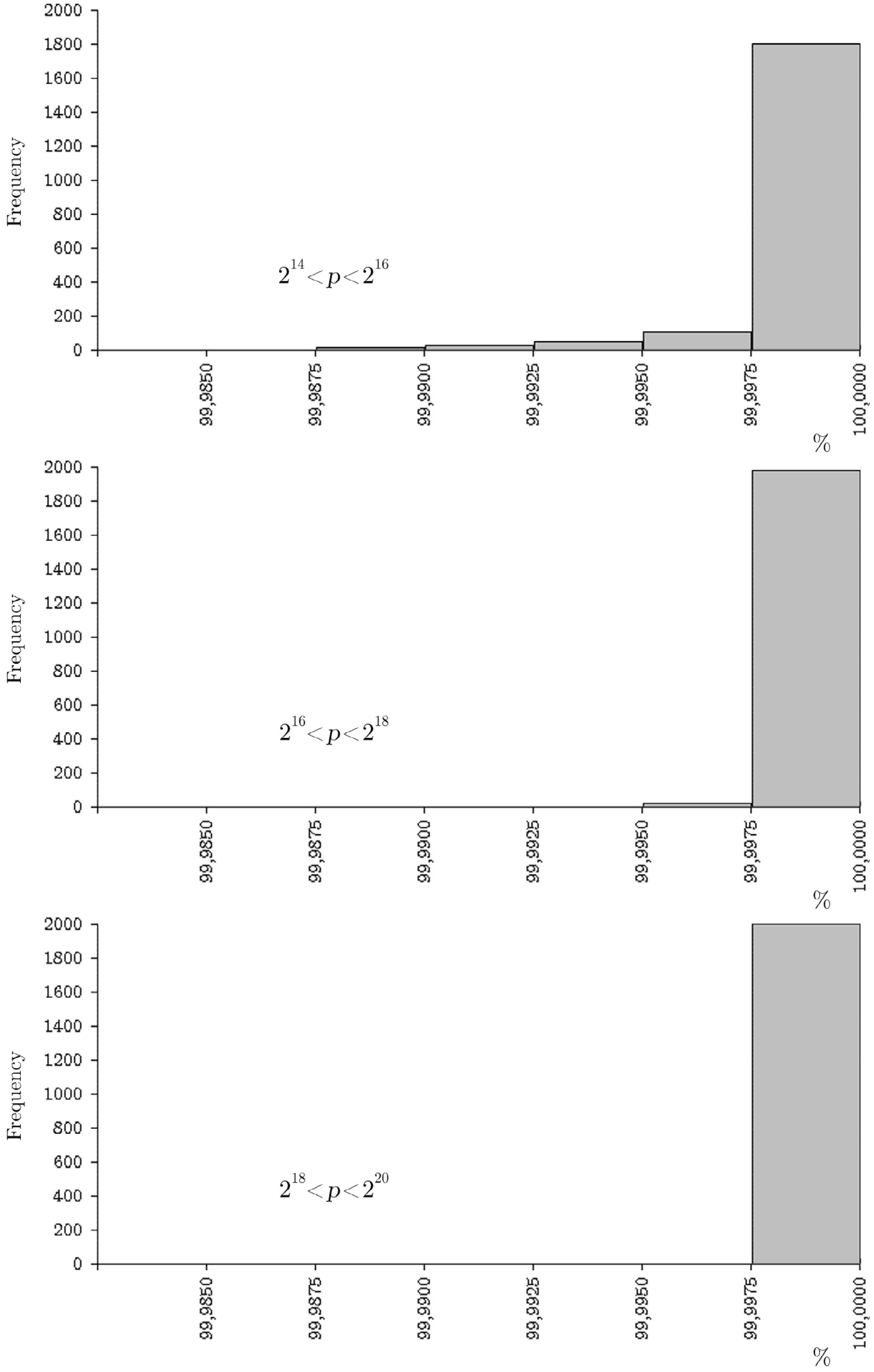


Figure 5.20: Relative size of Γ for $r = 2$, $h = 2$ and sample size two thousand

was to investigate whether it seemed possible to practically precompute a table $\mathbf{T}$ of the possible ν_{n_0} and corresponding $\log_\lambda \nu_{n_0}$. If so, we saw how the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ could then be solved by performing, in parallel, a discrete logarithm in $\mathbb{F}_{q^r}^*$ and one in E, followed by a table look-up in $\mathbf{T}$. That thus led us to study the size of the set $\Gamma = \{\nu_0, \nu_1, \nu_2, \ldots, \nu_{l-1}\} \subseteq \mathbb{F}_{q^r}^*$. Of course, the larger $\#\Gamma$ is, the longer it takes to compute $\mathbf{T}$.

A natural way of determining $\#\Gamma$ is to successively compute and store all ν_i $(0 \leq i < l)$ using the recurrence relation

$$\begin{cases} \nu_0 = 1, \\ \nu_{i+1} = k \cdot \nu_i \cdot \mathbf{c}_{\mathfrak{m}}(B, iB). \end{cases}$$

Thus using this method, l elements of $\mathbb{F}_{q^r}^*$ need to be computed and stored. Clearly, this can't be done for parameters that we would use in practice for cryptographic applications, as $l = \#E(\mathbb{F}_q)/h$ would then have roughly 160 bits. Nevertheless, it is always possible to compute $\#\Gamma$ for smaller values of l, and this is what we did with the help of MAGMA for primes q having between 15 and 20 bits. Altogether, a total of 42 000 generalized Jacobians have been pseudo-randomly generated and for each of them, the quantity $100 \cdot \#\Gamma/l$ was tabulated. Among these results, the minimum value that has been found was $77,4309\%$. Consequently, none of these generalized Jacobians had an associated set Γ of cardinality less than $3l/4$. In order to find heuristic arguments that would explain this behavior, we turned our attention to the Classical Occupancy Problem. This urn model was indeed simple enough to analyse its behavior as $q \to \infty$. In all cases, we obtained that $\lim\limits_{q\to\infty} \mathrm{E}[\mathbf{Y}]$ was defined and greater than 78%, while $\lim\limits_{q\to\infty} q^k \cdot \mathrm{Var}[\mathbf{Y}] = 0$ when $0 \leq k < r$ and $\lim\limits_{q\to\infty} q^r \cdot \mathrm{Var}[\mathbf{Y}]$ is a constant which is at most 5000. So from Chebyshev's inequality, it follows that the probability that $\#\Delta \leq 3l/4$ is less than 2^{-147} when q has at least 160 bits (this estimate holding for $r = 1$ and $h = 2$, 3, or 4 as well as for $r > 1$ and $h = 1$, 2, 3 or 4). This is excellent news concerning the Classical Occupancy Problem and exactly the kind of results we were hoping for.

The last step was then to see whether the simplified model provided by the Classical Occupancy Problem seemed to give a satisfactory approximation of the behavior of generalized Jacobians. Based on the results obtained from our simulations, the strong correlation between the two problems was manifest. Although far from being a proof, these qualitative observations provide a heuristic argument suggesting that in practice, it should be extremely unlikely that the number $\#\Gamma$ of entries in $\mathbf{T}$ will be less than $3l/4$, and consequently, that the time required to compute $\mathbf{T}$ allows to proceed in parallel as in Figure 5.10.

On the Amount of Balls Falling into each Box

In the previous section, we have studied the possibility that the set $\Gamma = \{\nu_0, \nu_1, \nu_2, \ldots, \nu_{l-1}\} \subseteq \mathbb{F}_{q^r}^*$ be small enough to build a table $\mathbf{T}$ containing the elements of Γ together with their discrete logarithms to the base λ. Even if we now have heuristics suggesting that this attack is unlikely to work, it may still be possible that an opponent gains some advantage in considering a particular (small) *subset* of Γ instead[24]. Indeed, there is a possibility that there exist up to $l - \#\Gamma + 1$ elements among ν_0, ν_1, ν_2, ..., ν_{l-1} that are all equal (to, say, $\nu \in \mathbb{F}_{q^r}^*$). In other words, each nonempty box would contain exactly one ball, *except one* that would be filled with the remaining $l - \#\Gamma + 1$ balls. From the previous section, we know[25] that $l - \#\Gamma + 1$ could be as large as $0.2131l$. In this eventuality, precomputing $\log_\lambda \nu$ would clearly be a good strategy since there would be over one chance out of five that solving a given instance[26] of the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ (as depicted in Figure 5.10) involves a table look-up of the value $\log_\lambda \nu$. Besides this worst-case scenario, it might more generally happen that *several* elements of Γ each have a non-negligible probability to arise while solving a DLP using the method outlined in Figure 5.10. If so, an adversary may then choose to precompute the discrete logarithms of some or all of these values.

So given an integer L such that $0 < L < l$, we now wish to know the likeliness that there is at least one box containing more than L balls. Our starting point will once more be the study of the Classical Occupancy Problem, where $\mathcal{B}$ balls are randomly distributed among $\mathcal{C}$ cells such that the probability that a given ball falls into any one of the boxes is always $1/\mathcal{C}$. Recall that the values of $\mathcal{B}$ and $\mathcal{C}$ we are here interested in satisfy $\mathcal{B} \leq \mathcal{C}$ (this property will be crucial in the argument that follows).

Now let $\mathbf{Z}_i$ be the random variable that represents the number of balls in box i after the $\mathcal{B}$ balls have been thrown ($1 \leq i \leq \mathcal{C}$). For each of the $\mathcal{B}$ independent throw, box i has probability $1/\mathcal{C}$ of receiving that ball, and thus, $\mathbf{Z}_i$ follows a binomial distribution with $\mathcal{B}$ Bernoulli trials and probability of success (i.e. the current ball goes into box i) $1/\mathcal{C}$. Hence, the probability that exactly j balls fall into box i is

$$\Pr\left(\mathbf{Z}_i = j\right) = \binom{\mathcal{B}}{j} \frac{1}{\mathcal{C}^j} \left(1 - \frac{1}{\mathcal{C}}\right)^{\mathcal{B}-j} \quad (0 \leq j \leq \mathcal{B}),$$

while the mean and variance are given by

$$\mathrm{E}[\mathbf{Z}_i] = \frac{\mathcal{B}}{\mathcal{C}} \text{ and } \mathrm{Var}[\mathbf{Z}_i] = \frac{\mathcal{B}}{\mathcal{C}} \left(1 - \frac{1}{\mathcal{C}}\right).$$

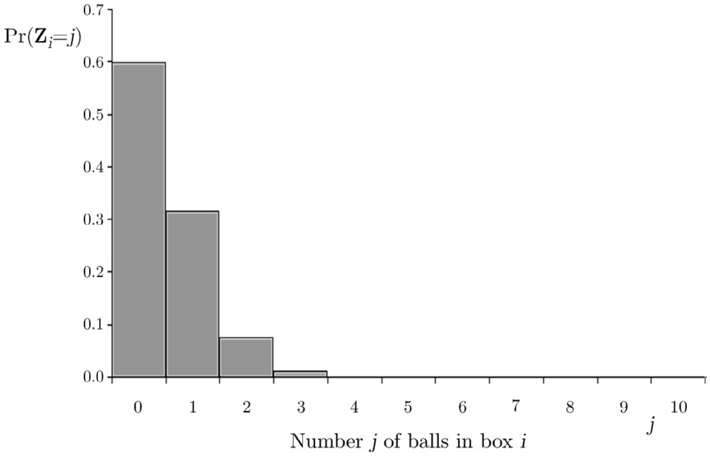
[24] I wish to thank Edlyn Teske for raising this possibility.

[25] Indeed, when $r = 1$ and $h = 2$, we expect that $l - \#\Gamma + 1 \approx l - 0.7869l = 0.2131l$.

[26] Chosen uniformly at random, of course.

Notice that both this mean and variance are less or equal to one since $\mathcal{B} \leq \mathcal{C}$. The possible values for j being 0 up to $\mathcal{B}$, we thus expect to have a positively skewed asymmetrical distribution.[27] In other words, using the analogy with the center of mass[28], this means that the majority of the weight of the distribution will be relatively close to zero.

Example 5.21 *Here is a tiny example to illustrate. The probability mass function for* $\mathcal{B} = 10$ *and* $\mathcal{C} = 20$ *is given explicitly in Figure 5.21: notice that the probability that box* i *contains at most two balls is already* $98,8\%$. *Moreover, a common mistake is to think that since we have 10 balls and 20 boxes, the probability that a box remains empty is* 50%, *while in reality it is almost* 60%. *However, we do have* $\mathrm{E}[\mathbf{Z}_i] = 0,5$ *(and* $\mathrm{Var}[\mathbf{Z}_i] = 0,475$*).*



j	$\Pr(\mathbf{Z}_i = j)$
0	$0,599$
1	$0,315$
2	$0,0746$
3	$0,0105$
4	$9,65 \times 10^{-4}$
5	$6,09 \times 10^{-5}$
6	$2,67 \times 10^{-6}$
7	$8,04 \times 10^{-8}$
8	$1,59 \times 10^{-9}$
9	$1,86 \times 10^{-11}$
10	$9,77 \times 10^{-14}$

Figure 5.21: Binomial distribution corresponding to $\mathcal{B} = 10$ and $\mathcal{C} = 20$

We now want to formalize our intuition that finding a box with a large number of balls is very unlikely (and at the same time get a better idea of what *'large'* and *'very unlikely'* mean in this context). As we now see, this will be a relatively easy task since it will turn out that a suitable upper bound can be obtained even if we make several gross approximations along the way. Concretely, for an integer L satisfying $0 < L < l$, we are first seeking an (easy to analyse and compute) upper bound for the probability that box i contains more than L balls:

[27] As opposed to a bell shape that we might be tempted to approximate by a normal distribution...

[28] *"Suppose* $\mathbf{X}$ *is a discrete random variable with values* x_i *and corresponding probabilities* p_i. *Now consider a weightless (horizontal) rod on which are placed weights, at locations* x_i *along the rod and having masses* p_i *(whose sum is one). The point at which the rod balances (its center of gravity) is* $\mathrm{E}[\mathbf{X}]$*."* - From Wikipedia, `http://en.wikipedia.org/wiki/Expected_value`.

$$\begin{aligned}
\Pr\left(\mathbf{Z}_i > L\right) &= \sum_{j=L+1}^{\mathcal{B}} \binom{\mathcal{B}}{j} \frac{1}{\mathcal{C}^j} \left(1 - \frac{1}{\mathcal{C}}\right)^{\mathcal{B}-j} \leq \sum_{j=L+1}^{\mathcal{B}} \binom{\mathcal{B}}{j} \frac{1}{\mathcal{C}^j} \\
&\leq \sum_{j=L+1}^{\mathcal{B}} \frac{\mathcal{B}^j}{j!} \cdot \frac{1}{\mathcal{C}^j} = \sum_{j=L+1}^{\mathcal{B}} \frac{1}{j!} \cdot \left(\frac{\mathcal{B}}{\mathcal{C}}\right)^j \leq \sum_{j=L+1}^{\mathcal{B}} \frac{1}{j!} \\
&= \sum_{j=0}^{\mathcal{B}} \frac{1}{j!} - \sum_{j=0}^{L} \frac{1}{j!} \leq \sum_{j=0}^{\infty} \frac{1}{j!} - \sum_{j=0}^{L} \frac{1}{j!} = e - \sum_{j=0}^{L} \frac{1}{j!}.
\end{aligned}$$

Notice that this upper bound does not even depend on the particular values of $\mathcal{B}$ and $\mathcal{C}$, but merely on the fact that $\mathcal{B} \leq \mathcal{C}$. We can thus build the following pocket size table (Table 5.6) to serve as a general rule of thumb.

L	$e - \sum_{j=0}^{L} \frac{1}{j!}$
2	0.218
4	0.00995
8	$3,06 \times 10^{-6}$
16	$2,98 \times 10^{-15}$
32	$1,19 \times 10^{-37}$
64	$1,23 \times 10^{-91}$
128	$2,03 \times 10^{-218}$

Table 5.6: Values of $e - \sum_{j=0}^{L} 1/j!$ for small powers of 2

From Table 5.6, we see that the probability that a given box contains more than 64 balls is already less than one chance over the (estimated!) number of atoms in the observable universe. Very well. However, we are here after an even stronger result. Indeed, it is not enough for us to know that any *particular* box has a very small probability of containing more than L balls, since the statement that we wish to make is that *among all $\mathcal{C}$ boxes, it is very unlikely to find even one box with more than L balls.* For this purpose, let $\mathbf{Z} = \max(\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{\mathcal{C}})$ be the maximum number of balls found within one box. Then,

$$\begin{aligned}
\Pr\left(\mathbf{Z} > L\right) &= \Pr\left((\mathbf{Z}_1 > L) \cup (\mathbf{Z}_2 > L) \cup \ldots \cup (\mathbf{Z}_{\mathcal{C}} > L)\right) \\
&\leq \Pr\left(\mathbf{Z}_1 > L\right) + \Pr\left(\mathbf{Z}_2 > L\right) + \ldots + \Pr\left(\mathbf{Z}_{\mathcal{C}} > L\right) \\
&= \mathcal{C} \cdot \Pr\left(\mathbf{Z}_1 > L\right).
\end{aligned}$$

A small calculation then yields the desired result when $r = 1$, $q \approx 2^{160}$ and L is chosen to equal, say, one hundred. Indeed,

$$\Pr\left(\mathbf{Z} > 100\right) \leq \mathcal{C} \cdot \Pr\left(\mathbf{Z}_1 > 100\right) \leq (q-1)\left(e - \sum_{j=0}^{100} \frac{1}{j!}\right) \approx 1,57 \times 10^{-112}.$$

Thus when $r = 1$ and $q \approx 2^{160}$ is of cryptographic size, the probability that there is a box containing more than a hundred balls is at most $1,57 \times 10^{-112}$. So in practice, we expect that *all* boxes will contain at most 100 balls. If a similar statement holds for generalized Jacobians, it would imply that for any $\nu_i \in \Gamma$, there is a negligible probability that a table look-up of the value $\log_\lambda \nu_i$ is needed when solving a randomly and uniformly chosen instance of the DLP in $\mathbb{F}_{q^r}^* \times \langle B \rangle$ as outlined in Figure 5.10. There would then be no significant advantage for an adversary (who wish to proceed as in Figure 5.10) to precompute $\log_\lambda \nu_i$.

We now take a look at experimental data. For each of the samples we considered in last section, we had also recorded the maximal number of balls within a box that we encountered, both for generalized Jacobians and for the pseudo-random counterpart. The results for $r = 1$ are shown in Table 5.7.

		$2^{14} < q < 2^{16}$	$2^{16} < q < 2^{18}$	$2^{18} < q < 2^{20}$
$h = 2$	Generalized Jacobians	8	9	9
	Pseudo-Random	8	9	9
$h = 3$	Generalized Jacobians	7	7	8
	Pseudo-Random	7	8	8
$h = 4$	Generalized Jacobians	6	7	8
	Pseudo-Random	6	7	7

Table 5.7: Maximal number of balls within a box encountered for $r = 1$ and sample size two thousand

To obtain a similar result when $r > 1$, it suffices to be just a little more careful with the upper bounds we choose. First notice that the current upper bound we have on $\Pr\left(\mathbf{Z} > L\right)$ depends on r:

$$\Pr\left(\mathbf{Z} > L\right) \leq \mathcal{C} \cdot \Pr\left(\mathbf{Z}_1 > L\right) = \left(q^r - 1\right) \cdot \Pr\left(\mathbf{Z}_1 > L\right).$$

It is however possible to obtain an upper bound which will solely depend on L. This can be achieved as follows. Instead on merely relying on the fact that $\mathcal{B}/\mathcal{C} \leq 1$, we will now make use of the slightly stronger inequality:

$$\frac{\mathcal{B}^3}{\mathcal{C}^2} \leq 1.$$

This holds for any value of $r > 1$ (as soon as $q > 7$) since

$$\mathcal{B}^3 \leq \left(q + 1 + 2\sqrt{q}\right)^3 = \left(\sqrt{q} + 1\right)^6 = \left(q^{3/2} + 3q + 3\sqrt{q} + 1\right)^2 \leq \left(q^2 - 1\right)^2 \leq \left(q^r - 1\right)^2 = \mathcal{C}^2.$$

We therefore have, for $r > 1$ and $1 < L < l$,

$$\begin{aligned}
\Pr(\mathbf{Z} > L) &\leq \mathcal{C} \cdot \Pr(\mathbf{Z}_1 > L) = \mathcal{C} \cdot \sum_{j=L+1}^{\mathcal{B}} \binom{\mathcal{B}}{j} \frac{1}{\mathcal{C}^j} \left(1 - \frac{1}{\mathcal{C}}\right)^{\mathcal{B}-j} \\
&\leq \sum_{j=L+1}^{\mathcal{B}} \binom{\mathcal{B}}{j} \frac{1}{\mathcal{C}^{j-1}} \leq \sum_{j=L+1}^{\mathcal{B}} \frac{\mathcal{B}^j}{j!} \cdot \frac{1}{\mathcal{C}^{j-1}} = \sum_{j=L+1}^{\mathcal{B}} \frac{\mathcal{B}^3}{\mathcal{C}^2} \cdot \left(\frac{\mathcal{B}}{\mathcal{C}}\right)^{j-3} \cdot \frac{1}{j!} \\
&\leq \sum_{j=L+1}^{\mathcal{B}} \frac{1}{j!} = \sum_{j=0}^{\mathcal{B}} \frac{1}{j!} - \sum_{j=0}^{L} \frac{1}{j!} \leq \sum_{j=0}^{\infty} \frac{1}{j!} - \sum_{j=0}^{L} \frac{1}{j!} = e - \sum_{j=0}^{L} \frac{1}{j!}.
\end{aligned}$$

Thus using our pocket size Table 5.6, we now get that for *any* $r > 1$ and $q > 7$, the probability that at least one box contains more than 64 balls is at most $1,23 \times 10^{-91}$.

Our simulations for $r = 2$ once more agree with this prediction: as a matter of fact, the largest number of balls found into one box was three. These amazingly simple results are summarized in Table 5.8.

		$2^{14} < q < 2^{16}$	$2^{16} < q < 2^{18}$	$2^{18} < q < 2^{20}$
$h = 1$	Generalized Jacobians	3	2	2
	Pseudo-Random	2	2	2
$h = 2$	Generalized Jacobians	2	2	2
	Pseudo-Random	2	2	2
$h = 3$	Generalized Jacobians	2	2	2
	Pseudo-Random	2	2	2
$h = 4$	Generalized Jacobians	2	2	2
	Pseudo-Random	2	2	2

Table 5.8: Maximal number of balls within a box encountered for $r = 2$ and sample size two thousand

Finally, if we are ready to believe that the Classical Occupancy Problem provides a reasonable approximation of generalized Jacobians (in terms of the maximal number of balls that can be found within one box), we then infer that an adversary has no tangible gain in precomputing a table of chosen elements of Γ along with their discrete logarithms. But of course, providing a formal proof instead of a heuristic argument is another story...

We have therefore seen in this chapter that the simple generalized Jacobian $\mathbb{F}_{q^r}^* \times \langle B \rangle$ fulfills all the conditions for a group to be suitable for discrete logarithm-based cryptography. This therefore provides the first example of a generalized Jacobian with nontrivial $L_\mathfrak{m}$ *and* J that could be used in public-key cryptography.

Chapter 6

Conclusion and Further Work

> *"Une approche qui débouche sur de bons problèmes doit fatalement donner quelque chose de bien."*
>
> *"An approach leading to challenging problems must inevitably yield something good."*
>
> *- Henri Darmon*

Throughout this thesis, we have used an approach by exploration in order to be as transparent as possible concerning the paths we followed when presenting original results. We hope that this unusual style for research related reports provided a satisfactory motivation at every step of the way.

This conclusion and further work will also follow the same lines: we will not draw a definitive conclusion nor provide a precise program of research for further work. The reason is simple: since this thesis introduced the use of generalized Jacobians to build cryptosystems, we believe that it would be premature, at such an early stage, to pretend that we now see enough of the picture to predict the future of generalized Jacobians in cryptography. In contrast, the approach we will follow comprises two parts: a quick summary of results, followed by a list of ideas for future explorations.

6.1 Summary of Results

In this dissertation, we have presented and studied generalized Jacobians from a cryptographic point of view. In particular, we have seen how several popular public-key cryptosytems can in fact be reinterpreted in the language of generalized Jacobians. From that point on, the relevance of these algebraic groups in cryptography was already established.

However, all of these cryptosystems had an underlying group with either a trivial linear group $L_{\mathfrak{m}}$ or a trivial Jacobian J. The next step was then to consider a generalized Jacobian with nontrivial $L_{\mathfrak{m}}$ *and* J.

Concretely, we chose to consider a generalized Jacobian of an elliptic curve (with respect to a modulus of degree 2 formed by points of $E(\mathbb{F}_{q^r})$) which was neither an abelian variety nor a torus in order to provide the first instance of a semi-abelian variety suitable for cryptography.

We have shown how the elements can be compactly represented, the group law efficiently computed and the group order readily determined. Lastly, we have proved that the DLP in this generalized Jacobian is at least as hard as the DLP in $E(\mathbb{F}_q)$ and at least as hard as the DLP in $\mathbb{F}_{q^r}^*$.

As a result, the group we obtained possesses a discrete logarithm problem that combines, in a natural fashion, the two most studied discrete logarithm problems to this date.

In the meantime, we have also characterized two subfamilies of generalized Jacobians on an elliptic curve which present different cryptographic properties: we therefore introduced the new concept of B-related and B-unrelated moduli in order to distinguish these two cases.

Finally, of independent interest is our discovery of an infinite family of unified point addition formulæ for elliptic curves given by a general Weierstraß equation. This therefore provides countermeasures to side-channel attacks on elliptic curve cryptosystems.

6.2 Work in Progress, Further Work and Open Problems

In the 1970s, Whitfield Diffie was keeping with him a list of what he called *Problems for an ambitious theory of cryptography* [Fur92]: whenever he encountered a problem that seemed interesting, he would jot it down on his list. Unfortunately, this precious document has since disappeared. But the rest is history...

Of course, we have no pretention of comparing our work to that of Diffie; still, while progressing in this thesis, we naturally kept track of the possible topics we could see for further work. The list we present here is therefore not a formal program of research, but rather a broad variety of problems that arose from our search; it is thus meant as a notebook in perpetual progression.

EXTENSION OF (ALGEBRAIC) GROUPS. In Chapter 5, we have studied the DLP of a specific generalized Jacobian. However, we already know some properties of the group law of an arbitrary generalized Jacobian, thanks to the theory of extensions of algebraic groups [Ser88, Chapter VII]. Is it possible to use this knowledge to derive further properties of the DLP in a general $J_\mathfrak{m}$?

ADD MORE COPIES OF $\mathbb{G}_\mathrm{m}$. From the presentation we made in Chapter 4, it was already clear that the simplest case of a generalized Jacobian of an elliptic curve with nontrivial $L_\mathfrak{m}$ and J is the case we chose to treat in Chapter 5. Now that we know that this group is suitable for DL-based cryptography, we may wonder what the situation would be if we considered a modulus of higher degree such that $L_\mathfrak{m}$ would now be an algebraic torus of higher dimension. Would we have in this case an interesting efficiency/security ratio? This analysis would of course have to take into account the compression factor inherited from the algebraic torus.

COMPARISON OF THE DLP IN TWO GENERALIZED JACOBIANS. Given a smooth algebraic curve C and two effective divisors $\mathfrak{m}_1$ and $\mathfrak{m}_2$ such that $\mathfrak{m}_1 \geq \mathfrak{m}_2$, what can be said about the relationship between the discrete logarithm problem in $J_{\mathfrak{m}_1}$ and in $J_{\mathfrak{m}_2}$?

HYPERELLIPTIC CURVES. Since the work of David Cantor [Can87] we are able to efficiently compute in the Jacobian of hyperelliptic curves. Since then, the method has been refined, and we are more than ever convinced that hyperelliptic curves of low genus are an interesting alternative to ECC. For more details, see *Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves* [PWGP03]. The main observation here is that since the group law on hyperelliptic curves of genus greater than one no longer only involves straight lines, then this will inevitably be reflected in the group law algorithm of their generalized Jacobians. It would thus be interesting to know if the efficient explicit formulæ already obtained for curves of low genus could be extended in order to efficiently compute in the corresponding generalized Jacobians.

EFFICIENCY OF THE GROUP LAW. In Section 5.3, we have derived a natural group law for the generalized Jacobian of an elliptic curve with respect to a modulus of the form $\mathfrak{m} = (M) + (N)$, where M, N are distinct nonzero points of E. This group law was based on the explicit bijection between $\mathrm{Pic}^0_\mathfrak{m}(E)$ and $\mathbb{G}_\mathrm{m} \times E$ that we obtained in Section 5.2. However, there may exist other ways to compute this group operation that would be more efficient.

WEIL AND TATE PAIRINGS. Anyone familiar with the explicit methods to compute the Weil and Tate pairings will have noticed some similarities with the group law of the generalized Jacobians of Chapter 5. Can the abundant litterature on the efficient computations of pairings be used for generalized Jacobians as well?

MUMFORD THETA GROUPS. The Mumford Theta groups, also called finite Heisenberg groups, are also extensions of abelian varieties by the multiplicative group $\mathbb{G}_\mathrm{m}$. Moreover, Miller's method for computing the Weil pairing can be reinterpreted in terms of the Theta groups, as sketched in [Mil04]. It would thus be an interesting avenue to explore Theta groups with cryptographic applications in mind. Further details on Theta groups can be found in [Gor02, Section 3.2]

THE CASE $M = -N$. With the settings of Chapter 5, we can easily see that the expression for the group law when $M = -N$ greatly simplifies since $x_M = x_N$. Is it possible to build an attack based on this? Or can we demonstrate that the DLP is still believed to be intractable in this case?

B-RELATED MODULI. In Section 5.3.4, we have pointed out that B-unrelated moduli seemed to be more efficient for concrete applications and less susceptible to side-channel attacks. However, this observation is based on the specific group law algorithm that we obtained. There is therefore more investigation to be done before we can claim that B-related moduli are less attractive for cryptographic purposes.

CHARACTERISTIC TWO. The simulations of Section 5.5.3 have been made only in the case of odd characteristic. An analogous study for elliptic curves over fields of characteristic two should be undertaken, since these curves are so important for cryptographic purposes.

PROBABILISTIC COUNTING. The simulations of Section 5.5.3 were restricted to relatively small values of q since the required computations were quite involving. An alternative would be to consider larger values of q, but instead of computing the *exact* cardinality of Γ, one may be able to improve the efficiency by considering approximations of $\#\Gamma$.

TO WHAT EXTENT DO THE ν'S BEHAVE LIKE RANDOM NUMBERS? We already saw two situations where the experiments show that the values of ν seem to behave like randomly chosen numbers. How hard is it to distinguish between such values generated from a generalized Jacobian and true random numbers? If this problem turns out to be easy, could it be used to mount an attack against a generalized Jacobian cryptosystem?

COMPUTATIONAL DIFFIE-HELLMAN PROBLEM. In Chapter 5, we have extensively studied the links between the DLP in the generalized Jacobians and the DLPs in the elliptic curve and in the finite field. However, the security of many protocols is based on the Computational Diffie-Hellman Problem (CDHP). It would therefore be relevant to study the potential correlation between the three following CDHPs: in the generalized Jacobian, in the elliptic curve and in the finite field.

PLAY WITH THE EQUIVALENCE RELATION. In this thesis, linear and $\mathfrak{m}$-equivalence of divisors played a central role. Indeed, the former leads to usual Jacobians while the latter yields generalized Jacobians. Thus, it may be worthwhile to explore other equivalence relations on divisors, both with a cryptographic and a cryptanalytic perspective in mind. Since much research on such equivalence relations has been done, it is even possible that we already possess all the tools at hand to use these notions in cryptology.

KEEP THE MODULUS SECRET. Finally, we mention an avenue that may be more a curiosity than an actual open question. The author is far from being a specialist in protocols, but somehow thinks that another interesting possibility would be to keep the modulus secret. For instance, in the case we studied in Chapter 5, suppose that the ECC parameters are publicly known but that the values of M and N are shared only among a select group of individuals. Then, it may be advantageous to have a common public-key infrastructure (PKI) that could both serve for elliptic and generalized Jacobians cryptosystems. Another possibility would be to explore if there would be any advantage of sharing the modulus among several parties. With the cryptosystem of Chapter 5, suppose for instance that Alice knows M and that Bob knows N. Then, they can certainly compute in the generalized Jacobian $J_{\mathfrak{m}}$ if they pool their shares. Would there be an advantage in proceeding this way? And if so, what can be said about the difficulty of recovering a modulus from partial information?

This naive list comprises the security and efficiency aspects, the possible generalizations that could be made as well as other directions that may be followed. Some of these problems appear to be easy, and some look challenging: this great diversity then shows that many avenues are open for future research in this area.

Bibliography

[Adl79] Leonard M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *20th Annual Symposium on Foundations of Computer Science (FOCS '79)*, pages 55–60. IEEE Computer Society Press, 1979.

[Adl94] Leonard M. Adleman. The function field sieve. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*, number 877 in Lecture Notes in Computer Science, pages 108–121. Springer-Verlag, 1994.

[AM93] A.O.L. Atkins and F. Morain. Elliptic curves and and primality proving. *Mathematics of computation*, 61:29–68, 1993.

[Art21] Emil Artin. *Quadratische Körper im Gebiete der höheren Kongruenzen.* PhD thesis, Jahrb. phil. Fak. Leipzig, 1921.

[Bar02] Paulo Barreto. *The pairing-based crypto lounge*, Online since 2002. Available at http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Advances in cryptology—CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, Berlin, 2001.

[BF03] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[BHV02] Wieb Bosma, James Hutton, and Eric R. Verheul. Looking beyond XTR. In *Advances in cryptology—ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 46–63. Springer, Berlin, 2002.

[BJ03] Billet and Joye. The jacobi model of an elliptic curve and side-channel analysis. In T. Hoholdt M. Fossorier and A. Poli, editors, *Applied Algebra, Algebraic algorithms and Error-correcting Codes*, volume 2143 of *Lecture Notes in Computer Science*, pages 34–42. Springer, 2003.

[BK98] R. Balasubramanian and Neal Koblitz. The improbablity that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *Journal of Cryptology*, 10(11):141–145, 1998.

[BL95] Dan Boneh and Richard J. Lipton. Quantum cryptanalysis of hidden linear functions. In Don Coppersmith, editor, *Advances in Cryptology - CRYPTO'95*, volume 963 of *Lecture Notes in Computer Science*, pages 424–437. Springer, 1995.

[Bla79] George R. Blakley. Safeguarding cryptographic keys. In Richard E. Merwin, Jacqueline T. Zanca, and Merlin Smith, editors, *1979 National Computer Conference*, volume 48 of *AFIPS Conference proceedings*, pages 313–317. AFIPS Press, 1979.

[Blu82] Manuel Blum. Coin flipping by telephone: A protocol for solving impossible problems. In *Proceedings of the 24th IEEE Computer Conference*, pages 133–137, 1982. See also *ACM SIGACT News*, 15(1), 1983.

[BN05] Paulo S.L.M. Barreto and Michael Naehrig. Pairing friendly elliptic curves of prime order. In *Selected Areas in Cryptography (SAC 2005)*, 2005.

[BP98] Daniel V. Bailey and Christof Paar. Optimal extension fields for fast arithmetic in public-key algorithms. In *Advances in Cryptology—CRYPTO 1984)*, volume 1462 of *Lecture Notes in Computer Science*, pages 472–485. Springer, 1998.

[Bre80] Richard P. Brent. An improved Monte Carlo factorization algorithm. *BIT*, 20(2):176–184, 1980.

[BS82] Robert G. Bartle and Donald R. Sherbert. *Introduction to real analysis*. John Wiley and son, New York, USA, 1982.

[BS04] Paulo S.L.M Barreto and Michael Scott. *Generating more MNT elliptic curves*, 2004. Cryptology ePrint Archive, Report 2004/058 (http://eprint.iacr.org).

[BSS99] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic curves in cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, New York, USA, 1999.

[BSS05] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Advances in elliptic curve cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, New York, USA, 2005.

[BSW94] Johannes A. Buchmann, Renate Scheidler, and Hugh C. Williams. A key-exchange protocol using real quadratic fields. *Journal of Cryptology*, 7(3):171–199, 1994.

[BW88] Johannes A. Buchmann and Hugh C. Williams. A key-exchange system based on imaginary quadratic fields. *Journal of Cryptology*, 1(2):107–118, 1988.

[BW90] Johannes A. Buchmann and Hugh C. Williams. A key-exchange system based on real quadratic fields. In *CRYPTO: Proceedings of CRYPTO 89*, volume 435 of *Lecture Notes in Computer Science*, pages 335–343, 1990.

[BW03] Friederike Brezing and Annegret Weng. *Elliptic curves suitable for pairing based cryptography*, 2003. Cryptology ePrint Archive, Report 2003/143 (http://eprint.iacr.org).

[CA89] David Chaum and Hans Van Antwerpen. Undeniable signatures. In G. Brassard, editor, *Advances in Cryptology—CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 212–216. Springer-Verlag, August 1989.

[Can87] David G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Mathematics of Computation*, 48(177):95–101, January 1987.

[CF05] Henri Cohen and Gerhard Frey, editors. *Handbook of Elliptic and Hyperelliptic Curve Cryptogrpahy.* Chapman and Hall/CRC, Boca Raton, 2005.

[CGMA85] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th Annual Symposium on Foundations of Computer Science*, pages 383–395. IEEE, 1985.

[Con99] Ian Connell. *Elliptic Curve Handbook*, 1999. Available at http://www.math.mcgill.ca/connell/public/ECH1/. 553 pages.

[Cop84] D. Coppersmith. Fast evaluation of logarithms in fields of characterstic two. *IEEE Transactions on Information Theory*, 30(4):587–594, 1984.

[COS86] D. Coppersmith, A. M. Odlyzko, and R. Schroeppel. Discrete logarithms in GF(p). *Algorithmica*, 1(1):1–15, 1986.

[CR00] Certicom Research. *Standards for efficient cryptography*, 2000. Available at http://www.secg.org.

[CvHP92] David Chaum, Eugène van Heijst, and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In J. Feigenbaum, editor, *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 470–484. Springer-Verlag, 1992.

[DEM05] Régis Dupont, Andreas Enge, and François Morain. Building curves with arbitrary small MOV degree over finite prime fields. *Journal of Cryptology*, 18(2):79–89, 2005.

[Den82] Dorothy Elizabeth Robling Denning. *Cryptography and data security.* Addison-Wesley, Reading, USA, 1982.

[Deu41] M. Deuring. Die typen der multiplikatorenringe elliptischer funktionenkörper. *Abh. Math. Sem. Hansischen Univ.*, 147:197–272, 1941.

[DH76a] Whitfield Diffie and Martin Hellman. Multiuser cryptographic techniques. In *Proc. AFIPS 1976 National Computer Conference*, pages 109–112. AFIPS, 1976.

[DH76b] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[Die85] Jean Dieudonné. *History of Algebraic Geometry.* Wadsworth Advanced Books and Software, Monterey, 1985.

[DvOW92] Whitfield Diffie, Paul C. van Oorschot, and Michael Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.

[DW04] M. Van Dijk and D. Woodruff. Asymptotically optimal communication for torus-based cryptography. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 157 – 178. Springer-Verlag, 2004.

[ElG85a] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology—CRYPTO 84)*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, Berlin, 1985.

[ElG85b] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Fel68] William Feller. *An introduction to probability theory and its applications. Vol. I.* Third edition. John Wiley & Sons Inc., New York, 1968.

[FMR99] Gerhard Frey, M.Ḿuller, and Hans-Georg Rück. The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45:1717–1718, 1999.

[Ful69] William Fulton. *Algebraic Curves.* Mathematics Lecture Note Series. W. A. Benjamin, New York-Amsterdam, 1969.

[Fur92] Franco Furger. *Interview with Whitfield Diffie on the Development of Public Key Cryptography* , 1992. Available at http://www.itas.fzk.de/mahp/weber/diffie.htm.

[Gal04] Steven Galbraith. *Easy decisions: Applications of pairings in cryptography*, 2004. available at http://www.isg.rhul.ac.uk/ sdg/chuo-uni.pdf.

[Gau00] Pierrick Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. *Lecture Notes in Computer Science*, 1807, 2000.

[Gen05] Rosario Gennaro. An improved pseudo-random generator based on the discrete logarithm problem. *Journal of Cryptology*, 18(2):91–110, 2005.

[GHS02] P. Gaudry, F. Hess, and N. P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *Journal of Cryptology*, 15(1):19–46, January 2002.

[Gol01] Oded Goldreich. *Foundations of cryptography: Basic tools.* Cambridge University Press, Cambridge, 2001.

[Gop88] V. D. Goppa. *Geometry and codes*, volume 24 of *Mathematics and its Applications (Soviet Series).* Kluwer Academic Publishers Group, Boston, 1988. Translated from the Russian by N. G. Shartse.

[Gor93] Daniel M. Gordon. Discrete logarithms in GF(p) using the number field sieve. *SIAM Journal on Discrete Mathematics*, 6(1):124–138, 1993.

[Gor98] Daniel M. Gordon. A survey of fast exponentiation methods. *Journal of Algorithms*, 27(1):129–146, 1998.

[Gor02] Eyal Z. Goren. *Lectures on Hilbert Modular Varieties and Modular Forms*, volume 14 of *CRM Monograph Series*. American Mathematical Society, Providence, 2002.

[GV05] Robert Granger and Frederik Vercauteren. On the discrete logarithm problem on algebraic tori. In *Advances in Cryptology (CRYPTO 2005)*, volume 3621 of *Lecture Notes in Computer Science*, pages 66–85. Springer, 2005.

[Har77] Robin Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1977.

[Has33] Helmut Hasse. Beweis des analogons der riemannschen vermutung für die artinschen und f.k. schmidtschen kongruenzzeta-funktionen in gewissen elliptischen fällen. *Nachr. Ges. Wiss. Göttingen, Math.-Phys. K.*, pages 253–262, 1933.

[HR83] Martin E. Hellman and Justin M. Reyneri. Fast computation of discrete logarithms in $GF(q)$. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology—CRYPTO 82*, pages 3–13. Plenum Press, New York and London, 1983.

[HS71] Peter Hilton and Urs Stammbach. *Course in Homological Algebra*. Number 4 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1971.

[HSS93] J. Håstad, A. W. Schrift, and A. Shamir. The discrete logarithm modulo a composite hides $O(n)$ bits. *Journal of Computer and System Sciences*, 47(3):376–404, December 1993.

[Hun74] Thomas W. Hungerford. *Algebra*, volume 73 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1974.

[Hus86] D. Husemöller. *Elliptic curves*, volume 111 of *Graduate texts in Mathematics*. Springer, 1986.

[IEE99] IEEE. *P1363: Standard Specifications For Public Key Cryptography, Draft P1363/D13*, 1999. Available at http://grouper.ieee.org/groups/1363/private/P1363-11-12-99-pdf.zip.

[Jac35] Carl Gustav Jakob Jacobi. De usu theoriae integralium ellipticorum et integralium abelianorum in analysi diophantea. *Crelle Journal für die reine und angewandte Mathematik*, 13:353–355, 1835.

[JK69] N. L. Johnson and S. Kotz. *Discrete Distributions*. Wiley Series in Probability and Mathematical Statistics. Wiley & Sons, Salt Lake City, 1969.

[JK77] N. L. Johnson and S. Kotz. *Urn Models and Their Applications*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, 1977.

[JN03] Antoine Joux and Kim Nguyen. Separating decision Diffie–Hellman from computational Diffie–Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–247, September 2003.

[JQ01] Marc Joye and Jean-Jacques Quisquater. Hessian elliptic curves and side-channel attacks. In D. Naccache Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *Lecture notes in computer science*, pages 402–410. Springer, 2001.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. *Journal des Sciences Militaires*, 1883.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology – CRYPTO ' 99*, volume 1666 of *Lecture Notes in Computer Science*, pages 399–397. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1999.

[KMV00] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. *Designs, Codes and Cryptography*, 19:173–193, 2000.

[Knu73] Donald E. Knuth. *The Art of computer programming, Vol. 3 : Sorting and Searching*. Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley, Reading, 1973.

[Knu81] Donald E. Knuth. *The Art of Computer Programming II: Seminumerical Algorithms*. Addison-Wesley Series in Computer Science and Information Processing. Addison–Wesley, Reading, Massachusetts, second edition, 1981.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.

[Kob89] Neal Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1(3):139–150, 1989.

[Kob90] Neal Koblitz. Constructing elliptic curve cryptosystems in characteristic 2. In *Advances in Cryptology—CRYPTO 1990)*, volume 537 of *Lecture Notes in Computer Science*, pages 156–167. Springer, 1990.

[Koc96] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology—CRYPTO 1996*, volume 1109, pages 104–113. International Association for Cryptologic Research, Springer-Verlag, Berlin, Germany, 1996.

[Lan01] Tanja Lange. *Efficient Algorithm on Hyperelliptic curves*, 2001. Ph.D. Thesis available at http://www.ruhr-uni-bochum.de/itsc/tanja/preprints/main.pdf.

[Lem03] Franz Lemmermeyer. *Higher Descent on Pell Conics: III: The First* 2*-descent*, 2003. Preprint available at http://www.fen.bilkent.edu.tr/ franz/publ.html.

[Len87] H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126:649–673, 1987.

[Len02] H. W. Lenstra, Jr. Solving the Pell equation. *Notices Amer. Math. Soc.*, 49(2):182–192, 2002.

[Lev01] Steven Levy. *CRYPTO: How the code rebels beat the government-saving privacy in the digital age*. Viking Penguin books, 2001.

[LS93] Michael J. J. Lennon and Peter J. Smith. LUC: A new public key system. Technical report, April 05 1993. Available at ftp://ripem.msu.edu/pub/crypt/docs/luc-public-key-paper.ps.Z.

[LS01] P.-Y. Liardet and N.P. Smart. Preventing spa/dpa in ecc system using the jacobi form. In D. Naccache Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 391–401. Springer, 2001.

[LV00] Arjen K. Lenstra and Eric R. Verheul. The XTR public key system. In Mihir Bellare, editor, *Advances in Cryptology—CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 1–19. International Association for Cryptologic Research, Springer-Verlag, Berlin, Germany, 2000.

[LV01] Arjen K. Lenstra and Eric R. Verheul. An overview of the XTR public key system. In *Public-key cryptography and computational number theory (Warsaw, 2000)*, pages 151–180. de Gruyter, Berlin, 2001.

[Mau94] Ueli M. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In Yvo G. Desmedt, editor, *Advances in Cryptology—CRYPTO 94*, volume 839 of *Lecture Notes in Computer Science*, pages 271–281. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1994.

[McC88] Kevin S. McCurley. A key distribution system equivalent to factoring. *Journal of Cryptology*, 1:95–105, 1988.

[McC89] Kevin S. McCurley. Cryptographic key distribution and computation in class groups. In *Number theory and applications (Banff, AB, 1988)*, volume 265 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 459–479. Kluwer Acad. Publ., Dordrecht, 1989.

[McC90] Kevin S. McCurley. The discrete logarithm problem. In Carl Pomerance, editor, *Cryptology and Computational Number Theory*, volume 42 of *Proceedings of Symposia in Applied Mathematics*, pages 49–74. American Mathematical Society, 1990.

[Men93] Alfred Menezes. *Elliptic Curve Public Key Cryptosystems*, volume 234 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, MA, 1993.

[Mic02] Sun Microsystems. *Elliptic Curve Cryptography: The Next Generation of Internet Security*, 2002. Available at http://research.sun.com/sunlabsday/ docs.2004/ECC-whitepaper.pdf.

[Mil86a] Victor S. Miller. Short programs for functions on curves. Technical report, 1986.

[Mil86b] Victor. S. Miller. Uses of elliptic curves in cryptography. In Hugh C. Williams, editor, *Advances in Cryptology—CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, Berlin, 1986. Springer-Verlag.

[Mil86c] J. S. Milne. Jacobian varieties. In *Arithmetic geometry*, pages 167–212. Springer, New York, 1986.

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

[MNT01] Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, E84-A(5):1234–1243, 2001.

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

[MTI86] T. Matsumoto, Y. Takashima, and H. Imai. On seeking smart public-key-distribution systems. *The Transactions of the IECE of Japan*, E69:99–106, 1986.

[Mum84] David Mumford. *Tata Lectures on Theta II*, volume 43 of *Prog. in Math.* Birkhäuser, Basel, 1984.

[MvOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press series on discrete mathematics and its applications. CRC Press, Boca Raton, 1996.

[MW96] Ueli M. Maurer and Stefan Wolf. Diffie-Hellman oracles. In Neal Koblitz, editor, *Advances in Cryptology—CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 268–282. Springer-Verlag, 1996.

[MW99] Ueli M. Maurer and Stefan Wolf. The relationship between breaking the Diffie–Hellman protocol and computing discrete logarithms. *SIAM Journal on Computing*, 28(5):1689–1721, October 1999.

[NIoST00] National Institute of Standards and Technology. *FIPS PUB 186-2: The Digital Signature Standard (DSS).* National Institute for Standards and Technology, Gaithersburg, 2000. Available at http://csrc.ncsl.nist.gov/fips/fips186-2.pdf.

[Odl85] Andrew M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In Thomas Beth, N. Cot, and I. Ingemarsson, editors, *Advances in cryptology: proceedings of EUROCRYPT 84*, volume 209 of *Lecture Notes in Computer Science*, pages 224–314, Berlin, 1985. Springer-Verlag.

[Odl00] Andrew M. Odlyzko. Discrete logarithms: the past and the future. *Designs, Codes, and Cryptography*, 19(2/3):129–145, 2000.

[Oka93] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *Advances in Cryptology—CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer-Verlag, 1993.

[Ols73] Loren D. Olson. An elementary proof that elliptic curves are abelian variety. *L'enseignement mathématique*, XIX:172–181, 1973.

[Per86] René C. Peralta. A simple and fast probabilistic algorithm for computing square roots modulo a prime number. *IEEE Transactions on Information Theory*, 32(6):846–847, 1986.

[PH78] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, 24:106–110, 1978.

[Pol78] John M. Pollard. Monte Carlo methods for index computation (mod$p$). *Mathematics of Computation*, 32(143):918–924, July 1978.

[Pol00] John M. Pollard. Kangaroos, monopoly and discrete logarithms. *Journal of Cryptology*, 13(4):437–447, 2000.

[PWGP03] Jan Pelzl, Thomas Wollinger, Jorge Guajardo, and Christof Paar. Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves. In *Workshop on Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2003.

[rBDJ] Éric Brier, Isabelle Déchène, and Marc Joye. Unified point addition formulae for elliptic curve cryptosystems. In N. Nedjah and L. de Macedo, editors, *Embedded Cryptographic Hardware: Methodolgies and Architectures.* Nova Science Publishers.

[Ros52] Maxwell Rosenlicht. Equivalence relations on algebraic curves. *Annals of Mathematics*, 56:169–191, July 1952.

[Ros54] Maxwell Rosenlicht. Generalized Jacobian varieties. *Annals of Mathematics*, 59:505–530, May 1954.

[Ros75] Maxwell Rosenlicht. Differential extension fields of exponential type. *Pacific J. Math.*, 57(1):289–300, 1975.

[Ros81] Michael Rosen. Abel's Theorem on the Lemniscate. *The American Mathematical Monthly*, 88(6):387–395, 1981.

[RS62] J. Barkley Rosser and Lowell Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois J. Math.*, 6:64–94, 1962.

[RS03] Karl Rubin and Alice Silverberg. Torus-based cryptography. In Dan Boneh, editor, *Advances in Cryptology—CRYPTO '03*, volume 2729 of *Lecture Notes in Computer Science*, pages 349–365. Springer-Verlag, 2003.

[RS04a] Karl Rubin and Alice Silverberg. Algebraic tori in cryptography. In *High primes and misdemeanours: lectures in honour of the 60th birthday of Hugh Cowie Williams*, volume 41 of *Fields Inst. Commun.*, pages 317–326. Amer. Math. Soc., Providence, 2004.

[RS04b] Karl Rubin and Alice Silverberg. *Miscellaneous results on algebraic tori*. Technical report, 2004. Available at http://www.math.uci.edu/ asilverb/bibliography/torusapp.pdf.

[RS04c] Karl Rubin and Alice Silverberg. Using primitive subgroups to do more with fewer bits. In Duncan Buell, editor, *Algorithmic Number Theory - ANTS 2004*, volume 3076 of *Lecture Notes in Computer Science*, pages 18–41. Springer-Verlag, Berlin Heidelberg, 2004.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sch85] Claus Peter Schnorr. Elliptic curves over finite fields and the computation of square roots module p. *Mathematics of Computation*, 44(170):483–494, 1985.

[Sch91] Claus Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Sch04] R. Schoof. *Structure of* $E(\mathbb{F}q)$, 2004. Talk presented during the ECRYPT Summer School on Elliptic Curves in Cryptography in Bochum, Germany.

[Ser75] Jean-Pierre Serre. *Groupes algébriques et corps de classes*. Hermann, Paris, 1975.

[Ser88] Jean-Pierre Serre. *Algebraic groups and class fields*, volume 117 of *Graduate texts in mathematics*. Springer-Verlag, New-York, 1988.

[Sev47] Francesco Severi. *Funzioni quasi abeliane*. Pontificiae Academiae Scientiarum Scripta Varia, v. 4. Vatican City, 1947.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Sha49] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, oct 1949.

[Sha71] Daniel Shanks. Class number, A theory of factorization and genera. In D. J. Lewis, editor, *Proceedings of the Symposion on Pure Mathematics*, pages 415–440. AMS, 1971.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Shafi Goldwasser, editor, *Proceedings: 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Computer Society Press, 1994.

[Sho97a] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[Sho97b] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT ' 97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1997.

[Sil86] Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986.

[Sil94] Joseph H. Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1994.

[Sin99] Simon Singh. *The code book: the evolution of secrecy from Mary Queen of Scots to quantum cryptography*. Anchor Books, New York, 1999.

[SS90] A. W. Schrift and A. Shamir. The discrete log is very discreet. In Baruch Awerbuch, editor, *Proceedings of the twenty-second annual ACM Symposium on Theory of Computing*, pages 405–415. ACM Press, 1990.

[ST92] Joseph H. Silverman and John Tate. *Rational points on elliptic curves*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1992.

[Sta96] Markus A. Stadler. Publicly verifiable secret sharing. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT ' 96*, volume 1070 of *Lecture Notes in Computer Science*. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1996.

[Tes01] Edlyn Teske. Computing discrete logarithms with the parallelized kangaroo method. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 130, 2001.

[The03] Nicolas Theriault. Index calculus attack for hyperelliptic curves of small genus. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*. LNCS, Springer-Verlag, 2003.

[TW95] Richard Taylor and Andrew Wiles. Ring-theoretic properties of certain hecke algebras. *Annals of Mathematics*, 141(3):553–572, 1995.

[vDGP$^+$05] Marten van Dijk, Robert Granger, Dan Page, Karl Rubin, Alice Silverberg, Martijn Stam, and David Woodruff. Practical cryptography in high dimensional tori. In *Advances in Cryptology—CRYPTO 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 234–250. Springer-Verlag, 2005.

[Ver19] Gilbert S. Vernam. *U.S. Patent 1,310,719*, 1919. Available at http://cryptome.org/vernam-patent.htm.

[Ver26] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal American Institute of Electrical Engineers*, XLV:109–115, 1926.

[vOW99] Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, 1999.

[Was03] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography.* Discrete Mathematics and its Application. CRC Press, Boca Raton, 2003.

[Wei40] André Weil. Sur les fonctions algébriques à corps de constantes fini. *C. R. Acad. Sci. Paris*, 210:592–594, 1940.

[Wei46] André Weil. *Foundations of algebraic geometry,*, volume 29 of *American Mathematical Society Colloquium Publications.* American Mathematical Society, New York, 1946.

[Wei48] André Weil. *Variétés abéliennes et courbes algébriques*, volume 1064 of *Actualités Sci. Ind.* Hermann & Cie, Paris, 1948.

[Wei55] André Weil. On algebraic groups of transformations. *Amer. J. Math.*, 77:355–391, 1955.

[Wei69] Edwin Weiss. *Cohomology of groups.* Pure and Applied Mathematics, Vol. 34. Academic Press, New York, 1969.

[Wil95] Andrew Wiles. Modular elliptic curves and Fermat's last theorem. *Annals of Mathematics*, 141(3):443–551, 1995.

Index